



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
REPUBLIK INDONESIA
2023

INFORMATIKA

Edisi Revisi

Mushthofa, dkk.

SMA/MA/SMK/MAK KELAS X

Hak Cipta pada Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia
Dilindungi Undang-Undang

Penafian: Buku ini disiapkan oleh Pemerintah dalam rangka pemenuhan kebutuhan buku pendidikan yang bermutu, murah, dan merata sesuai dengan amanat dalam UU No. 3 Tahun 2017. Buku ini disusun dan ditelaah oleh berbagai pihak di bawah koordinasi Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. Buku ini merupakan dokumen hidup yang senantiasa diperbaiki, diperbarui, dan dimutakhirkan sesuai dengan dinamika kebutuhan dan perubahan zaman. Masukan dari berbagai kalangan yang dialamatkan kepada penulis atau melalui alamat surel buku@kemdikbud.go.id diharapkan dapat meningkatkan kualitas buku ini.

Informatika untuk SMA/MA/SMK/MAK Kelas X (Edisi Revisi)

Penulis

Mushthofa
Auzi Asfarian
Hanson Prihantoro Putro
Irya Wisnubhadra
Endang Purnama Giri
Dean Apriana Ramadhan
Wahyono

Penelaah

Asep Wahyudin
Leli Alhapip

Penyelia/Penyelarass

Supriyatno
Lenny Puspita Ekawaty
Maharani Prananingrum
Eko Budiono

Kontributor

Sri Mujiasti
Gian Hadiansyah

Ilustrator

Nana Maulana

Editor

Christina Tulalessy
Lenny Puspita Ekawaty

Editor Visual

Is Yuniarto Nafawi

Desainer

Syndhi Renolarisa

Penerbit

Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi

Dikeluarkan oleh

Pusat Perbukuan
Kompleks Kemdikbudristek Jalan RS Fatmawati, Cipete, Jakarta Selatan
<https://buku.kemdikbud.go.id>

Edisi Revisi, 2023

ISBN 978-623-118-492-4 (no.jil.lengkap PDF)
ISBN 978-623-118-493-1 (jil.1 PDF)

Isi buku ini menggunakan Noto Serif 9/12 pt, Steve Matteson, Open Font License.
xii, 260 hlm.: 17,6 x 25 cm.

Kata Pengantar

Pusat Perbukuan; Badan Standar, Kurikulum, dan Asesmen Pendidikan; Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi memiliki tugas dan fungsi mengembangkan buku pendidikan pada satuan Pendidikan Anak Usia Dini, Pendidikan Dasar, dan Pendidikan Menengah, termasuk Pendidikan Khusus. Buku berkaitan erat dengan kurikulum. Buku yang dikembangkan saat ini mengacu pada kurikulum yang berlaku, yaitu Kurikulum Merdeka.

Salah satu bentuk dukungan terhadap implementasi Kurikulum Merdeka di satuan pendidikan ialah mengembangkan buku teks utama yang terdiri atas buku siswa dan panduan guru. Buku ini merupakan sumber belajar utama dalam pembelajaran bagi siswa dan menjadi salah satu referensi atau inspirasi bagi guru dalam merancang dan mengembangkan pembelajaran sesuai karakteristik, potensi, dan kebutuhan peserta didik. Keberadaan buku teks utama ini diharapkan menjadi fondasi dalam membentuk Profil Pelajar Pancasila yang beriman dan bertakwa kepada Tuhan yang Maha Esa, dan berakhlak mulia; berkebinekaan global, berjiwa gotong royong, mandiri, kritis, dan kreatif.

Buku teks utama, sebagai salah satu sarana membangun dan meningkatkan budaya literasi masyarakat Indonesia, perlu mendapatkan perhatian khusus. Pemerintah perlu menyiapkan buku teks utama yang mengikuti perkembangan zaman untuk semua mata pelajaran wajib dan mata pelajaran peminatan, termasuk Pendidikan Khusus. Sehubungan dengan hal itu, Pusat Perbukuan merevisi dan menerbitkan buku-buku teks utama berdasarkan Capaian Pembelajaran dalam Kurikulum Merdeka.

Kami mengucapkan terima kasih kepada semua pihak yang telah berkolaborasi dalam upaya menghadirkan buku teks utama ini. Kami berharap buku ini dapat menjadi landasan dalam memperkuat ketahanan budaya bangsa, membentuk mentalitas maju, modern, dan berkarakter bagi seluruh generasi penerus. Semoga buku teks utama ini dapat menjadi tonggak perubahan yang menginspirasi, membimbing, dan mengangkat kualitas pendidikan kita ke puncak keunggulan.

Jakarta, Desember 2023

Kepala Pusat Perbukuan,

Supriyatno, S.Pd., M.A.





Prakata



Selamat datang di jenjang SMA. Selamat bertemu kembali dengan Informatika, mata pelajaran yang membuat kamu akan menguasai dasar-dasar teknologi informasi. Buku ini merupakan bahan ajar utama mata pelajaran Informatika untuk siswa kelas X SMA dan sederajat.

Informatika ialah bidang ilmu mengenai studi, perancangan, dan pembuatan sistem komputasi, serta prinsip-prinsip yang menjadi dasar perancangan tersebut. Landasan berpikir untuk belajar informatika dinamakan berpikir komputasional (*computational thinking*). Berpikir komputasional merupakan suatu kerangka dan proses berpikir yang mencakup perangkat keras, perangkat lunak, dan menalar (*reasoning*) mengenai sistem dan persoalan. Moda berpikir (*thinking mode*) ini didukung dan dilengkapi dengan pengetahuan teoretis dan praktis, serta teknik untuk menganalisis, memodelkan, dan menyelesaikan persoalan.

Elemen pengetahuan Informatika dibagi menjadi 4 bagian: Berpikir Komputasional (BK), Algoritma dan Pemrograman (AP), Analisis Data (AD), dan Literasi Digital (LD). Keempat elemen pengetahuan ini akan dipelajari bersama di buku ini. Setiap materi disajikan dengan diawali penjelasan mengenai semua konsep dan pengetahuan yang perlu kamu pahami. Setiap konsep dan topik pengetahuan yang dibahas kemudian dilengkapi dengan beberapa aktivitas yang digunakan untuk menunjang dan menguji pemahamanmu terhadap konsep dan topik tersebut. Aktivitas dapat berupa aktivitas individu ataupun aktivitas berkelompok yang merangsang kamu untuk tidak hanya bernalar kritis, kreatif, tetapi juga bekerja sama dan bergotong royong. Aktivitas juga menggunakan moda *plugged* (membutuhkan komputer) dan *unplugged* (tidak memerlukan komputer). Hal ini agar kamu tetap dapat memahami konsep-konsep dasar informatika tanpa harus menggunakan perangkat komputer.

Akhir kata, semoga buku ini dapat menjadi sarana pembelajaran yang efektif dan bermanfaat bagi kamu dalam mempelajari Informatika. Semoga buku ini juga dapat membantu menyiapkan kamu sebagai generasi penerus bangsa yang memiliki daya saing yang tinggi di kancah dunia yang makin modern dan berteknologi tinggi. Selamat berselancar bersama Informatika.

Bogor, Februari 2023

Penulis





Daftar Isi

Kata Pengantar	iii
Prakata	iv
Daftar Isi	v
Daftar Gambar	vii
Daftar Tabel	ix
Petunjuk Penggunaan Buku	x
Bab 1 Data, Informasi, dan Validasinya	1
A. Pencarian dengan Banyak Variabel	3
B. Koleksi Data	8
C. Ekosistem Periksa Fakta	16
D. Evaluasi Data dan Sumbernya	18
Uji Kompetensi	22
Pengayaan	25
Refleksi	25
Bab 2 Algoritma dan Struktur Data	27
A. Mengetahui Algoritma dan Pemrograman	29
B. Membuat Program Sesuai Struktur Kendalinya	45
C. Memilih Algoritma untuk Masalah di Kehidupan Nyata	80
D. Memilih Struktur Data untuk Masalah di Kehidupan Nyata	96
Uji Kompetensi	106
Pengayaan	109
Refleksi	110
Bab 3 Teknologi dan Budaya Digital	111
A. Sistem Komputer	114
B. Otomatisasi dengan Perangkat Lunak Perkantoran	136
C. Pemrosesan Dokumen dengan Perangkat Lunak Perkantoran	148
D. Budaya Baru di Era Informatika	154
E. Budaya Karier dan Beragam Sertifikasi di Bidang Informatika	167

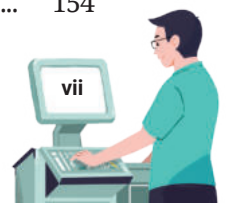


Uji Kompetensi	183
Pengayaan	185
Refleksi	186
Bab 4 Keamanan dan Manajemen Informasi.....	187
A. Pendahuluan	189
B. Jaringan Komputer dan Internet	193
C. Proteksi Data di Internet	204
D. Meningkatkan Keamanan Informasi	220
Uji Kompetensi	238
Pengayaan	239
Refleksi	239
Glosarium	240
Daftar Pustaka	245
Daftar Sumber Gambar	247
Indeks	247
Profil Pelaku Perbukuan	250



Daftar Gambar

Gambar 1.1	Contoh pemanfaatan kata kunci dalam mesin pencari.....	4
Gambar 1.2	Mesin pencarian dengan banyak variabel	5
Gambar 1.3	Pencarian dengan menyertakan gambar pada Google Lens	6
Gambar 1.4	Contoh tampilan halaman web pencari kerja.....	8
Gambar 1.5	Membuka <i>file</i> baru di Google Collab.	9
Gambar 1.6	Program Hello World dengan bahasa Python di Google Collab.	10
Gambar 1.7	Luaran Program Hello World	10
Gambar 1.8	Kode program untuk <i>parsing</i> sebuah alamat web.....	11
Gambar 1.9	Luaran program halaman web pencari kerja.	11
Gambar 1.10	Kode program untuk <i>parsing</i> halaman kedua	12
Gambar 1.11	Kode program untuk ambil detil lowongan pekerjaan	13
Gambar 1.12	Luaran kode program yang menghasilkan tabel rekapitulasi data	13
Gambar 2.1	Elemen generik dari bahasa pemrograman prosedural. ...	45
Gambar 2.2	Pencarian (a) buku di perpustakaan, (b) informasi di internet.	81
Gambar 2.3	Ilustrasi struktur data antrean dan tumpukan dan contohnya di dunia nyata.	96
Gambar 3.1	Diagram blok konseptual komputer yang mengacu pada arsitektur Von Neuman.	114
Gambar 3.2	Contoh Microcomputer Raspberry Pi 4	117
Gambar 3.3	Contoh Personal Computer	117
Gambar 3.4	Contoh Mini PC, Coffee Lake-U-based Bean Canyon Intel NUC8i5BEK2	118
Gambar 3.5	Contoh minicomputer PDP-8e minicomputer system	118
Gambar 3.6	Salah Satu Contoh Komputer Mainframe IBM z Systems z13	119
Gambar 3.7	Sierra/ATS-2 Supercomputer	119
Gambar 3.8	Ilustrasi Multitasking	121
Gambar 3.9	(a) Simulasi Penjadwalan Proses, (b) Simulasi CPU	124
Gambar 3.10	Diagram Arsitektur Sederhana Komputer	125
Gambar 3.11	Siklus mesin pada komputer sebelum dan setelah bekerja.	126
Gambar 3.12	Memory Address seperti Kotak Surat	128
Gambar 3.13	Siklus Ambil dan Jalankan (Fetch Execute Cycle)	129
Gambar 3.14	Muncul Sheet seperti pada Microsoft Excel	138
Gambar 3.15	Insert Chart pada MS Word	139
Gambar 3.16	Muncul Sheet seperti pada Microsoft Excel	139
Gambar 3.17	Ilustrasi Object Linking Antardua Dokumen	140
Gambar 3.18	Era Revolusi Industri 4.0	154



Gambar 3.19	Ilustrasi Ekonomi Kreatif di Indonesia	157
Gambar 3.20	Komik tentang Seseorang yang Sedang Membeli Komputer	163
Gambar 3.21	Ilustrasi Beberapa Profesi di Bidang Informatika	168
Gambar 3.22	Inovasi-inovasi unggul di bidang Informatika banyak bermula dari coretan-coretan kreatif di buku catatan atau jurnal seseorang.	180
Gambar 3.23	Tujuan Pembangunan Berkelanjutan	180
Gambar 3.24	Ilustrasi gambaran dunia di masa depan, Masyarakat 5.0 (<i>Society 5.0</i>) (a) abstraksi, (b) contoh aktivitas	181
Gambar 4.1	Contoh Jaringan Lokal	193
Gambar 4.2	Jaringan lokal dapat diubah menjadi jaringan internet.	195
Gambar 4.3	Diagram konektivitas internet jaringan menggunakan kabel melalui Dial-Up dan ASDL.	196
Gambar 4.4	Konektivitas Jaringan LAN	197
Gambar 4.5	Ilustrasi konektivitas jaringan nirkabel via GPRS	198
Gambar 4.6	Ilustrasi konektivitas jaringan nirkabel Wifi.	198
Gambar 4.7	Ilustrasi konektivitas jaringan nirkabel melalui satelit	199
Gambar 4.8	Salah satu alur kerja proses komunikasi data SMS melalui ponsel.	200
Gambar 4.9	Contoh kabel data USB-A ke USB-C	201
Gambar 4.10	Contoh ilustrasi komunikasi data menggunakan bluetooth	201
Gambar 4.11	Contoh ilustrasi komunikasi data menggunakan wifi.	202
Gambar 4.12	Kasus Perancangan Jaringan	203
Gambar 4.13	Enkripsi dan Dekripsi Data	205
Gambar 4.14	Cara Kerja Antivirus	209
Gambar 4.15	Contoh Situs dengan Perbedaan Satu Huruf	222
Gambar 4.16	Contoh tampilan pengguna untuk memilih mengaktifkan <i>cookie</i> di situs antivirus.	224
Gambar 4.17	Contoh tampilan pemberitahuan penggunaan <i>cookie</i> pada situs salah satu bank di Indonesia.	224
Gambar 4.18	Contoh pengaturan fitur <i>cookie</i> pada peramban Chrome. .	225
Gambar 4.19	Contoh pengaturan riwayat di facebook.....	226





Daftar Tabel

Tabel 1.1	Variabel yang Dapat Digunakan di Mesin Pencari Google..	5
Tabel 2.1	Simbol Diagram Alir Beserta Maknanya	31
Tabel 2.2	Ilustrasi Penelusuran Diagram Alir di Buku Kerja Siswa ...	38
Tabel 2.3	Pseudokode 1: Menghitung Luas Persegi	40
Tabel 2.4	Pseudokode 2: Menghitung Luas Permukaan Kubus	40
Tabel 2.5	Pseudokode 3: Membagi Bilangan	41
Tabel 2.6	Pseudokode 4: Menghitung Mundur dari N hingga 1	41
Tabel 2.7	Pseudokode 5: Mencari Bilangan Terbesar dari Suatu Himpunan Bilangan	42
Tabel 2.8	Spesifikasi Program yang Diberikan pada Problem Latihan	46
Tabel 2.9	Operator Matematika yang Umum Digunakan pada Bahasa	49
Tabel 2.10	Operator Logika, Relasional, dan Kesamaan pada Bahasa Pemrograman	51
Tabel 3.3	Profesi di Bidang Informatika	170
Tabel 3.4	Jenis-Jenis Profesi di Bidang Informatika	171



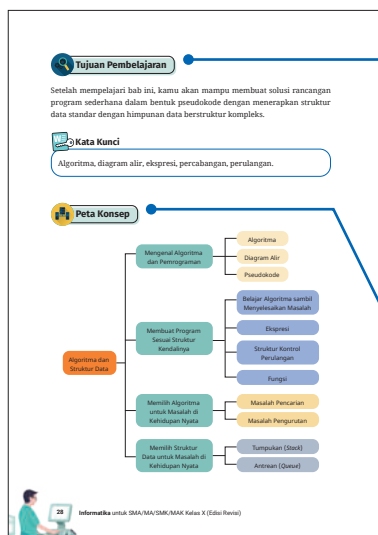
Petunjuk Penggunaan Buku

Fitur-Fitur dan Komponen dalam Buku Siswa

Buku teks pelajaran Informatika dirancang sebagai pendampingmu dalam melakukan pembelajaran. Buku ini berisi konsep dan aktivitas yang merangsang kamu untuk dapat belajar dan melakukan eksplorasi secara mandiri. Bagian-bagian penting dari buku ini ialah seperti berikut.

Pertanyaan Pemantik

Berisi pertanyaan yang memantik untuk berpikir dan menjawab pertanyaan mengenai hal baru yang akan dipelajari. kamu dapat mendiskusikan jawaban tersebut dengan guru.



Tujuan Pembelajaran

Berisi kompetensi utama yang akan dicapai pada bab tertentu pada buku. Tujuan pembelajaran diharapkan dapat kamu capai setelah proses pembelajaran pada bab tersebut selesai.

Peta Konsep

Berisi gambaran umum atau abstraksi dari konsep yang akan kamu pelajari pada tiap bab. Peta konsep berbentuk konsep yang terhubung.



Merupakan cerita awal atau kegiatan untuk mendekatkan konsep dan pengetahuan baru agar kamu dapat memusatkan perhatian sepenuhnya pada pengetahuan baru tersebut.

Di jenjang SMP, kamu telah mengeksplorasi dasar-dasar dunia data dan juga informasi digital. Misalnya, saat kamu mencari di buku atau menggunakan Google untuk mencari informasi tentang pelajaran idiom, dan kemudian harus membedakan antara artikel ilmiah dan opini pribadi. Di kelas ini, kamu akan mengambil langkah lebih jauh. Kamu akan belajar tidak hanya bagaimana cara mendapatkan data, tetapi juga bagaimana memproses dan menganalisisnya. Sebelum masuk lebih dalam, terdapat beberapa konsep yang dapat didiskusikan untuk membaurkan memahami materi. Jawablah beberapa pertanyaan berikut.

1. Dalam melakukan pencarian, terdapat variabel yang dapat digunakan. Apa itu variabel pencarian?

- Hasil dari suatu pencarian dapat berupa fakta atau bukan fakta. Apa itu fakta?
- Untuk mendukung sebuah fakta, ada ekosistem yang dapat diikuti. Apa itu ekosistem?
- Fakta juga perlu didukung dengan data yang valid dan sumber data yang valid. Apa itu valid?
- Untuk mengetahui sebuah informasi valid atau tidak, kamu dapat melakukan proses membaca lateral. Apa itu membaca lateral?

A. Pencarian dengan Banyak Variabel

Pernahkah kamu mengalami saat sedang mencari informasi menemukan bahwa

Kata Kunci

Daftar kata penting yang akan kamu pelajari pada tiap bab.

 Tujuan Pembelajaran

Setelah mempelajari bab ini, kamu akan mampu membuat solusi rancangan program sederhana dalam bentuk pseudokode dengan menerapkan struktur data standar dengan himpunan data berstruktur kompleks.

 Kata Kunci

Algoritma, diagram alir, ekspresi, percabangan, perulangan.

Aktivitas

Kegiatan pembelajaran berbasis siswa, yang mengharuskan kamu akan melakukan kegiatan untuk membentuk kompetensi yang telah ditetapkan pada tiap bab. Aktivitas ini dapat dilakukan secara individu, berpasangan, maupun kelompok untuk mendapatkan keseimbangan kompetensi dalam hal kemandirian dan yang sering dilakukan pada kegiatan pengembangan produk informatika. Guru akan memberikan penjelasan mengenai aktivitas ini dan kamu diharapkan dapat bereksplorasi secara maksimal. Jenis-jenis aktivitas ialah sebagai berikut.

Ayo, Berdiskusi!

Aktivitas yang memandu kamu untuk saling berdiskusi pada suatu topik yang diberikan. Di akhir diskusi, kamu akan berbagi dengan kelompok diskusi lainnya.

4. Setelah mengetahui aspek dan dampak ekonomi dari informatika, apakah kamu tertarik untuk mengembangkan bisnis yang menggunakan informatika?
5. mengetahui aspek budaya dan folk mengenai literasi di bidang informatika, apakah kamu akan memilih studi mengenai karir-karya di bidang informatika? Apakah kamu akan mulai mengorganisir pergerakan literasi, budaya, aspek, atau folk digital dalam dengan bernilai yang benar, dengan mengorganisir hal yang positifnya?

Apr 11 10:44 AM

Sih Program Studi dan Perguruan Tinggi

o, pertaina bimbang troli agi yang kalen nta citakan. Uraia setiap poin di
sach 14, jikirakulah dengan baik berdasarkan paparan di atas. Kalen juga
sa melakukan pencarian di internet atau membaca buku di perpustakaan
tak menjadi bahan berpikir. Kerenitas, tidikin hasil pemikiran kalian
da buku catatan.

- Apakah kalian bernilai untuk studi lanjutan di bidang kefarmasian? Mengapa?
Jika bernilai, program studi apa yang akan kalian pilih? Jelaskan alasannya.
Apabila tidak bernilai, menurut kalian, adakah pemse dari informanika dalam bidang yang kalian minati? Jelaskan seperti apa peran informanika menurut pemahaman kalian.

- Informetika untuk Masa Depan Indonesia:**
- aktivitas dan inovasi berasal dari berpikir dan berimajinasi (3-13). Banyak ilmu dan teknologi yang berasal dari mengimajinasikan bayangan mereka akan masa depan. Banyak yang mengimajinasikan seperti apa dunia di abad ke-21, dan telah jauh lebih baik karena manusia sudah mendesain teknologi. Imajinasi itu jadi menjadi standar manusia, tetapi akhirnya menjadi standar inspirasi para inovator dan penemu yang perlahan-lahan mengubah dunia dengan cara teknologi inovatif yang mereka ciptakan. Nah, ini, itulah semangat yang jadi modal berimajinasi dan menyebarkan seperti apa negeri kita ini.
- Negeri Kesatuan Republik Indonesia, di masa depan

Ayo, Renungkan!

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam jurnal.

1. Apakah kamu merasa bahwa membuat algoritma dapat mempermudah kamu dalam menyelesaikan masalah?
2. Mana yang paling mudah untuk kamu gunakan, diagram alir atau pseudokode?
3. Setelah ini, apa yang akan kamu lakukan untuk dapat membaca dan menulis algoritma dengan baik?
4. Pelajaran paling berkesan apa yang kamu dapatkan dari pertemuan ini?

B. Membuat Program Sesuai Struktur Kendalinya

Belajar bahasa pemrograman sama halnya dengan belajar bahasa apa pun, dimulai dengan secara intuitif mengenal dan langsung memakai bahasa tersebut untuk keperluan sehari-hari yang penting sesuai kebutuhan, bukan dari teori bahasa. Seseorang dengan bahasa ibu bahasa Indonesia, saat belajar bahasa Inggris, akan mulai mengenal bahasa Inggris melalui "membaca" contoh-contoh kalimat sederhana, yang sangat diperlukan dalam kehidupan sehari-hari, seperti mulai menyapa "Selamat pagi!", "Jam berapa?". Melalui contoh tersebut, dia akan belajar kosakata penting dan pola kalimat, misalnya kalimat pernyataan SPOK (Subjek, Predikat, Objek, Keterangan); struktur kalimat tanya, kalimat aklamasi, dan lain-lain. Selanjutnya, baru beranjak ke konsep yang lebih kompleks yang ada pada bahasa asing tersebut, dan mengenal tata bahasa secara lebih formal dan mulai menulis.

Belajar pemrograman pada hakikatnya sama dengan belajar bahasa natural (bahasa manusia sehari-hari): seseorang belajar dari "membaca" program terlebih dulu, daripada "menulis" kode program. Proses menulis kode (coding) dapat dimulai setelah kamu membaca contoh-contoh program yang menjadi pola pembangun program kompleks. Bedanya dengan belajar bahasa natural, teks dalam bahasa pemrograman yang ditulis bukan dilafalkan dan dipahami sebagai teks "statis", melainkan juga dapat dipahami oleh mesin dan dapat dieksekusi (dijalankan).

Bab 3 Teknologi dan Budaya Digital

45

Ayo, Kita Renungkan

Aktivitas refleksi yang dikerjakan di akhir aktivitas, subbab, atau bab pada Informatika Kelas X. Refleksi ini berfungsi untuk merekatkan pengalaman belajarmu dalam ingatan jangka panjang yang akan bermanfaat di kemudian hari.

Pengayaan

Berisi konsep, bahan, dan kegiatan yang dapat dilakukan jika telah mencapai atau menyelesaikan tujuan pembelajaran dan tertarik untuk mempelajari lebih lanjut tentang bab yang telah selesai. Kamu dapat berdiskusi dengan guru tentang hal ini.



Pengayaan

Jika tertarik dengan materi ini dan ingin mempelajari lebih lanjut, kamu dapat mengakses ke link berikut ini.

1. Jobs ID (2020). Info Lowongan Kerja Terbaru dan Populer 2020. Diakses dari <http://jobs.id>
2. Wikipedia (2020). Web Scraping. Diakses dari https://en.wikipedia.org/wiki/Web_scraping
3. Google Colaboratory (2020). Welcome to Colaboratory. Diakses dari <https://colab.research.google.com/>
4. Dataquest (2020). Tutorial: Web Scraping with Python using BeautifulSoup. Diakses dari <https://www.dataquest.io/blog/web-scraping-tutorialpython/>
5. PyData (2021). Pandas: Python Data Analysis Library. Diakses dari <https://pandas.pydata.org/>
6. Plotly (2021). Plotly Python Open Source Graphing Library Basic Chart. Diakses dari <https://plotly.com/python/basic-charts/>
7. Plotly (2021). Plotly | Dash. Diakses dari <https://dash-gallery.plotly.host/dash-web-trader/>



Refleksi

Dalam bab "Data, Informasi, dan Validasinya" ini, kamu telah mempelajari tentang web scraping menggunakan Python, sebuah metode yang memudahkan pengumpulan data dari internet, yang merupakan alternatif dari pencarian manual. Kegiatan ini sejalan dengan konsep, di mana kamu belajar tentang pentingnya mengumpulkan data yang akurat dan efisien. Pertanyaan renungan: Bagaimana teknik web scraping ini membantu dalam mengumpulkan data yang valid dan bagaimana ini berbeda dari pencarian manual yang telah kamu pelajari sebelumnya?

Bab 1 Data, Informasi, dan Validasinya

25

Data, Informasi, dan Validasinya

“
Dari banyak informasi yang dapat kamu peroleh, dari media cetak atau media elektronik, bagaimana menentukan apakah informasi tersebut benar atau tidak?
”





Tujuan Pembelajaran

Setelah mempelajari bab ini, kamu akan mampu melakukan secara efektif pencarian informasi dengan variabel yang lebih banyak, memahami ekosistem untuk memilah fakta dan bukan fakta, menerapkan analisis dalam pengambilan data, membaca lateral serta mengevaluasi validitas data dan sumber datanya.

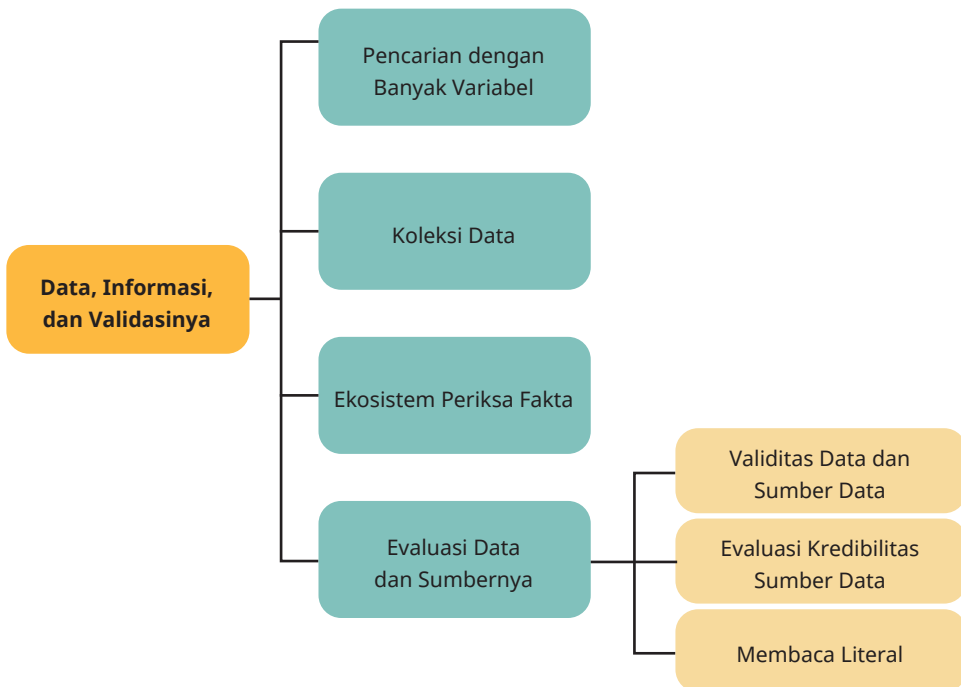


Kata Kunci

Kata kunci pencarian, variabel pencarian, ekosistem, pengecekan, fakta, cek fakta, data, sumber data, koleksi data, valid, *web scraping*, akurat, evaluasi data, membaca lateral.



Peta Konsep



Di jenjang SMP, kamu telah mengeksplorasi dasar-dasar dunia data dan juga informasi digital. Misalnya, saat kamu mencari di buku atau menggunakan Google untuk mencari informasi tentang perubahan iklim, dan kemudian harus membedakan antara artikel ilmiah dan opini pribadi. Di kelas ini, kamu akan mengambil langkah lebih jauh. Kamu akan belajar tidak hanya bagaimana cara mendapatkan data, tetapi juga bagaimana memproses dan menganalisisnya. Sebelum masuk lebih dalam, terdapat beberapa konsep yang dapat didiskusikan untuk membantumu memahami materi. Jawablah beberapa pertanyaan berikut.

1. Dalam melakukan pencarian, terdapat variabel yang dapat digunakan. Apa itu variabel pencarian?
2. Hasil dari suatu pencarian dapat berupa fakta atau bukan fakta. Apa itu fakta?
3. Untuk mendukung sebuah fakta, ada ekosistem yang dapat diikuti. Apa itu ekosistem?
4. Fakta juga perlu didukung dengan data yang valid dan sumber data yang valid. Apa itu valid?
5. Untuk mengetahui sebuah informasi valid atau tidak, kamu dapat melakukan proses membaca lateral. Apa itu membaca lateral?

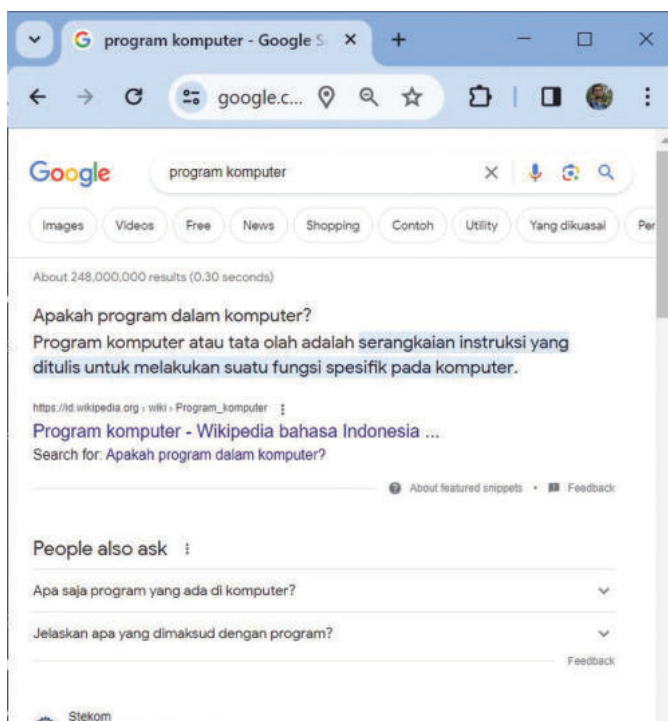
A. Pencarian dengan Banyak Variabel

Pernahkah kamu mengalami saat sedang mencari informasi menemukan bahwa informasinya tersedia di banyak tempat? Misalnya, saat kamu mendapatkan tugas terkait membuat sebuah program komputer. Saat kamu mencari buku terkait program komputer kepada pustakawan atau petugas perpustakaan tanpa penjelasan tambahan lain, mungkin dia akan memberikan banyak sekali buku terkait program dan komputer. Begitu juga jika kamu mencari artikel di internet dengan kata kunci: program komputer, kamu juga akan mendapatkan banyak sekali halaman web yang muncul di hasil pencarian. Kamu harus membuka buku atau halaman web tersebut satu per satu, bahkan mengunduhnya untuk setiap halaman. Kemudian, kamu juga harus menganalisis dan membandingkan untuk mengambil keputusan buku atau halaman web mana yang akan digunakan untuk mengerjakan tugasmu. Membuka dan mengunduh halaman web tersebut harus dilakukan manual. Akan sangat melelahkan, bukan?

Untuk pencarian artikel di internet, kamu sudah mengenal mesin pencari seperti Google yang akan memberikan hasil sesuai kata kunci yang diberikan. Kata kunci adalah kumpulan kata atau frasa yang digunakan untuk menemukan



informasi yang berkaitan dengan suatu topik di dalam basis data, mesin pencari, atau sistem informasi lainnya. Namun demikian, Google bukan hanya alat untuk mencari informasi umum. Ia juga dapat menjadi asisten pribadi yang efektif dalam pencarian suatu informasi spesifik. Mungkin kamu masih menggunakan Google hanya dengan mengetikkan kata kunci sederhana dan mengklik hasil pertama yang muncul. Namun, Google menawarkan fitur pencarian lanjutan yang dapat membantu kamu mencari informasi yang lebih spesifik dan relevan. Sebagai contoh pada Gambar 1.1, terdapat dua kata kunci yang digunakan dalam kontak pencarian sehingga menghasilkan lebih dari dua ratus juta halaman web. Dari jutaan halaman web tersebut, banyak sekali halaman web yang tidak relevan dengan harapanmu dalam mencari informasi untuk tugas membuat program komputer.



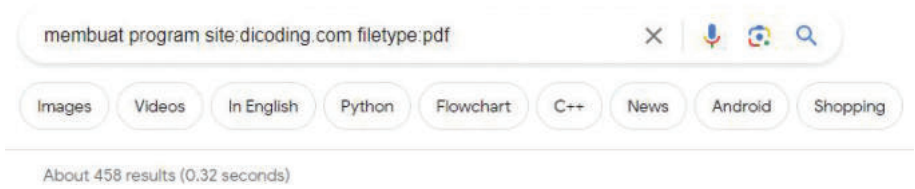
Gambar 1.1 Contoh pemanfaatan kata kunci dalam mesin pencari.

Sumber: Tangkapan layar Google.com.

Agar mendapatkan hasil pencarian yang lebih relevan dan fokus, kamu dapat menggunakan mesin pencari secara lanjut dengan memanfaatkan banyak variabel. Variabel atau parameter adalah kata kunci khusus yang dapat digunakan dalam pencarian lanjut untuk memfokuskan hasil. Sebagai



contoh pada Gambar 1.2, terdapat dua variabel tambahan, yaitu alamat situs yang menjadi fokus pencarian dan tipe *file* yang dicari.



Gambar 1.2 Mesin pencarian dengan banyak variabel.

Sumber: Tangkapan layar Google.com.

Variabel tersebut akan mengecilkan ruang lingkup pencarian hingga mampu mengurangi hasil pencarian. Jika sebelumnya pencarian mendapatkan lebih dari dua ratus juta halaman web, dengan memanfaatkan variabel ini, kamu hanya akan mendapatkan kurang dari lima ratus halaman web. Selain menggunakan operator seperti tanda petik (“), tanda minus (-), dan kata kunci OR, penggunaan variabel akan dapat memudahkanmu untuk memutuskan halaman web mana yang sesuai dengan tugasmu karena hasil yang ditampilkan dan perlu dianalisis lebih sedikit. Detailnya diperlihatkan pada Tabel 1.1 yang mencontohkan penggunaan variabel untuk melakukan pencarian kerja. Kamu dapat mencoba penggunaan variabel tersebut di peramban (*browser*) favoritmu, modifikasi teksnya hingga kamu memahami penggunaannya.

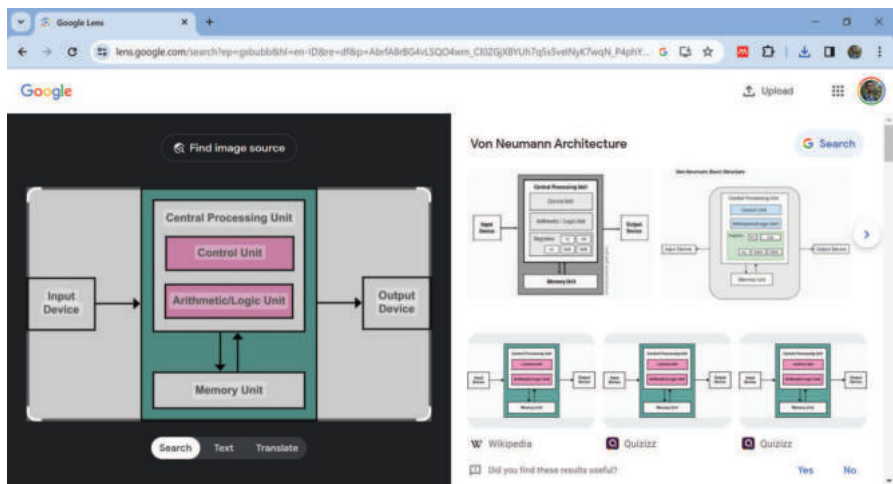
Tabel 1.1 Variabel yang Dapat Digunakan di Mesin Pencari Google

No.	Variabel	Penjelasan	Contoh Penggunaan
1.	site:	Mencari konten yang muncul di alamat (domain) website tertentu.	desainer site:linkedin.com untuk mencari artikel terkait pekerjaan desainer di LinkedIn
2	cache:	Menemukan versi salinan sementara (<i>cache</i>) terbaru dari halaman web tertentu.	programmer cache:jobstreet.com untuk melihat versi <i>cache</i> terbaru dari situs web Jobstreet terkait pekerjaan <i>programmer</i>
3	inanchor:	Mencari halaman dengan teks pranala (<i>anchor</i>) tertentu.	inanchor:alumni untuk mencari <i>anchor</i> dengan teks alumni



No.	Variabel	Penjelasan	Contoh Penggunaan
4	inurl:	Mencari halaman dengan kata kunci tertentu dalam alamat pranala (URL).	<code>inurl:karir</code> untuk mencari halaman yang mengandung URL teks karir
5	intitle:	Mencari halaman dengan judul tertentu.	<code>intitle:lowongan</code> untuk mencari halaman dengan judul yang mengandung teks lowongan
6	intext:	Mencari kata kunci di dalam teks (<i>body</i>) halaman.	<code>intext:"daftar"</code> untuk mencari web yang <i>body</i> halamannya mengandung kata daftar
7	daterange:	Mencari konten yang dipublikasikan dalam rentang tanggal tertentu.	<code>manajer</code> <code>daterange:2022-2023</code> untuk mencari informasi tentang pekerjaan manajer dari tahun 2022 hingga 2023

Selain penggunaan variabel di atas, kamu dapat melakukan pencarian lanjut dengan modal lain selain tulisan teks atau kata kunci. Dengan mesin pencari lanjut saat ini, kamu dapat menyertakan suara dan gambar sehingga hasil pencarian dapat sesuai dengan yang kamu harapkan. Seperti diperlihatkan pada Gambar 1.3, kamu dapat meng-*upload* sebuah gambar di mesin pencari seperti Google Lens untuk mencari gambar yang mirip dengan yang kamu sertakan.



Gambar 1.3 Pencarian dengan menyertakan gambar pada Google Lens.

Sumber: Tangkapan layar lens.google.com.



Aktivitas 10.1-01-P

Mencari dengan Banyak Variabel dalam Mesin Pencari

Dari informasi yang telah disampaikan sebelumnya, berikan kata kunci beserta variabel yang tepat untuk kasus-kasus berikut.

1. Kamu adalah seorang pelajar yang ingin mencari informasi mengenai Software Engineer di situs Medium.com saja.
2. Kamu adalah seorang peneliti yang ingin mencari artikel ilmiah tentang Krisis Iklim yang telah dipublikasikan antara tahun 2018 hingga 2020.
3. Kamu ingin belajar Python untuk menganalisis data dan mencari tutorial yang khusus membahas tentang pustaka pengolahan data ini nama pustaka pemrograman, sudah benar.
4. Kamu ingin mengetahui berita terkini tentang Olimpiade 2024, tetapi tidak ingin melihat berita menceritakan Indonesia.
5. Kamu adalah seorang calon mahasiswa yang ingin mengetahui testimoni dari alumni Universitas Indonesia mengenai program studi Teknik Informatika.



Aktivitas 10.1-02-U

Mencari dengan Banyak Variabel di Perpustakaan

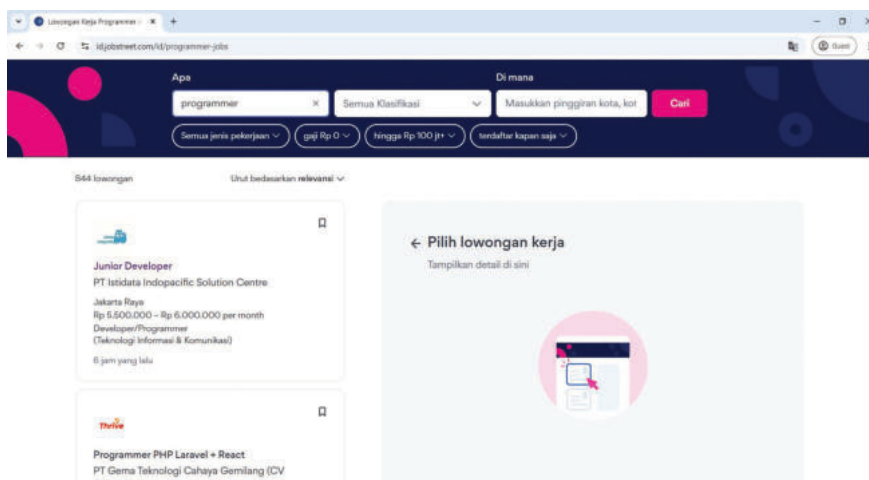
Untuk aktivitas *unplugged*, kamu bisa bermain peran secara berpasangan di perpustakaan sekolah. Orang pertama berperan sebagai ‘mesin’ yang akan mencari buku atau artikel yang ada di perpustakaan. Orang kedua berperan sebagai orang yang akan mencari dengan memanfaatkan bantuan dari ‘mesin’ tersebut. Orang kedua akan memberikan sederet kata kunci beserta variabelnya kepada orang pertama dalam secarik kertas. Orang pertama kemudian harus mencari di setiap rak di perpustakaan sekolah berdasarkan permintaan orang kedua dan mengembalikan dua-tiga buku atau artikel yang sesuai.

Dari aktivitas tersebut, tuliskan kesimpulan terkait bagaimana kamu dapat melakukan pencarian agar hasil pencarian tepat sesuai yang diharapkan (efektif) dengan waktu yang relatif cepat tanpa harus menghasilkan buku-buku yang tidak diperlukan (efisien)!



B. Koleksi Data

Selain langsung mencari dari Google, contoh web lain yang dapat digunakan untuk mencari pekerjaan di Indonesia secara spesifik, yaitu ID JobStreet.com. Ketikkan <https://id.jobstreet.com/> di *browser*. Di halaman web seperti diperlihatkan pada Gambar 1.4, kamu dapat mencari pekerjaan dengan memasukkan kata kunci pekerjaan di formulir pencarian yang ada, misalnya pekerjaan *programmer*.



Gambar 1.4 Contoh tampilan halaman web pencari kerja.

Sumber: Tangkapan layar JobStreet Indonesia.

Dari halaman tersebut, kamu akan memperoleh daftar pekerjaan yang ditampilkan cukup detail dalam beberapa halaman. Dengan tampilan tersebut, kamu perlu melakukan *scroll* sekitar delapan kali untuk mendapatkan sekitar 32 lowongan pekerjaan. Bagaimana jika kamu merangkum lowongan pekerjaan tersebut sehingga lebih mudah untuk melihat dan memilih pekerjaan yang diinginkan?

Dalam kajian analisis data, terdapat teknik yang disebut *scraping*. *Scraping* adalah salah satu bentuk penyalinan, di mana data tertentu dikoleksi dan disalin dari sebuah halaman web, dapat ke dalam basis data, *spreadsheet* atau tampilan tertentu untuk pengambilan atau analisis data. *Scraping* dapat dilakukan dengan menggunakan bahasa pemrograman, salah satunya ialah Python. Dengan *scraping*, kamu bisa mendapatkan rangkuman dari suatu halaman web. Dengan demikian, 32 lowongan yang sebelumnya ditampilkan dalam beberapa *scroll* kini menjadi diringkas dalam satu tampilan dengan data penting yang diinginkan saja.

Dalam aktivitas ini, kamu akan melakukan *scraping* dengan membuat program yang mengambil data dari sebuah halaman website. Program *scraping* ini akan menggunakan bahasa pemrograman Python. Python dipilih karena menjadi bahasa yang sering digunakan dalam analisis data. Python memiliki banyak sekali fungsi dan pustaka (*library*) yang memudahkan untuk melakukan analisis data, salah satunya untuk melakukan *scraping*.

Kemudian, untuk membuat program Python, kamu perlu alat bantu IDE (*Integrated Development Environment*) atau lingkungan kerja untuk menulis dan menjalankan program Python tersebut. Salah satu alat bantu yang dapat kamu gunakan ialah Google Colaboratory atau Google Colab. Google Colab dipilih karena dapat digunakan secara *online* untuk mengambil data dari web yang *online* pula. Selain itu, banyak fungsi dan pustaka Python yang sudah terpasang dalam Google Colab sehingga dapat langsung digunakan.



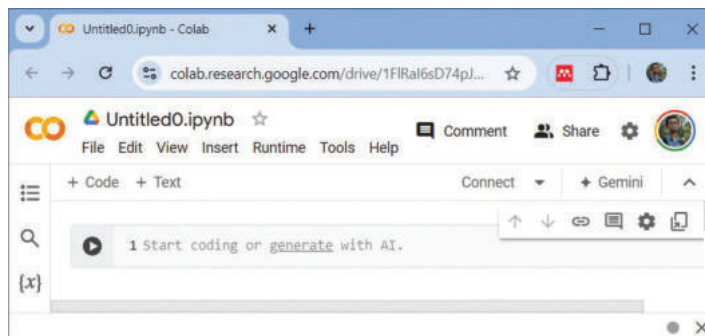
Aktivitas 10.1-03-P

Melakukan Koleksi Data dengan Web Scraping

Dalam aktivitas ini, kamu akan melakukan *scraping* dengan membuat program yang mengoleksi data dari sebuah halaman website lowongan pekerjaan. Berikut beberapa langkah aktivitasnya.

1. Buka Google Colab

Buka aplikasi melalui link <https://colab.research.google.com/>. Jika diminta untuk *sign-in*, silakan masuk menggunakan akun Google/Gmail kamu. Selanjutnya, buka catatan baru melalui menu File > New notebook atau Notebook baru, seperti tampilan di Gambar 1.5.



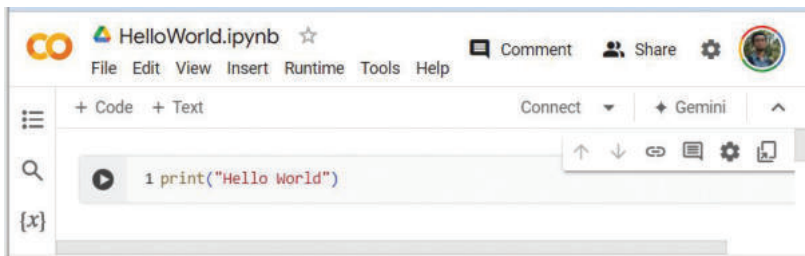
Gambar 1.5 Membuka *file* baru di Google Colab.

Sumber: Tangkapan layar Google Colab.



2. Cek Lingkungan IDE

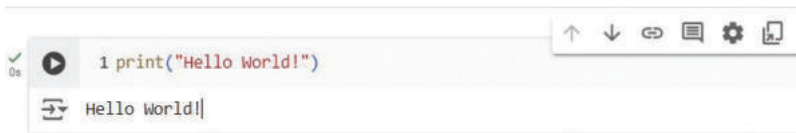
Sebagai awalan mencoba bahasa pemrograman dan mencoba lingkungan pengembangan, program sederhana Hello World dapat dicoba. Ketikkan perintah print (“Hello World”) di area kerja editor Google Colab, kemudian jalankan program dengan menekan tombol segitiga di kiri kode program seperti pada Gambar 1.6.



Gambar 1.6 Program Hello World dengan bahasa Python di Google Colab.

Sumber: Tangkapan layar Google Colab.

Jika tidak ada kendala, program akan menampilkan teks Hello World sebagai luaran dari kode program tersebut (Gambar 1.7).



Gambar 1.7 Luaran Program Hello World

Sumber: Tangkapan layar Google Colab.

3. Parse Halaman Web

Setelah uji editor berhasil dan siap digunakan, *scraping* dapat dimulai dengan proses *parsing*. *Parsing* adalah mengambil kode program dari sebuah halaman web secara utuh untuk kemudian dilakukan analisis. Gambar 1.8 menunjukkan kode untuk melakukan *parsing* pada sebuah alamat website lowongan pekerjaan yang digunakan sebelumnya. Ketik beberapa baris kode program berikut di layar Google Colab kamu (menggantikan baris perintah Hello World sebelumnya).



```

1 import requests
2 import pandas as pd
3 from bs4 import BeautifulSoup
4
5 th = "https://id.jobstreet.com/id/programmer-jobs"
6 halaman = requests.get(th)
7 hasil = BeautifulSoup(halaman.content, 'html.parser')
8 lowkers = hasil.find_all('article')
9 lowkers

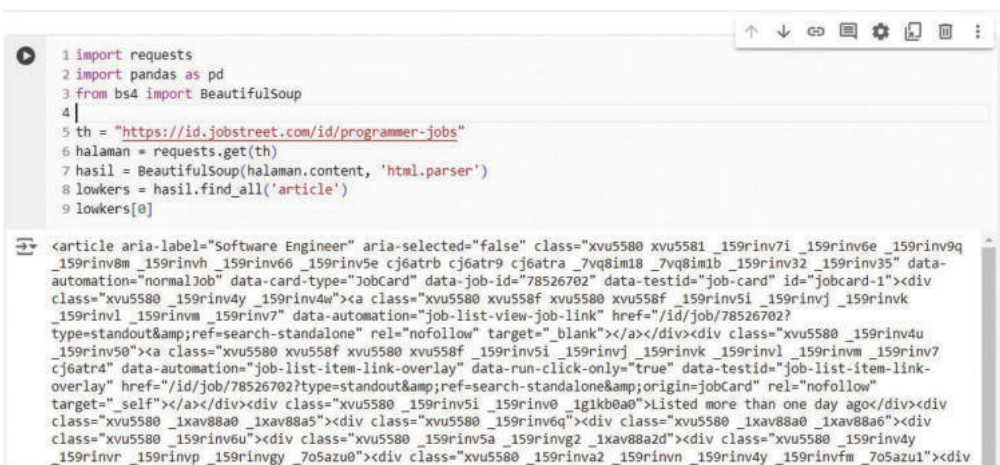
```

Gambar 1.8 Kode program untuk *parsing* sebuah alamat web.

Sumber: Tangkapan layar Google Colab.

Jika dijalankan, kode tersebut akan menyalin kode program yang ada di alamat website yang di-*request* seperti dicontohkan pada Gambar 1.9. Program akan mengambil semua kode web di pranala yang ditunjukkan pada baris-5 (baris nomor 5 seperti yang ditunjukkan pada Gambar 1.8). Sebagai informasi, nomor baris yang ditampilkan tersebut hanya digunakan agar penjelasan ini dapat mengacu ke baris yang sesuai pada gambar. Penomoran yang muncul di IDE kamu mungkin akan berbeda atau bahkan tanpa nomor baris, jadi kamu dapat menyesuaikan.

Kembali ke baris-5, kode web tersebut memiliki bagian atau elemen yang disebut sebagai *tag* HTML. Setelah dianalisis, informasi mengenai lowongan pekerjaan di web *JobStreet* tersedia dalam kelas “columns opportunity” sehingga *tag* tersebut yang harus kamu ambil (baris-9).



```

1 import requests
2 import pandas as pd
3 from bs4 import BeautifulSoup
4
5 th = "https://id.jobstreet.com/id/programmer-jobs"
6 halaman = requests.get(th)
7 hasil = BeautifulSoup(halaman.content, 'html.parser')
8 lowkers = hasil.find_all('article')
9 lowkers[0]

```

<article aria-label="Software Engineer" aria-selected="false" class="xvu5580 xvu5581 _159rin7i _159rin7e _159rin7q
 _159rin7m _159rin7h _159rin76 _159rin75e c66atrb c66at9 c66atra 7vq8im18 7vq8im1b _159rin732 _159rin735" data-
 automation="normalJob" data-card-type="JobCard" data-job-id="78526702" data-testid="job-card" id="jobcard-1"><div
 class="xvu5580 _159rin74y _159rin74w"><a class="xvu5580 xvu558f xvu5580 xvu558f _159rin75i _159rin7j _159rin7k
 _159rin7l _159rin7m _159rin77" data-automation="job-list-view-job-link" href="/id/job/78526702?
 type=standout&ref=search-standalone" rel="nofollow" target="_blank"></div><div class="xvu5580 _159rin74u
 _159rin750"><a class="xvu5580 xvu558f xvu5580 xvu558f _159rin75i _159rin7j _159rin7k _159rin7l _159rin7m _159rin7
 c66at9" data-automation="job-list-item-link-overlay" data-run-click-only="true" data-testid="job-list-item-link-
 overlay" href="/id/job/78526702?type=standout&ref=search-standalone&origin=jobCard" rel="nofollow"
 target="_self"></div><div class="xvu5580 _159rin75i _159rin70 _1g1kb0a0">listed more than one day ago</div><div
 class="xvu5580 _1xav88a0 _1xav88a5"><div class="xvu5580 _159rin76q"><div class="xvu5580 _1xav88a0 _1xav88a6"><div
 class="xvu5580 _159rin76u"><div class="xvu5580 _159rin75a _159rin72 _1xav88a2d"><div class="xvu5580 _159rin74y
 _159rin7v _159rin7p _159rin7y _7o5azu0"><div class="xvu5580 _159rin7a2 _159rin7n _159rin74y _159rin7fm _7o5azu1"><div

Gambar 1.9 Luaran program halaman web pencari kerja.

Sumber: Tangkapan layar Google Colab.




```

13 8 len(lowkers)
64
[ ] 1 posisi = []
    2 instansi = []
    3 lokasi = []
    4 gaji = []
    5
    6 for job_card in lowkers:
    7     tPos = job_card.find('a', attrs={'data-automation': 'jobTitle'}).text.strip()
    8     posisi.append(tPos)
    9
    10    eIns = job_card.find('a', attrs={'data-automation': 'jobCompany'})
    11    tIns = eIns.text.strip() if eIns else 'No info'
    12    instansi.append(tIns)
    13
    14    eLok = job_card.find('span', attrs={'data-automation': 'jobCardLocation'})
    15    tLok = eLok.text.strip() if eLok else 'No info'
    16    lokasi.append(tLok)
    17
    18    eGaji = job_card.find('span', attrs={'data-automation': 'jobSalary'})
    19    tGaji = eGaji.text.strip() if eGaji else 'No info'
    20    gaji.append(tGaji)
    21
    22 print(posisi)
    23 print(instansi)
    24 print(lokasi)
    25 print(gaji)

['Programmer Back End Dan Front End', 'Junior Full Stack Developer (URGENT)', 'Software Engine
f'PT DIBUILTADI TEKNOLOGI KREATIF', 'PT Peruri Digital Security', 'PT Asuransi Wahana Tata', '

```

Gambar 1.11 Kode program untuk ambil detail lowongan pekerjaan.

Sumber: Tangkapan layar Google Colab.

6. Tampilkan Data

Terakhir, bingkai *data array* tersebut ke dalam tabel yang ditampilkan oleh Python seperti diperlihatkan kode pada Gambar 1.12. Ketikkan kode program baris 1-7, kemudian jalankan sehingga menghasilkan luaran seperti pada tabel di bawahnya.

```

[12] ['Semarang', 'Jakarta Barat', 'Surabaya', 'Tangerang', 'Jakarta Pusat', 'Jakarta Barat', 'Jakarta Utara', 'Kendal', 'Jakarta Raya', 'Jakarta UT
      'Rp\xa3.500.000 - Rp\xa84.000.000 per month', 'No info', 'No info', 'No info', 'No info', 'Rp\xa85.000.000 - Rp\xa85.500.000 per month', 'Rp\

```

```

1 lowker = pd.DataFrame({
2     "Posisi": posisi,
3     "Instansi": instansi,
4     "Lokasi": lokasi,
5     "gaji": gaji,
6 })
7 lowker

```

	Posisi	Instansi	Lokasi	gaji
0	Mobile Developer	PT Artha Mulia Trijaya	Semarang	Rp 3 500 000 - Rp 4 000 000 per month
1	Web Developer	No info	Jakarta Barat	No info
2	PROGRAMMER	PT Global Terminal Marunda	Surabaya	No info
3	Programmer IT	Creazy studio	Tangerang	No info
4	Developer and Operation (ERP/SYSTEM DEVELOPMEN...	Hand International Pte. Ltd.	Jakarta Pusat	No info
...	...	...	...	...
59	Full Stack Software Developer	PT Ecogreen Oleochemicals	Batam	No info
60	Full Programmer	PT Gunatronkatama Cipta (GASI)	Jakarta Selatan	Rp 14,164,387 - Rp 21,246,561 per month
61	IT Mnager - CRM & Web Master (Sales Force)	PT Mirah Global Investment	Badung	Rp 10.000.000 - Rp 15.000.000 per month
62	Web Mobile Programmer	PT Micronet Gigatech Indoglobal	Surabaya	No info
63	IT Web Developer Intern	sns.ac	Tangerang Selatan	No info

64 rows x 4 columns

Gambar 1.12 Luaran kode program yang menghasilkan tabel rekapitulasi data.

Sumber: Tangkapan layar Google Colab.



Gambar 1.12 menunjukkan hasil akhir proses *scraping* data dari web *JobStreet* untuk mengoleksi data lowongan pekerjaan yang meringkasnya ke dalam sebuah tabel. Bagaimana komentarmu, membaca lowongan pekerjaan dari Gambar 1.4 sebagai halaman web asli, dan Gambar 1.12 sebagai hasil *scraping*? Dengan penyajian seperti pada Gambar 1.12, data yang diperoleh akan lebih mudah dipahami dan kemudian diolah/dianalisis. Catatan bahwa tampilan tersebut merupakan data saat dibuat. Bisa jadi, keluaran tidak sama persis, bahkan mungkin juga terjadi *error*.

Langkah-langkah *scraping* menggunakan bahasa Python seperti ini dapat dipelajari secara lebih detail dari tutorial yang ada di internet seperti misalnya di <https://buku.kemdikbud.go.id/s/rnsiwu>. *Scraping* juga dapat dilakukan menggunakan bahasa lain (PHP atau R-Language) dan IDE lain.



Aktivitas 10.1-04-U Koleksi Data Manual

Untuk aktivitas *unplugged*, kamu dapat mengumpulkan data melalui survei dengan kuesioner singkat kepada orang-orang di sekitarmu. Kuesioner dapat menanyakan terkait pekerjaan-pekerjaan yang mungkin dapat mereka tawarkan. Tanyakan terkait posisi atau jabatan, instansi atau organisasi penyedia pekerjaan, lokasi atau alamat, serta gaji. Selanjutnya, kumpulkan jawaban mereka dalam sebuah tabel dengan kolom-kolom tersebut!



Aktivitas 10.1-05-P Mengumpulkan Data dengan Teknologi

Tujuan:

Kamu akan belajar bagaimana membuat dan mendistribusikan survei daring untuk mengumpulkan data, serta menganalisis hasil daring tersebut menggunakan teknologi seperti Google Forms, Microsoft Forms, atau SurveyMonkey.



Deskripsi Aktivitas:

1. **Pendahuluan:** Guru akan menjelaskan pentingnya pengumpulan data dalam penelitian dan bagaimana teknologi dapat mempermudah proses ini. Kamu akan diperkenalkan dengan berbagai alat survei daring seperti Google Forms, Microsoft Forms, dan SurveyMonkey.
2. **Persiapan:** Kamu akan dibagi ke dalam kelompok kecil. Setiap kelompok akan memilih topik survei yang relevan dengan minat kelompokmu atau sesuai dengan tema yang diberikan oleh guru. Topik survei dapat berkaitan dengan kehidupan sehari-hari, lingkungan sekolah, atau isu-isu sosial.
3. **Pembuatan Survei:** Setiap kelompok akan merancang kuesioner menggunakan salah satu alat survei daring. Kamu belajar tentang berbagai jenis pertanyaan seperti pilihan ganda, isian singkat, skala Likert, dan lain sebagainya. Guru akan memberikan panduan tentang cara merumuskan pertanyaan yang jelas dan tidak bias.
4. **Distribusi Survei:** Setelah survei selesai dibuat, kamu akan mendistribusikannya kepada responden yang sesuai, seperti teman sekelas, guru, atau anggota keluarga. Kamu akan mempelajari cara membagikan tautan survei dan mengumpulkan respons dari berbagai platform.
5. **Pengumpulan dan Analisis Data:** Kamu akan mengumpulkan data dari respons yang masuk. Dengan bantuan guru, kalian akan menganalisis data menggunakan fitur-fitur yang tersedia di alat survei daring. Kamu belajar cara membaca grafik dan tabel, serta menarik kesimpulan dari data yang telah dikumpulkan.
6. **Presentasi Hasil:** Setiap kelompok akan membuat presentasi singkat tentang temuan survei kelompoknya. Presentasi harus mencakup tujuan survei, metode yang digunakan, hasil utama, dan kesimpulan. Kamu akan mempresentasikan hasil kelompokmu di depan kelas dan menerima umpan balik dari guru dan teman-teman sekelas.

Hasil yang Diharapkan:

- Kamu mampu membuat kuesioner survei yang efektif.
- Kamu dapat menggunakan alat survei *online* untuk mendistribusikan dan mengumpulkan data.



- Kamu dapat menganalisis data yang dikumpulkan dan menyajikan hasilnya secara efektif.
- Kamu belajar bekerja sama dalam kelompok dan mengembangkan keterampilan komunikasi anggota kelompokmu.

Alat dan Sumber Daya:

- Komputer atau perangkat dengan akses internet.
- Akun Google atau Microsoft untuk menggunakan Google Forms atau Microsoft Forms.
- Akses ke SurveyMonkey atau alat survei daring lainnya (opsional).

C. Ekosistem Periksa Fakta

Setelah mempelajari dan berlatih melakukan pencarian informasi dengan berbagai variabel baik melalui Google ataupun simulasi lain, kamu akan beralih ke topik lain yang sangat relevan: Sistem Periksa Fakta. Mengapa ini penting? Karena dalam era informasi yang serba melimpah ini, kemampuan untuk memeriksa benar atau tidaknya suatu informasi ialah keterampilan yang sangat berharga. Tidak semua informasi yang kamu temukan dalam pencarian merupakan informasi yang benar atau dapat dipercaya.

Terdapat fenomena dalam jurnalisme pemeriksaan fakta sebagai tren baru dalam jurnalisme digital, yaitu memeriksa kebenaran informasi atau fakta (Nurlatifah & Irwansyah, 2019). Ingatkah kamu bahwa fakta adalah informasi yang benar dan dapat dibuktikan dengan bukti atau data yang objektif? Jurnalis bekerja sama dengan banyak orang, termasuk pengguna internet, untuk memastikan dan membuktikan bahwa informasi yang diberikan benar dan lengkap. Artikel ini juga menjelaskan bahwa dalam memeriksa fakta, yang penting ialah bagaimana cerita itu diceritakan, bukan hanya kutipan dari orang-orang. Selain itu, jenis jurnalisme ini sering membahas masalah yang sedang hangat seperti isu politik dan publik, rumor dan hoaks, serta topik khusus, kontroversi, dan berbagai konflik.

Ekosistem periksa fakta (*fact-checking*) ialah sebuah rangkaian proses dan entitas yang terlibat dalam verifikasi informasi untuk memastikan kebenarannya. Ini merupakan bagian penting dari jurnalisme, penelitian, dan



diseminasi informasi yang bertujuan untuk memerangi penyebaran informasi yang salah atau menyesatkan. Berikut ini beberapa komponen utama dalam ekosistem periksa fakta.

1. Organisasi Periksa Fakta: Entitas ini berfokus dalam memeriksa klaim atau informasi dan melaporkan kebenarannya, contohnya Snopes dan FactCheck.org di luar negeri, atau [CekFakta.com](https://cekfakta.com) dan [Turnbackhoax.id](https://turnbackhoax.id) di Indonesia.
2. Jurnalis dan Peneliti: Mereka sering kali melakukan periksa fakta sebagai bagian dari pekerjaan mereka, seperti cekfakta.kompas.com, cekfakta.tempo.co, liputan6.com/cek-fakta.
3. Platform Media Sosial: Platform seperti Facebook, Twitter, dan Google sering kali berkolaborasi dengan organisasi periksa fakta untuk mengidentifikasi dan menandai informasi yang salah atau menyesatkan seperti *Meta's Third-Party Fact-Checking Program* (Facebook, 2023).
4. Teknologi Kecerdasan Buatan: Teknologi *Artificial Intelligence* (AI) dapat digunakan untuk membantu dalam proses periksa fakta, misalnya untuk mengidentifikasi berita palsu atau informasi yang menyesatkan.
5. Komunitas Online: Ada banyak forum dan grup di media sosial di mana anggota komunitas berbagi dan memeriksa informasi seperti grup Whatsapp atau Telegram.
6. Pendidikan Media: Beberapa program pendidikan yang mengajarkan keterampilan periksa fakta dan literasi media juga merupakan bagian dari ekosistem ini.
7. Regulasi dan Kebijakan: Beberapa negara termasuk Indonesia memiliki regulasi yang dirancang untuk memerangi penyebaran informasi palsu. Sebagai contoh Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik, termuat pada Pasal 45A ayat (1).
8. Publik: Masyarakat umum juga berperan penting dalam ekosistem periksa fakta. Makin banyak orang yang terdidik tentang bagaimana memeriksa fakta, makin efektif pula upaya untuk memerangi informasi palsu.





Aktivitas 10.1-06-U

Mendiskusikan Ekosistem Periksa Fakta

Dari artikel berjudul “Tips Menghadapi Infodemi yang Tak Kalah Ngeri dari Pandemi” (Detik.com, 2020), diskusikan beberapa pertanyaan berikut!

1. Bagaimana infodemi dapat lebih berbahaya daripada pandemi itu sendiri dalam konteks masyarakat?
2. Apakah organisasi pemeriksa fakta cukup efektif dalam membantu masyarakat menghadapi infodemi?
3. Bagaimana literasi media dapat membantu individu dalam mengidentifikasi dan menghindari infodemi?

D. Evaluasi Data dan Sumbernya

Setelah mempelajari tentang proses pencarian dan koleksi data, langkah selanjutnya yang tak kalah penting ialah memastikan validitas dari data dan sumber informasi yang telah kamu kumpulkan. Mengapa? Karena data yang tidak valid dapat menyesatkan dan berakibat fatal, terutama saat digunakan dalam konteks akademis atau pengambilan keputusan. Oleh karena itu, evaluasi baik data maupun sumber datanya merupakan langkah krusial yang harus dilakukan.

1. Validitas Data dan Sumber Data

Data yang valid adalah data atau fakta yang benar-benar akurat dan dapat dipercaya. Misalnya, jika kamu sedang mengerjakan tugas sejarah tentang Revolusi Prancis, data yang valid dapat berupa tanggal-tanggal penting, peristiwa, atau tokoh yang terlibat, yang kamu dapatkan dari buku sejarah atau jurnal ilmiah. Sumber data yang valid ialah tempat kamu mendapatkan data tersebut, dan sumber ini harus kredibel atau dapat dipercaya. Contohnya, untuk tugas sejarah tadi, sumber data yang valid dapat berupa buku teks sejarah yang ditulis oleh sejarawan terkenal atau artikel dari jurnal ilmiah yang telah direviu oleh para ahli.



Saat membuat tugas, karya tulis atau proyek penelitian, kamu membutuhkan keterampilan untuk mengevaluasi kevalidan sumber dan data. Ini penting agar informasi yang digunakan benar-benar dapat dipertanggungjawabkan dan tidak menyesatkan. Dalam membuat berbagai laporan tugas dan makalah penelitian, diperlukan keterampilan literasi informasi tingkat tinggi seperti menemukan berbagai jenis sumber, mengevaluasi validitas sumber tersebut, dan menganalisis secara kritis konten dari sumber-sumber tersebut (Rivoltela, 2008).

2. Evaluasi Kredibilitas Sumber Data

Mengevaluasi kredibilitas suatu sumber ialah cara penting untuk menyaring informasi yang salah dan menentukan apakah kamu harus menggunakannya dalam penelitianmu. Salah satu metode yang dapat digunakan termasuk tes CRAAP (Ryan, 2023). CRAAP adalah pendekatan untuk mengevaluasi dengan menanyakan kredibilitas suatu sumber data. CRAAP kepanjangan istilah berikut.

- *Currency* atau keaktualan
Apakah sumber tersebut mencerminkan penelitian yang baru-baru ini dilakukan?
- *Relevance* atau relevansi
Apakah sumber tersebut berkaitan dengan topik kajian yang sedang dilakukan?
- *Authority* atau otoritas
Apakah ini merupakan publikasi atau artikel yang bereputasi, berasal dari komunitas akademis atau profesional di bidangnya?
- *Accuracy* atau keakuratan
Apakah sumber tersebut mendukung argumen dan kesimpulannya dengan bukti yang logis?
- *Purpose* atau tujuan
Apa tujuan atau konteks penulis menulis publikasi atau artikel tersebut?



3. Membaca Lateral

Membaca lateral merupakan strategi yang digunakan untuk mengevaluasi kualitas informasi yang diperoleh secara *online*. Membaca secara lateral berarti ketika kamu menemukan informasi atau berita di internet, kamu tidak langsung percaya. Alih-alih, kamu akan keluar dari situs web tersebut untuk mencari informasi lebih lanjut (Brodsky dkk, 2021). Tujuannya ialah untuk memastikan apakah informasi itu benar atau tidak. Beberapa cara dapat dilakukan dalam membaca lateral. Pertama, kamu perlu mencari sumber asli dari informasi atau klaim yang dibuat. Kedua, kamu juga perlu mengetahui lebih banyak tentang orang atau organisasi yang memberikan informasi tersebut. Terakhir, kamu juga harus memeriksa kebenaran informasi dengan menggunakan situs web yang memeriksa fakta, atau mencari informasi lain di internet yang bereputasi.

Misalkan, kamu menemukan sebuah postingan di media sosial yang mengklaim bahwa vaksinasi COVID-19 dapat menyebabkan efek samping yang sangat serius. Postingan tersebut menunjukkan angka dan data yang tampaknya meyakinkan. Dengan melakukan langkah-langkah membaca lateral, kamu dapat mengikuti alur berikut ini. Sebelum percaya dan meneruskan informasi tersebut, kamu memutuskan untuk memeriksa kebenarannya. Kamu mencari informasi dari sumber resmi seperti situs Kementerian Kesehatan atau Organisasi Kesehatan Dunia (WHO). Selanjutnya, kamu akan membandingkan data efek samping yang disebutkan di kiriman dengan data resmi. Apakah angkanya mirip? Apakah efek samping yang disebutkan memang benar? Kamu juga mencari tahu siapa yang membuat kiriman tersebut. Apakah dia seorang ahli medis atau hanya seseorang yang tidak memiliki latar belakang di bidang kesehatan? Terakhir, kamu juga dapat mencari artikel atau penelitian lain yang membahas efek samping vaksinasi untuk mendapatkan gambaran yang lebih lengkap.

Jika setelah melakukan semua itu, kamu menemukan bahwa informasi dalam kiriman tersebut ternyata tidak akurat dan cenderung menyesatkan, kamu tidak hanya menghindari disinformasi, tetapi juga menjadi lebih kaya dengan lebih banyak informasi untuk membuat keputusan. Dalam hal ini, keputusan untuk divaksin yang tentunya akan lebih menguntungkanmu secara nyata.





Aktivitas 10.1-07-U

Mengevaluasi Data dan Sumber Data

Kamu menemukan sebuah pesan layanan pendek dari nomor tidak dikenal +52 889 160 6441 sebagai berikut.

Hallo, nama saya Malia dan saya bekerja di departemen SDM di <<sensor>> Cabang Indonesia. Perusahaan ini mencari karyawan paruh waktu secara *online*.

Kerja paruh waktu itu mudah, yang Anda butuhkan hanyalah ponsel. 10 hingga 20 menit kerja paruh waktu! Hal ini bisa dilakukan di waktu senggang Anda tanpa mengganggu pekerjaan Anda saat ini. Pendaftar baru langsung mendapatkan Rp20.000, gaji harian: Rp200.000 - 2.000.000. Jika Anda tertarik dengan pekerjaan paruh waktu ini, balas kami dengan klik tautan WhatsApp berikut:

Bandingkan dengan pesan yang kamu peroleh dari nomor yang memang dideteksi berasal dari gurumu sebagai berikut.

Selamat pagi <<menyebutkan namamu>>

Ada lowongan pekerjaan paruh waktu dari rekan alumni di <<sensor>>. Pekerjaannya mengolah beberapa data terkait hasil survei pengguna yang mereka lakukan di sekolah. Mungkin akan mengganggu waktu belajarmu, tapi jika diambil, kamu akan mendapatkan pengalaman yang bermanfaat. Untuk diskusi lebih lanjut, kamu dapat menemui saya di Ruang Bersama Guru saat istirahat siang.

Bagaimana, kamu tertarik?

Bandingkan kedua pesan di atas.

Jelaskan bagaimana cara untuk menentukan apakah pesan-pesan tersebut valid. Apakah sumber dari pesan tersebut valid?



Pilih satu jawaban yang paling tepat.

1. Mengapa mesin pencari seperti Google dapat dianggap sebagai “asisten pribadi” dalam konteks pencarian pekerjaan? Karena mesin ini . . .
 - A. mampu menampilkan iklan pekerjaan berdasarkan riwayat pencarian pengguna
 - B. mampu merangkum lowongan pekerjaan dari berbagai sumber ke dalam satu panel
 - C. mengizinkan pengguna untuk langsung mengajukan lamaran melalui platformnya
 - D. menawarkan fitur pengingat untuk membantu pengguna melacak lowongan pekerjaan
 - E. mampu menampilkan iklan pekerjaan apa pun
2. Apa yang harus dipertimbangkan saat menggunakan mesin pencari untuk informasi penting seperti pencarian pekerjaan?
 - A. Kecepatan pencarian informasi
 - B. Kemudahan navigasi antarhalaman hasil pencarian
 - C. Keakuratan dan keandalan informasi yang disajikan
 - D. Jumlah sumber informasi yang diakses oleh mesin pencari
 - E. Page Rank dari hasil pencarian
3. Seorang peneliti ingin mencari jurnal ilmiah tentang perubahan iklim yang hanya dipublikasikan oleh universitas terkemuka dan dari suatu dokumen. Kata kunci beserta variabel pencarian Google mana yang paling tepat digunakan untuk mencapai tujuan ini?
 - A. perubahan iklim filetype:doc
 - B. perubahan iklim site:edu filetype:”doc”
 - C. “perubahan iklim” site:ac.id filetype:pdf
 - D. perubahan iklim OR site:”ac.id” filetype:pdf
 - E. perubahan iklim filetype:ppt



4. Dalam konteks jurnalisme digital, mengapa periksa fakta dianggap lebih penting daripada era jurnalisme tradisional? Karena jurnalisme digital . . .
 - A. lebih cepat dalam menyebarkan informasi
 - B. lebih mengutamakan opini daripada fakta
 - C. memungkinkan lebih banyak orang untuk menjadi jurnalis tanpa pelatihan formal
 - D. sudah memiliki mekanisme periksa fakta yang efektif
 - E. memungkinkan hanya penerbit yang mempublikasikan konten
5. Sebuah organisasi periksa fakta menemukan bahwa sebuah klaim mengenai efek samping dari vaksin Covid-19 adalah palsu. Namun, informasi ini tetap menyebar luas di media sosial. Komponen ekosistem periksa fakta manakah yang paling berpotensi gagal dalam situasi ini?
 - A. Organisasi Periksa Fakta
 - B. Jurnalis dan Peneliti
 - C. Platform Media Sosial
 - D. Publik
 - E. Penerbit Digital
6. Salah satu cara untuk meningkatkan kemampuan membaca lateral ialah dengan . . .
 - A. memilih hanya satu sumber informasi yang dianggap terpercaya
 - B. membatasi diri hanya pada satu topik pembelajaran
 - C. menghindari pertanyaan yang menantang
 - D. membaca dari berbagai sumber yang berbeda dan mengasumsikan semuanya benar
 - E. tidak peduli dengan latar belakang atau kepercayaan dari setiap sumber informasi
7. Seorang peneliti menemukan sebuah jurnal yang mengklaim memiliki penemuan revolusioner tentang perubahan iklim. Jurnal tersebut belum melewati proses peninjauan. Tindakan apa yang paling tepat untuk peneliti tersebut?
 - A. Langsung mengutip jurnal tersebut dalam penelitiannya.
 - B. Mengabaikan jurnal tersebut karena belum ditinjau.
 - C. Melakukan validasi lebih lanjut sebelum mengutip jurnal tersebut.
 - D. Menghubungi penulis jurnal untuk meminta data mentah.
 - E. Menghubungi penulis jurnal meminta data hasil olahan.



8. Seorang mahasiswa menemukan informasi dari sebuah blog yang mengatakan bahwa vaksinasi dapat menimbulkan efek samping serius. Bagaimana dia harus mengevaluasi informasi tersebut?
- A. Menerima informasi tersebut karena penulisnya ialah seorang jurnalis terkenal.
 - B. Menggunakan tes CRAAP untuk mengevaluasi kredibilitas blog tersebut.
 - C. Mengutip informasi tersebut dalam tugas akademisnya tanpa memeriksa lebih lanjut.
 - D. Mengandalkan blog tersebut sebagai satu-satunya sumber informasi karena ditulis dengan bahasa yang meyakinkan.
 - E. Menerima informasi blog tersebut karena berasal dari platform terkenal.
9. Ketika melakukan membaca lateral, apa yang paling penting untuk diperhatikan?
- A. Jumlah pengikut atau *subscribers* dari sumber informasi.
 - B. Kecepatan akses ke situs web sumber informasi.
 - C. Otoritas dan kredibilitas sumber informasi alternatif.
 - D. Desain dan tata letak dari situs web sumber informasi.
 - E. Kemampuan perangkat untuk berpindah layar.
10. Apa yang paling perlu dilakukan saat menemukan data yang kontradiktif dari dua sumber yang sama-sama kredibel?
- A. Memilih sumber yang lebih sesuai dengan kepercayaan pribadi.
 - B. Mengabaikan kedua sumber dan mencari sumber ketiga sebagai *tiebreaker*.
 - C. Memeriksa metodologi dan konteks di balik data dari kedua sumber.
 - D. Mengutip kedua sumber dan menyatakan bahwa lebih banyak penelitian diperlukan.
 - E. Memeriksa penulis dari kedua sumber dan menentukan sumber berdasarkan popularitas penulisnya.





Pengayaan

Jika tertarik dengan materi ini dan ingin mempelajari lebih lanjut, kamu dapat mengakses ke link berikut ini.

1. Jobs ID (2020). Info Lowongan Kerja Terbaru dan Populer 2020. Diakses dari <http://jobs.id>
2. Wikipedia (2020). Web Scraping. Diakses dari https://en.wikipedia.org/wiki/Web_scraping
3. Google Colaboratory (2020). Welcome to Colaboratory. Diakses dari <https://colab.research.google.com/>
4. Dataquest (2020). Tutorial: Web Scraping with Python using BeautifulSoup. Diakses dari <https://www.dataquest.io/blog/web-scraping-tutorialpython/>
5. PyData (2021). Pandas: Python Data Analysis Library. Diakses dari <https://pandas.pydata.org/>
6. Plotly (2021). Plotly Python Open Source Graphing Library Basic Chart. Diakses dari <https://plotly.com/python/basic-charts/>
7. Plotly (2021). Plotly | Dash. Diakses dari <https://dash-gallery.plotly.host/dash-web-trader/>



Refleksi

Dalam bab “Data, Informasi, dan Validasinya” ini, kamu telah mempelajari tentang *web scraping* menggunakan Python, sebuah metode yang memudahkan pengumpulan data dari internet, yang merupakan alternatif dari pencarian manual. Kegiatan ini sejalan dengan konsep, di mana kamu belajar tentang pentingnya mengumpulkan data yang akurat dan efisien. Pertanyaan renungan: Bagaimana teknik *web scraping* ini membantu dalam mengumpulkan data yang valid dan bagaimana ini berbeda dari pencarian manual yang telah kamu pelajari sebelumnya?



Ketika berbicara tentang data yang tidak publik, seperti *email* pribadi atau informasi dari halaman media sosial yang terbatas, pertanyaan etis muncul: Apakah kamu boleh mengambil data tersebut?

Mengenai *tools*, kamu telah menggunakan *library* Python untuk *scraping*. Bagaimana menurutmu cara kerja program *scraper* ini dalam mengambil isi halaman web? Python menawarkan berbagai *library* lain untuk analisis data. Eksplorasi lebih lanjut tentang *library* apa lagi yang tersedia dan bagaimana ini dapat mendukung proses analisis data.

Dalam konteks visualisasi data *real-time*, seperti sistem pelacakan ojek *online* atau tampilan informasi di bandara, kamu telah melihat bagaimana data yang berubah-ubah ditampilkan secara aktual. Ini mengajarkan tentang pentingnya menyajikan data yang *up-to-date*, yang sangat relevan dalam konteks data, informasi, dan validasinya. Jika ingin belajar lebih jauh tentang sistem *dashboard*, kamu dapat mempelajari website <https://dash-gallery.plotly.host/dash-web-trader/>. Bagaimana menurutmu peran visualisasi data dalam menyajikan informasi yang valid dan terkini? Buatlah infografis yang menggambarkan kerja sama antara berbagai agen pengolah data dalam sistem ini.

Terakhir, refleksi tentang proyek *web scraping* kamu. Ini hanya sebagian dari proses analisis data yang lebih besar, yang mencakup pengambilan, pemrosesan, presentasi, analisis, dan pengambilan keputusan. Diskusikan, proses mana yang dapat diotomasi dan mana yang memerlukan intervensi manusia. Apakah otomatisasi selalu lebih menguntungkan?



Algoritma dan Struktur Data

“Bagaimanakah caranya kamu dapat merancang dan menulis instruksi/program yang membuat komputer dapat melakukan pemrosesan yang diperlukan untuk menyelesaikan permasalahan yang diberikan?”





Tujuan Pembelajaran

Setelah mempelajari bab ini, kamu akan mampu membuat solusi rancangan program sederhana dalam bentuk pseudokode dengan menerapkan struktur data standar dengan himpunan data berstruktur kompleks.

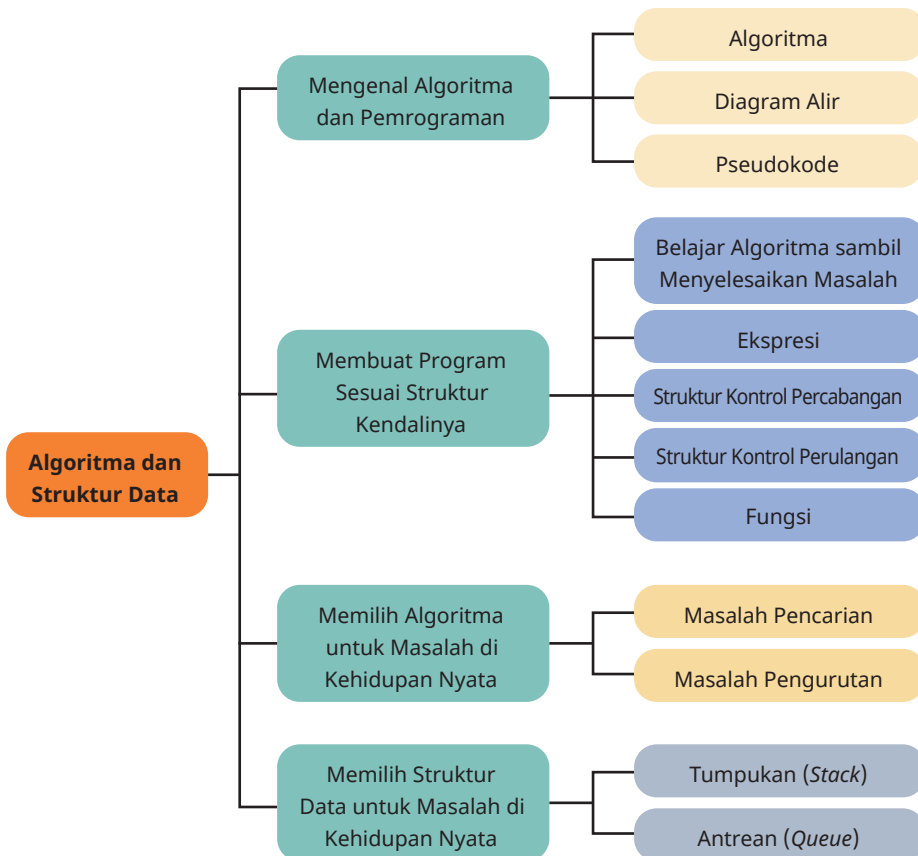


Kata Kunci

Algoritma, diagram alir, ekspresi, percabangan, perulangan.



Peta Konsep



Pada bab ini, kamu akan mengenal tentang algoritma dan struktur data. Dalam mata pelajaran Matematika, kamu telah mengenali adanya prosedur yang dapat digunakan untuk menghitung luas dan volume suatu bangun ruang dengan cara yang terstruktur dan menghasilkan keluaran yang benar. Dalam kehidupan, kamu menghasilkan dan mengolah banyak data, misalnya nilai-nilai di sekolah, pengeluaran belanja harianmu, sampai barang-barang yang kamu miliki. Agar dapat bermanfaat, data tersebut perlu disimpan dan diolah secara terstruktur. Pada bab ini, kamu akan mengenali konsep-konsep dasar algoritma dan struktur data yang dapat digunakan untuk menyimpan dan mengolah data tersebut menjadi suatu informasi yang bermanfaat bagi manusia. Dengan hal ini, kamu bahkan dapat membuat program komputer yang dapat melakukan proses tersebut secara cepat dan otomatis. Jangan khawatir jika kamu tidak memiliki komputer karena sebagian proses pemrograman tidak perlu menggunakan komputer.

Sebelum kamu bertualang dalam pembelajaran kali ini, jawablah dulu beberapa pertanyaan berikut.

1. Apa itu algoritma?
2. Unsur-unsur apa saja yang ada dalam sebuah algoritma?

A. Mengetahui Algoritma dan Pemrograman

Pada bagian ini, kamu akan mengenal lebih jauh tentang algoritma. Setelah itu, kamu akan belajar bagaimana diagram alir akan membentuk suatu algoritma. Akhirnya, kamu akan mengenal pseudocode, kode semu yang akan melengkapi kekurangan pada suatu algoritma.

1. Algoritma

Algoritma adalah suatu kumpulan instruksi terstruktur dan terbatas yang dapat diimplementasikan di dalam kehidupan, termasuk dalam bentuk program komputer untuk menyelesaikan suatu permasalahan tertentu. Algoritma juga merupakan bentuk dari suatu strategi atau ‘resep’ yang kamu gunakan untuk menyelesaikan suatu masalah. Algoritma lahir dari suatu proses berpikir komputasional oleh seseorang untuk menemukan solusi dari suatu permasalahan yang diberikan. Dengan demikian, berpikir komputasional merupakan keahlian yang kamu perlukan untuk dapat membuat algoritma, program, atau suatu karya informatika yang dapat digunakan dengan efektif dan efisien.



Kamu menganalisis suatu problem menggunakan teknik abstraksi dan dekomposisi, kemudian menyusun algoritma dengan melakukan pengenalan pola dari problem sejenis. Algoritma tersebut harus direpresentasikan dalam bentuk yang dapat dipahami oleh orang lain. Selain itu, karena pada akhirnya akan diubah dalam bentuk kode program, algoritma tersebut harus ditulis dalam bentuk yang terdefinisi dengan baik (*well-defined*) dengan jumlah langkah yang terbatas. Algoritma merupakan rancangan logis dan sistematis dari sebuah program. Dengan demikian, kemampuan menuliskan algoritma dengan baik akan membantu kamu dalam membuat program yang baik dan benar.

Algoritma tidak harus selalu terkait dengan pemrograman. Kamu sebenarnya sering menemui algoritma dalam kehidupan sehari-hari. Pada dasarnya, sebuah rangkaian instruksi yang tersusun secara sistematis dan digunakan untuk menyelesaikan suatu permasalahan atau mencapai tujuan tertentu, dapat disebut sebagai algoritma. Misalnya, sebuah resep bagaimana memasak suatu hidangan dapat disebut sebagai sebuah algoritma.

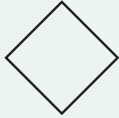
Pada bagian ini, kamu akan mempelajari dua cara untuk merepresentasikan algoritma, yaitu lewat diagram alir dan pseudokode. Untuk itu, kamu perlu mempelajari teknik untuk membaca suatu algoritma (yang disebut sebagai penelusuran atau *tracing*) dan cara untuk menuliskan suatu algoritma. Perlu diingat bahwa menulis algoritma berbeda dengan menulis program. Program ditulis agar dapat dipahami oleh mesin, sedangkan algoritma ditulis agar dapat dipahami oleh manusia. Untuk program yang sederhana, algoritma akan sangat mirip, bahkan sama dengan program. Jika persoalan makin kompleks, algoritma hanya berisi abstraksi, yang akan mempermudah implementasinya menjadi program.

2. Diagram Alir

Diagram alir merupakan notasi algoritma dengan menggunakan gambar atau simbol sebagai penjelas dari suatu algoritma. Diagram alir memungkinkan suatu proses dapat dipahami dengan mudah dan cepat. Diagram alir dibuat dalam bentuk aliran simbol yang dapat ditelusuri dari suatu titik permulaan hingga titik akhir dari program. Diagram alir dibuat menggunakan simbol standar ANSI/ISO yang beberapa simbol dasarnya diberikan pada Tabel 2.1.



Tabel 2.1 Simbol Diagram Alir Beserta Maknanya

Simbol	Nama	Deskripsi
	Garis alir (<i>flowline</i>)	Arah yang menunjukkan aliran program dari awal hingga akhir.
	Terminator	Titik awal atau titik akhir suatu program.
	Proses	Suatu kegiatan komputasi yang dilakukan oleh program: misalnya operasi aritmatika.
	Percabangan	Merupakan titik percabangan yang salah satu cabangnya dapat dilalui oleh program berdasarkan suatu kondisi.
	Masukan (<i>Input</i>)/ Keluaran (<i>Output</i>)	Melambangkan titik saat program akan menerima suatu data atau menghasilkan suatu informasi.
	Subprogram	Melambangkan suatu kegiatan atau proses lain yang telah didefinisikan sebelumnya.
	Penghubung dalam Halaman	Digunakan untuk menghubungkan suatu titik pada diagram alir ke titik lain pada halaman yang sama.
	Penghubung Antarhalaman	Digunakan untuk menghubungkan suatu titik pada diagram alir ke titik lain pada halaman yang berbeda. Digunakan apabila diagram lain cukup kompleks sehingga tidak dapat digambar dalam satu halaman.

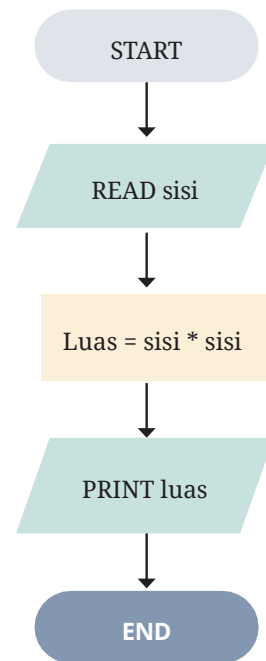
Untuk memahami bagaimana diagram alir digunakan untuk menggambarkan suatu algoritma, pada bagian berikut, diberikan lima buah contoh diagram alir dari beberapa proses berpikir yang telah kamu kenal.



a. Menghitung Luas Persegi

Diagram alir di samping dapat dibaca sebagai berikut.

- Diagram alir dibaca mulai dari simbol START, lalu mengikuti arah panah.
- Untuk menghitung luas persegi, kamu memerlukan sebuah data, yaitu panjang sisi. Panjang sisi ini dibaca pada diagram alir dengan menggunakan kata kunci READ dan disimpan dalam sebuah variabel bernama *sisi*.
- Setelah itu, kamu melakukan suatu proses ekspresi matematika untuk menghitung luas persegi menggunakan rumus yang telah kamu ketahui, yaitu $\text{luas} = \text{sisi} \times \text{sisi}$. Hasil perhitungan tersebut disimpan pada sebuah variabel bernama *luas*.
- Walaupun telah memperoleh jawaban yang dicari, komputer perlu diinstruksikan secara spesifik untuk mengeluarkan jawaban tersebut.
- Setelah PRINT, algoritma berakhir karena simbol setelahnya ialah simbol END.



Secara ringkas, kamu pun dapat menulis diagram alir tersebut dalam tiga bagian, yaitu Masukan (*Input*), Proses, dan Luaran (*Output*) menjadi:

INPUT/MASUKAN : sisi

PROSES : $\text{luas} = \text{sisi} \times \text{sisi}$

OUTPUT/LUARAN : luas

b. Menghitung Luas Permukaan Kubus

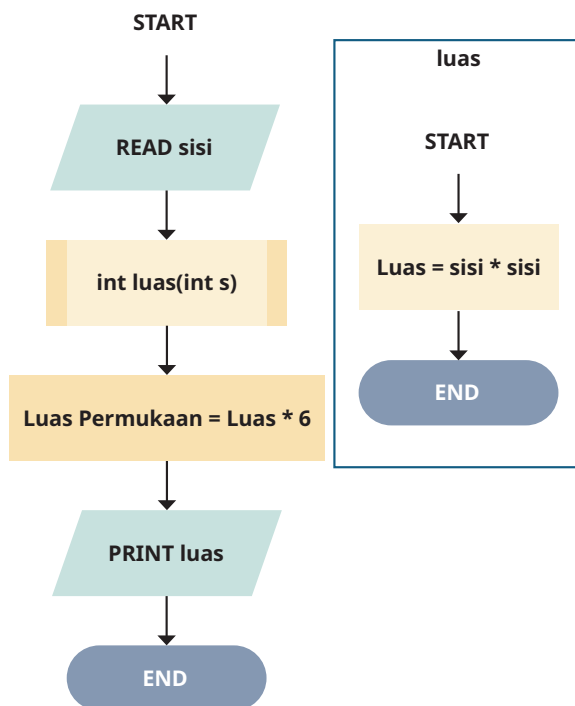
Saat menyusun solusi untuk menyelesaikan suatu permasalahan, kamu sering kali membutuhkan solusi dari permasalahan lain yang lebih sederhana. Misalnya, kita harus menghitung luas permukaan dari sebuah kubus. Pada prosesnya, kamu perlu menghitung luas persegi yang membentuk kubus tersebut. Hal ini dapat digambarkan pada diagram alir menggunakan simbol subprogram. Pada diagram alir ini, terlihat bahwa



proses akan memanggil subprogram menghitung luas persegi yang telah kamu buat sebelumnya. Subprogram dapat kamu gunakan untuk menggambarkan abstraksi dan dekomposisi yang telah kamu pelajari pada berpikir komputasional.

Diagram alir utama di samping dapat ditulis sebagai:
 INPUT/MASUKAN : sisi
 PROSES : luas * 6
 OUTPUT/LUARAN : luas permukaan

Diagram alir dari subprogram dapat ditulis sebagai:
 INPUT/MASUKAN : sisi
 PROSES : luas = sisi * sisi
 OUTPUT/LUARAN : luas



c. Membagi Bilangan

Diagram alir dapat memiliki beberapa kemungkinan aliran sehingga suatu algoritma dapat adaptif terhadap masukan yang diberikan. Hal ini dimungkinkan dengan adanya simbol percabangan. Aliran keluar dari simbol percabangan akan bergantung pada kondisi yang ada di dalam simbol percabangan.

Pada contoh ini, simbol percabangan digunakan untuk menghindari dijalankannya suatu operasi matematika yang tidak dapat dieksekusi oleh komputer, yaitu operasi pembagian dengan pembagi bernilai 0. Apabila operasi tersebut dilakukan, komputer akan menampilkan pesan kesalahan dan program akan berhenti secara tidak wajar.



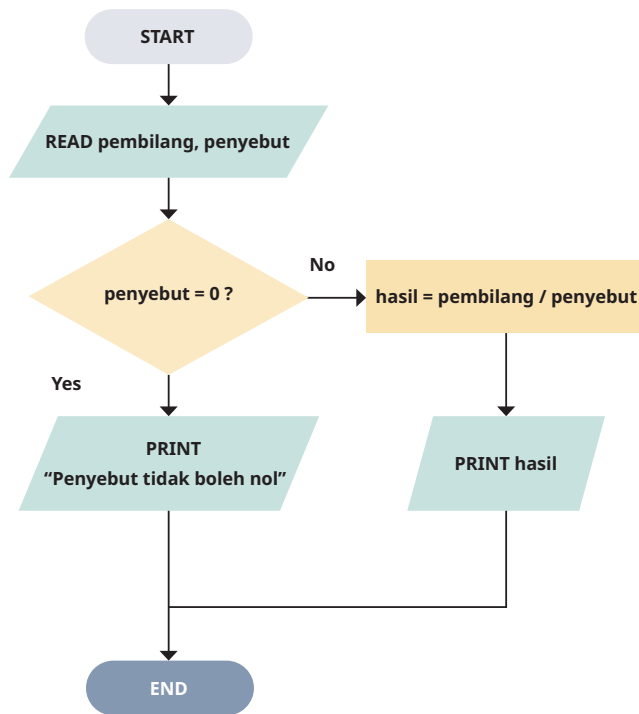


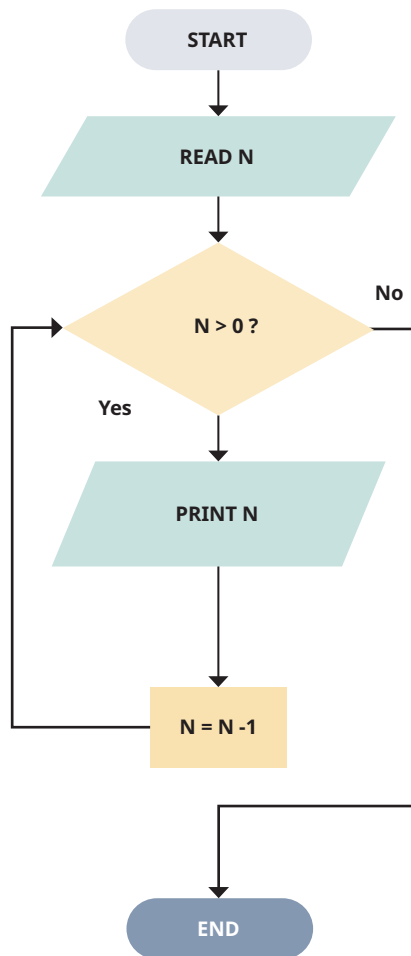
Diagram alir ini merupakan proses untuk membagi *pembilang* dengan *penyebut*. Akan tetapi, sebelum operasi pembagian dilakukan, diagram akan mengecek terlebih dahulu nilai dari *penyebut*. Apabila *penyebut* bernilai 0, operasi pembagian tidak dilakukan dan pesan yang sesuai akan ditampilkan. Jika tidak, operasi dapat dilakukan dengan aman dan hasil pembagian dapat ditampilkan.

d. Menghitung Mundur dari N hingga 1

Aliran pada diagram alir dapat diatur sehingga satu lebih simbol dijalankan berulang kali. Pada contoh berikut, perulangan dilakukan sehingga diagram alir tersebut akan menghasilkan barisan bilangan bulat dari N hingga 1. Misalkan N bernilai 5, diagram alir akan mencetak angka 5 4 3 2 1.

Tentunya, perulangan tidak dapat dilakukan terus-menerus sehingga diperlukan suatu kondisi untuk menghentikan perulangan. Simbol percabangan dapat digunakan untuk menghentikan perulangan tersebut pada kondisi yang kita tetapkan.

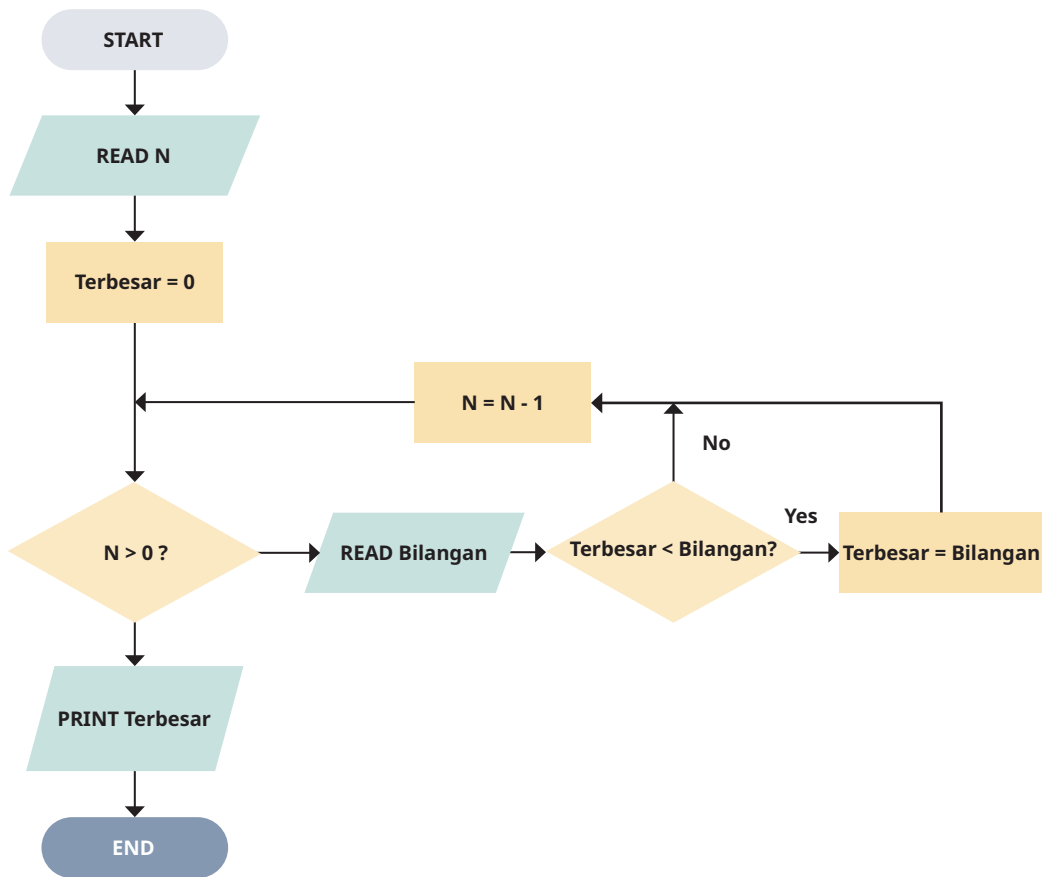




e. Mencari Bilangan Terbesar dari Suatu Himpunan Bilangan

Tentunya, simbol-simbol dasar pada diagram alir dapat dipadukan untuk menghasilkan sebuah proses yang lebih kompleks. Diagram alir berikut menggambarkan proses mencari bilangan terbesar dari suatu himpunan N buah bilangan. Diagram alir berikut akan membaca sebanyak N buah bilangan dan akan menghasilkan bilangan yang paling besar di antara bilangan tersebut.





f. Menelusuri Diagram Alir

Di buku kerjamu, kamu dapat melakukan penelusuran (*tracing*) secara terstruktur dengan membuat tabel sederhana yang terbagi tiga bagian, yaitu masukan, nilai variabel, dan keluaran. Bagian masukan akan diisi dengan data yang akan diproses, bagian nilai variabel akan menjadi tempatmu mencatat nilai yang disimpan di dalam variabel, dan bagian keluaran akan diisi dengan data yang dihasilkan oleh diagram alir. Penelusuran ini menjadi penting saat kamu ingin memahami perilaku dari suatu algoritma, atau saat kamu ingin mengecek ketepatan keluaran dari suatu algoritma.

Sebagai contoh, misal kamu mencari bilangan terbesar dari 4 bilangan berikut: 1, 3, 2, 4 menggunakan diagram alir kelima. Ada beberapa kegiatan inti yang akan kamu lakukan dalam melakukan penelusuran.



- Mempersiapkan lembar kerja penelusuran (Tabel 2.2.a), menuliskan data yang akan diolah, menuliskan nama variabel yang digunakan pada diagram alir, dan mengosongkan bagian keluaran.
- Menelusuri diagram alir dari bagian permulaan.
- Ketika menemukan simbol untuk membaca suatu data, kamu dapat mencoret masukan yang dibaca, kemudian meletakkannya ke variabel yang tepat. Misalnya, berada pada simbol masukan READ N . Letakkan bilangan pertama di bagian masukan (yaitu 4) ke variabel N (Tabel 2.2.b).
- Ketika tiba di suatu simbol proses yang menyimpan suatu nilai pada variabel (penugasan atau *assignment*), meletakkan nilai tersebut pada bagian nilai variabel. Misalnya, saat bertemu dengan $Terbesar = 0$ pada diagram alir, menuliskan nilai 0 pada variabel terbesar (Tabel 2.2.c).
- Proses juga dapat berisi ekspresi matematika, misalnya $N = N - 1$. Untuk mengerjakan ekspresi tersebut, kerjakan dahulu bagian kanan dari ekspresi, yaitu $N - 1$. Cek nilai N saat ini di bagian nilai variabel, dan akan menemukan nilai 4. Kerjakan ekspresi tersebut, yaitu $4 - 1 = 3$, lalu simpan hasilnya ke sisi kanan dari ekspresi, yaitu variabel N . Lewat ekspresi ini, nilai N yang tadinya 4, sekarang telah berubah menjadi 3. Kamu dapat mencoret nilai 4 dan menuliskan nilai 3 pada Lembar Kerja (Tabel 2.2.d).
- Kemudian, dapat melanjutkan proses dan akhirnya menemukan nilai N sekarang bernilai 0. Lembar kerjamu akan berisi seperti Tabel 2.2.e.
- Kemudian, menemukan simbol keluaran PRINT Terbesar. Pada tahap ini, kamu dapat menuliskan isi dari variabel terbesar ke dalam bagian keluaran di lembar kerjamu (Tabel 2.2.f).
- Terakhir, menelusuri dan menemukan terminator END sehingga penelusuran berakhir. Dengan demikian, diagram alir tadi menghasilkan keluaran berupa nilai 4 pada kasus yang diberikan. *Selamat! Kamu telah berhasil menelusuri diagram alir!*



Tabel 2.2 Ilustrasi Penelusuran Diagram Alir di Buku Kerja Siswa

(a) Sebelum menelusuri.	(b) Saat menemukan simbol masukan dengan perintah READ N .	(c) Saat menemukan simbol proses dengan perintah terbesar = 0.
Masukan 4 1 3 2 4	Masukan 4 1 3 2 4	Masukan 4 1 3 2 4
Nilai variabel N: Bilangan: Terbesar:	Nilai variabel N : 4 Bilangan: Terbesar:	Nilai variabel N : 4 Bilangan: Terbesar: 0
Keluaran	Keluaran	Keluaran
(d) Saat menemukan simbol proses dengan ekspresi $N = N - 1$.	(e) Saat N bernilai 0, Lembar Kerja akan menjadi seperti ini.	(f) Saat menemukan simbol keluaran PRINT Terbesar.
Masukan 4 1 3 2 4	Masukan 4 1324	Masukan 4 1324
Nilai variabel N : 4 3 Bilangan: Terbesar:	Nilai variabel N : 4 324 0 Bilangan: 1 324 0 Terbesar: 4	Nilai variabel N : 4 324 0 Bilangan: 1 324 0 Terbesar: 4
Keluaran	Keluaran	Keluaran 4

Walaupun ilustrasi ini terdiri atas beberapa tabel, kamu cukup bekerja dengan satu tabel saat menelusuri diagram alir. Teknik penelusuran ini tidak hanya dapat digunakan untuk membaca suatu diagram alir, tetapi



juga dapat digunakan untuk membaca pseudokode atau kode program. Untuk algoritma yang pendek, kamu mungkin tidak membutuhkan lembar kerja seperti ini, tetapi lembar kerja ini akan sangat bermanfaat ketika menelusuri suatu algoritma yang panjang dan kompleks. Sekarang, saatnya kamu berlatih menggunakan teknik ini!



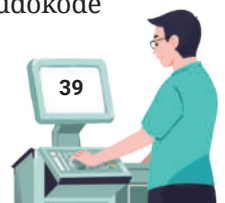
Aktivitas 10.2-01-U Menelusuri Diagram Alir

Sebelum membuat diagram alir sendiri, kamu perlu kemampuan untuk membaca diagram alir yang telah tersedia. Oleh karena itu, pada latihan ini, kamu akan menelusuri diagram alir yang telah tersedia di atas ketika diberikan suatu kasus untuk diselesaikan. Walaupun kamu telah mengetahui apa hasilnya, lakukanlah penelusuran secara terstruktur mulai dari awal hingga akhir diagram alir.

1. Hitunglah luas persegi dengan panjang sisi 3 cm menggunakan diagram alir pertama.
2. Hitunglah luas permukaan sebuah kubus yang memiliki panjang sisi 20 cm dengan menggunakan diagram alir kedua.
3. Hitunglah dua buah operasi pembagian berikut dengan menggunakan diagram alir ketiga. Pertama, pembagi bernilai 10 dan penyebut bernilai 2. Kedua, pembagi bernilai 8 dan penyebut bernilai 0.
4. Lakukan hitung mundur dari angka 5 hingga 1 dengan menelusuri diagram alir keempat.

3. Pseudokode

Pseudokode (kode semu atau kode pseudo) adalah suatu bahasa buatan manusia yang sifatnya informal untuk merepresentasikan algoritma menggunakan konvensi gabungan bahasa pemrograman. Pseudokode dibuat untuk menutupi kekurangan diagram alir dalam merepresentasikan konsep-konsep pemrograman terstruktur. Caranya dengan memberikan representasi langkah-langkah yang lebih detail dan dekat dengan bahasa pemrograman. Pseudokode



juga dapat merepresentasikan penggunaan simbol aritmatika, logika, dan penugasan (*assignment*) dengan lebih baik. Karena sifatnya yang informal, tidak ada aturan khusus dalam standar notasi yang dapat digunakan. Akan tetapi, ada beberapa prinsip dasar yang perlu diperhatikan, yaitu satu baris untuk satu **pernyataan** (*statement*) dan pentingnya indentasi dalam menuliskan pernyataan. Indentasi ada untuk hierarki dari pernyataan. Misalnya, untuk menunjukkan bahwa suatu pernyataan merupakan bagian dari sebuah struktur kontrol percabangan atau struktur kontrol perulangan (lihat konsep blok pada pemrograman visual yang telah kamu pelajari di tingkat SMP).

Kelima diagram alir pada bagian sebelumnya dapat ditulis dalam bentuk pseudokode sebagai berikut.

Tabel 2.3 Pseudokode 1: Menghitung Luas Persegi

Deskripsi Naratif	Pseudokode
1. Baca nilai sisi persegi. 2. Hitung luas dengan mengkuadratkan nilai sisinya. 3. Cetak luas.	Algoritma Menghitung Luas Persegi Input: Nilai Panjang Sisi Persegi. Proses: BACA sisi luas $\leftarrow$ sisi * sisi CETAK luas Output: Luas Persegi tercetak.

Tabel 2.4 Pseudokode 2: Menghitung Luas Permukaan Kubus

Deskripsi Naratif	Pseudokode
1. Baca nilai sisi kubus. 2. Hitung luas (persegi) dari sisi kubus. Kalikan luas kubus dengan angka 6 (banyaknya jumlah persegi pada kubus) untuk mendapatkan luas permukaan. 3. Cetak luas_permukaan.	Algoritma Menghitung Luas Permukaan Kubus Input: Nilai Panjang Sisi Kubus. Proses: BACA sisi luas_permukaan $\leftarrow$ luas(sisi) * 6 CETAK luas_permukaan Output: Luas Permukaan Kubus tercetak.



Tabel 2.5 Pseudokode 3: Membagi Bilangan

Deskripsi Naratif	Pseudokode
1. Baca nilai pembilang dan penyebut. 2. Jika penyebut bernilai 0, cetak tulisan “Penyebut tidak boleh nol”. 3. Jika penyebut tidak bernilai 0, lakukan pembagian pembilang dengan penyebut dan simpan hasilnya. 4. Cetak hasil pembagian.	Algoritma Membagi Bilangan Input: Pembilang dan Penyebut. Proses: BACA pembilang, penyebut IF penyebut = 0 PRINT “Penyebut tidak boleh nol” ELSE hasil $\leftarrow$ pembilang / penyebut CETAK hasil Output: Hasil Pembagian tercetak.

Tabel 2.6 Pseudokode 4: Menghitung Mundur dari N hingga 1

Deskripsi Naratif	Pseudokode
1. Baca nilai N. 2. Selama $N > 0$ Ulangi. 3. Cetak tulisan N. 4. Kurangi nilai N dengan 1.	Algoritma Menghitung Mundur dari N hingga 1 Input: Nilai N. Proses: BACA N WHILE $N > 0$ CETAK N $N \leftarrow N - 1$ Output: Angka hasil hitung mundur dari nilai N sampai 1 tercetak.



Tabel 2.7 Pseudokode 5: Mencari Bilangan Terbesar dari Suatu Himpunan Bilangan

Deskripsi Naratif	Pseudokode
- Jika himpunan bilangan kosong, tidak ada bilangan terbesar. - Jika himpunan bilangan tidak kosong, asumsikan bilangan pertama sebagai bilang terbesar saat ini. - Untuk setiap bilangan anggota himpunan: bandingkan bilangan tersebut dengan bilangan terbesar saat ini. Apabila bilangan tersebut lebih besar, bilangan tersebut akan menjadi bilangan terbesar saat ini. - Apabila langkah 3 telah dilakukan pada semua bilangan, bilangan terbesar saat ini akan menjadi bilangan terbesar di himpunan bilangan tersebut.	Algoritma Mencari Bilangan Terbesar Input: Himpunan bilangan L. Proses: IF size of (L)= 0 RETURN null largest ← L[0] FOR EACH item in L, DO IF item > largest, THEN largest ← item RETURN largest Output: Bilangan terbesar pada himpunan bilangan L

Perhatikan bahwa pseudokode mengandung kata-kata yang pada contoh di atas, dicetak dengan huruf kapital seperti IF, THEN, FOR, RETURN, WHILE. Kata-kata tersebut merupakan kata-kata terbatas (*reserved word*) yang merupakan konvensi umum pada saat menulis bahasa pemrograman. Arti dari kata-kata tersebut secara lebih detail akan dijelaskan pada bab ini.

Algoritma yang telah disusun kemudian dapat dibuat menjadi suatu program komputer dengan melewati suatu proses pemrograman. Seorang pemrogram akan menggunakan bahasa pemrograman untuk menulis kode program yang akan dijalankan oleh komputer. Ada banyak bahasa pemrograman yang dapat digunakan dengan karakteristik masing-masing, misalnya C, C++, C#, Python, Java, dan R.

Aktivitas 10.2-02-U
Menelusuri Pseudokode

Dengan menggunakan instruksi yang sama dengan Aktivitas 10.2-01-U Menelusuri Diagram Alir, lakukan penelusuran pada pseudokode yang telah diberikan pada contoh.



Aktivitas 10.2-03-U Menulis Algoritma

Pada latihan ini, kamu diminta untuk menuliskan suatu algoritma berdasarkan deskripsi yang diberikan. Deskripsi ini berupa pernyataan dalam bahasa sehari-hari dari algoritma yang perlu kamu buat dalam bentuk diagram alir dan pseudokode. Setelah selesai, kamu dapat menunjukkan hasil pekerjaanmu kepada temanmu untuk ditelusuri.

Soal 1: Membayar Bakso (Tingkat Kesulitan: ★ ★)

Buatlah sebuah diagram alir atau pseudokode dari proses berikut.

Sebuah mesin pembayaran otomatis dirancang untuk mampu menangani pembayaran pembelian bakso secara mandiri. Mesin ini mampu untuk memberikan kembalian dalam bentuk uang kertas atau uang logam. Mesin akan menerima dua buah masukan, yaitu total bayar dan jumlah uang yang dibayarkan oleh pelanggan. Apabila jumlah uang yang dibayarkan lebih besar atau sama dengan total bayar, mesin akan menghitung kembalian yang harus diberikan kepada pelanggan. Apabila terjadi sebaliknya, mesin akan menampilkan teks “Uang yang dibayarkan kurang”.

Setelah diagram alir selesai, kamu dapat menelusurinya dengan menggunakan kasus berikut.

Kasus	Masukan	Keluaran
1	Total Bayar: 10.000 Jumlah Uang: 15.000	5.000
2	Total Bayar: 20.000 Jumlah Uang: 10.000	Uang yang dibayarkan kurang.

Soal 2: Hadiah Bakso Gratis (Tingkat Kesulitan: ★ ★ ★)

Kamu ialah pengusaha bakso yang sukses. Agar usaha baksomu bisa lebih berkembang, kamu berencana untuk menambah sentuhan teknologi sehingga beberapa proses dapat berjalan secara otomatis. Inovasi yang kamu pikirkan ialah menggunakan sistem poin untuk memberikan diskon kepada pelanggan. Poin ini akan diberikan pada saat pelanggan membayar di mesin pembayaran yang akan kamu



buat. Setiap membayar, pelanggan akan menerima poin senilai harga bakso yang dia beli. Apabila total poin mencapai 100.000, pelanggan akan menerima satu porsi bakso gratis.

Kamu lalu memikirkan suatu proses berikut: setelah memesan bakso, pelanggan dapat membayar dengan menggunakan ponsel miliknya. Kemudian, mesin tersebut menambahkan total pembayaran ke total poin yang saat ini dimiliki oleh pelanggan. Apabila total poin yang dimiliki pelanggan lebih besar dari 100.000, mesin akan mengeluarkan kalimat “Anda mendapatkan kupon bakso gratis” dan mengurangi total poin pelanggan dengan nilai 100.000. Setelah itu, mesin akan menampilkan total poin pelanggan saat ini.

Setelah diagram alir selesai, kamu dapat menelusurinya dengan menggunakan kasus berikut.

Kasus	Masukan	Keluaran
1	Total Pembayaran: 80.000 Total Poin Pelanggan Saat Ini: 10.000	Poin Anda saat ini: 90.000
2	Total Pembayaran: 20.000 Total Poin Pelanggan Saat Ini: 90.000	Anda mendapatkan kupon bakso gratis! Poin Anda saat ini: 10.000

Ayo, Renungkan!

Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal.

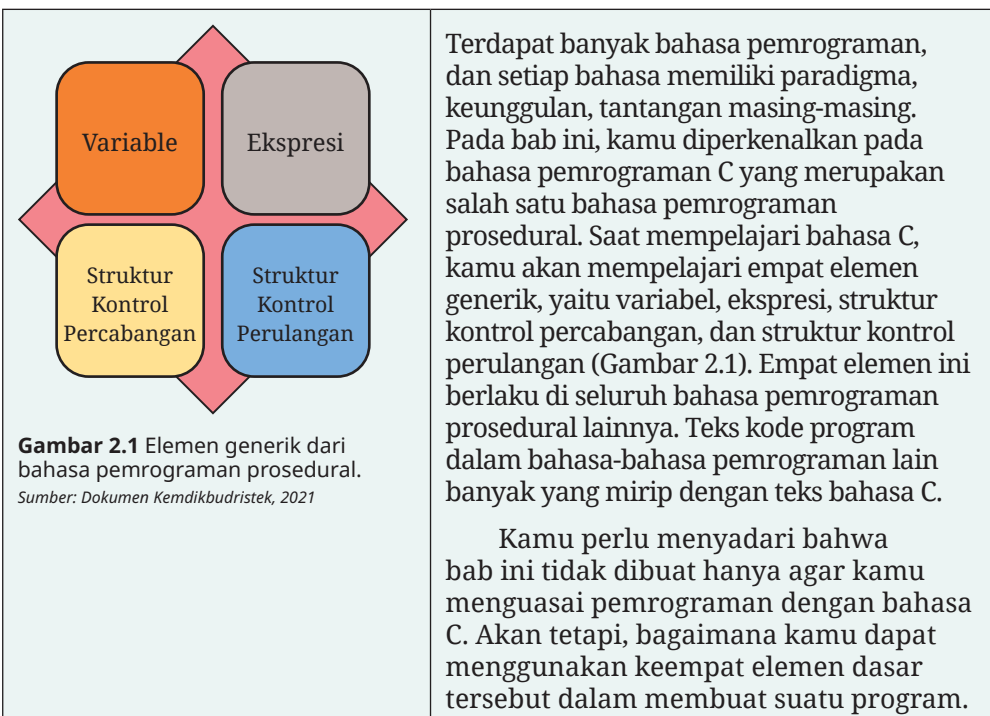
1. Apakah kamu merasa bahwa membuat algoritma dapat mempermudah kamu dalam menyelesaikan masalah?
2. Mana yang paling mudah untuk kamu gunakan, diagram alir atau pseudokode?
3. Setelah ini, apa yang akan kamu lakukan untuk dapat membaca dan menulis algoritma dengan baik?
4. Pelajaran paling berkesan apa yang kamu dapatkan dari pertemuan ini?



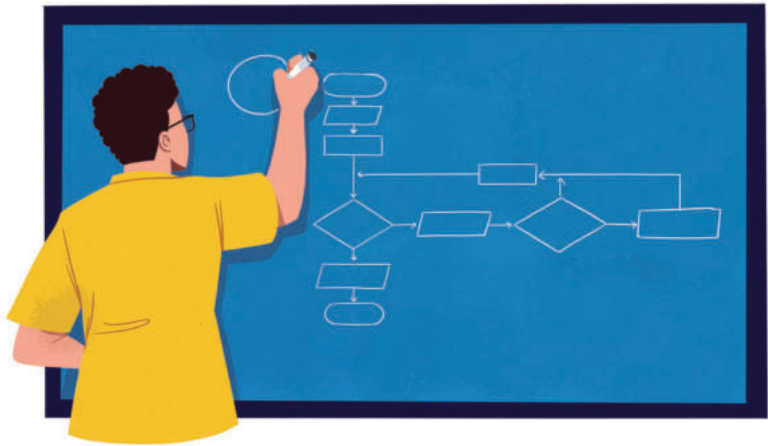
B. Membuat Program Sesuai Struktur Kendalinya

Belajar bahasa pemrograman sama halnya dengan belajar bahasa apa pun, dimulai dengan secara intuitif mengenal dan langsung memakai bahasa tersebut untuk keperluan sehari-hari yang penting sesuai kebutuhan, bukan dari teori bahasa. Seseorang dengan bahasa ibu bahasa Indonesia, saat belajar bahasa Inggris, akan mulai mengenal bahasa Inggris melalui “membaca” contoh-contoh kalimat sederhana, yang sangat diperlukan dalam kehidupan sehari-hari, seperti mulai menyapa “Selamat pagi”, “Jam berapa?”. Melalui contoh tersebut, dia akan belajar kosakata penting dan pola kalimat, misalnya kalimat pernyataan SPOK (Subjek, Predikat, Objek, Keterangan); struktur kalimat tanya, kalimat aklamasi, dan lain-lain. Selanjutnya, baru beranjak ke konsep yang lebih kompleks yang ada pada bahasa asing tersebut, dan mengenal tata bahasa secara lebih formal dan mulai menulis.

Belajar pemrograman pada hakikatnya sama dengan belajar bahasa natural (bahasa manusia sehari-hari): seseorang belajar dari “membaca” program terlebih dulu, daripada “menulis” kode program. Proses menulis kode (*coding*) dapat dimulai setelah kamu membaca contoh-contoh program yang menjadi pola pembangun program kompleks. Bedanya dengan belajar bahasa natural, teks dalam bahasa pemrograman yang ditulis bukan dilafalkan dan dipahami sebagai teks “statis”, melainkan juga dapat dipahami oleh mesin dan dapat dieksekusi (dijalankan).



1. Belajar Algoritma sambil Menyelesaikan Masalah



Aktivitas Ayo, Kita Berlatih sebagai bentuk latihan menyelesaikan suatu problem yang diberikan dengan pemrograman. Karena materi ini bersifat pengenalan pada kegiatan menulis kode program (*coding*), problem yang diberikan dalam bentuk spesifikasi yang telah terstruktur. Pada kenyataannya, ketika kamu membuat program untuk menyelesaikan suatu permasalahan nyata, kamu perlu membuat sendiri spesifikasi dari program yang akan dibuat. Kamu perlu menetapkan sendiri tujuan program, format masukan, serta format keluaran program. Aktivitas ini akan kamu lakukan di akhir bab. Sekarang, kamu dapat fokus membuat program berdasarkan spesifikasi yang telah diberikan.

Pada setiap aktivitas, kamu akan diberikan setidaknya satu permasalahan untuk diselesaikan dengan menggunakan konsep yang telah diberikan pada buku. Bentuk permasalahan tersebut akan diberikan dengan struktur pada Tabel 2.8.

Tabel 2.8 Spesifikasi Program yang Diberikan pada Problem Latihan

Bagian	Penjelasan
Nomor dan Nama Problem	Identitas dari permasalahan.
Deskripsi Soal	Memberikan konteks permasalahan yang perlu diketahui oleh siswa dalam membuat program.



Bagian	Penjelasan
Format Masukan	Memberikan susunan data yang diberikan pada program oleh pengguna, beserta ukuran dari data tersebut.
Format Keluaran	Memberikan susunan informasi yang akan dikeluarkan oleh program kepada pengguna.
Contoh Masukan	Memberikan contoh data yang dimasukkan.
Contoh Keluaran	Memberikan contoh informasi yang dikeluarkan program berdasarkan data yang dimasukkan.

Bagian format keluaran dan format masukan menjadi penting karena *autograder* (jika kamu menggunakan ini di kelas) akan memeriksa kode programmu dengan sangat ketat. Pemeriksaan tersebut berdasarkan pasangan masukan-keluaran yang telah disiapkan, atau disebut sebagai kasus uji (*test case*). Apabila program kamu membaca atau menulis dengan format yang salah, *auto-grader* tidak akan menganggap programmu benar. Untuk membantu kamu memahami format tersebut, satu atau lebih contoh masukan dan keluaran akan diberikan.

Praktik Baik Pemrograman

Pada setiap problem, kamu akan menemukan beberapa kasus uji yang dapat digunakan untuk menguji program yang kamu buat. Akan tetapi, bukan berarti kamu hanya perlu mengecek programmu dengan kasus uji yang ada di buku. Kamu perlu menguji programmu dengan menggunakan berbagai kasus uji yang kamu buat sendiri. Beberapa problem dirancang dengan adanya jebakan yang dapat membuat programmu gagal berjalan pada kasus tertentu. Misalnya, ketika masukan yang diberikan memiliki rentang yang besar atau adanya kondisi yang dapat menyebabkan program berhenti.

2. Ekspresi

Ekspresi merupakan bagian yang tidak terpisahkan dari program. Di dalam matematika, ekspresi terdiri atas kombinasi beberapa *operand* dan operator yang memiliki makna. Kamu telah terbiasa menulis ekspresi pada matematika, misalnya dalam penjumlahan $10 + 5$ yang melibatkan dua buah *operand* (10 dan 5) dan sebuah operator (+). Ekspresi pada pemrograman mirip dengan



ekspresi yang kamu pelajari pada matematika, tetapi diperkaya dengan tambahan operator untuk memudahkan kamu dalam menulis program.

a. Ekspresi berdasarkan Jumlah *Operand*

Berdasarkan jumlah *operand*, suatu ekspresi dapat dibagi menjadi 3 ekspresi berikut.

- *Unary* (satu *operand*), misalnya $-a$ untuk menegasikan suatu bilangan.
- *Binary* (dua *operand*), misalnya $a+b$ untuk penjumlahan.
- *Ternary* (tiga *operand*), misalnya $a ? b : c$ yang akan dijelaskan lebih rinci.

b. Operator berdasarkan Fungsi

Berdasarkan fungsinya, operator dapat dibagi menjadi operator penugasan, operator aritmatika, operator logika, serta *increment* dan *decrement*.

1) Operator Penugasan

Operator penugasan (simbol ‘=’) digunakan untuk memberikan suatu nilai konstanta atau nilai yang diperoleh dari suatu ekspresi ke dalam variabel. *Operand* di sisi kiri akan menerima nilai dari *operand* di sisi kanan operator penugasan. Contoh penugasan telah kamu lakukan pada saat melakukan deklarasi variabel seperti:

$a = 5;$

Perhatikan dan ingat baik-baik bahwa makna dari simbol ‘=’ dalam pemrograman berbeda dengan tanda “sama dengan” dalam matematika yang menandakan persamaan!

2) Operator Aritmatika

Operator aritmatika digunakan untuk melakukan operasi matematika yang kamu kenal. Ada perbedaan notasi penulisan operator dengan yang kamu pelajari pada mata pelajaran Matematika yang dapat dilihat pada Tabel 2.9.



Tabel 2.9 Operator Matematika yang Umum Digunakan pada Bahasa

Operasi Matematika	Contoh Ekspresi Aljabar	Operator Bahasa Pemrograman	Ekspresi Bahasa Pemrograman
Penjumlahan	$1 + 2$	$+$	$1 + 2$
Pengurangan	$a - b$	$-$	$1 - b$
Perkalian	ab	$*$	$a * b$
Pembagian	a/b	$/$	a / b
Modulo (sisir bagi)	$a \bmod b$	$\%$	$a \% b$

Seperti pada matematika, suatu ekspresi dapat terdiri atas banyak operator. Agar ekspresi tersebut dieksekusi dengan benar, diperlukan suatu urutan operasi (atau *operator precedence*). Urutan operasi dari operator pada bahasa pemrograman seperti C ialah:

- Tanda kurung $()$
- Operator perkalian $*$, pembagian $/$, dan modulo $\%$.
- Operator penjumlahan $+$ dan pengurangan $-$

Selain operator yang kamu temukan di matematika, bahasa pemrograman juga memiliki operator lain seperti operator *left shift* $\ll$ dan *right shift* $\gg$ untuk mengalikan atau membagi nilai dengan faktor dua. Urutan operator yang lebih lengkap dapat dilihat pada dokumentasi bahasa pemrograman C yang dapat diakses di <https://buku.kemdikbud.go.id/s/xs8l5e>.

Hasil ekspresi dapat disimpan dalam suatu variabel menggunakan operator penugasan. Misalnya:

$a = 5;$

$b = a + 1;$

$b = b + 5;$

$c = (a + b) * 2 + a;$

$d = c \% b;$

Ekspresi seperti $b = b + 5$ mungkin tidak pernah kamu temui pada mata pelajaran Matematika. Akan tetapi, ekspresi ini sangat lazim ditemukan dalam menulis kode program. Makna dari ekspresi ini ialah kamu melakukan



perubahan pada nilai suatu variabel berdasarkan nilai variabel pada saat ini. Dalam hal ini, kamu mengubah nilai variabel b dengan menambahkan 5 terhadap variabel tersebut. Hal ini telah kamu lakukan pada saat melakukan penelusuran pada Diagram Alir 4 di bagian awal bab ini.

Untuk ekspresi seperti ini, bahasa C menyediakan bentuk yang lebih ringkas yang disebut sebagai *compound assignment operator* atau operator penugasan majemuk. Misalnya, $b = b + 5$ dapat disingkat menjadi $b += 5$. Hal ini juga berlaku untuk operator matematika lainnya seperti $b *= 2$, $c /= 3$, dan seterusnya.

3) Operator Increment dan Decrement

Selain operator penugasan majemuk pada bagian sebelumnya, terdapat operator unary yang lebih khusus untuk menambah (++) atau mengurangi nilai suatu variabel (--) dengan angka 1. Operator ini dapat diletakkan sebelum (*prefix*) atau setelah (*postfix*) *operand*. Pada penulisan dalam bentuk *prefix*, perubahan nilai akan langsung dilakukan pada nilai variabel sebelum nilai variabel tersebut digunakan pada ekspresi. Sebagai contoh, setelah dua baris kode berikut dijalankan, nilai x dan y akan bernilai 2.

```
x = 1;  
y = ++x;
```

Sebaliknya, pada penulisan dalam bentuk *postfix*, nilai variabel akan digunakan terlebih dahulu pada ekspresi, baru perubahan dilakukan.

Sebagai contoh, setelah dua baris kode berikut dijalankan, nilai x dan y akan bernilai 1.

```
x = 1;  
y = x++;
```

4) Operator Logika, Relasional, dan Kesamaan

Di samping operator aritmatika, juga dikenal operator logika, relasional, dan kesamaan. Ekspresi yang menggunakan operator ini akan memiliki nilai benar (*true* atau dalam bahasa C bernilai tidak sama dengan 0) atau salah (*false* atau bernilai 0). Operator pada kategori ini memiliki peran yang sama dengan operator logika, relasional, dan kesamaan pada mata pelajaran Matematika. Operator tersebut dapat dilihat pada Tabel 2.10.



Tabel 2.10 Operator Logika, Relasional, dan Kesamaan pada Bahasa Pemrograman

Aljabar	Program	Contoh	Makna
Operator Kesamaan/Pertidaksamaan			
=	==	$a == b$	Apakah nilai a sama dengan b ?
≠	!=	$a != b$	Apakah nilai a tidak sama dengan b ?
Operator Relasional			
>	>	$a > b$	Apakah a lebih besar daripada b ?
<	<	$a < b$	Apakah a lebih kecil daripada b ?
≥	>=	$a >= b$	Apakah a lebih besar atau sama dengan b ?
≤	<=	$a <= b$	Apakah a lebih kecil atau sama dengan b ?
Operator Logika			
AND	&&	$a > 0 \ \&\& \ b > 0$	Apakah a dan b lebih besar daripada 0?
OR		$a > 0 \ \ b > 0$	Apakah a atau b lebih besar daripada 0?
NOT	!	$!(a > 0)$	Apakah a tidak lebih besar daripada 0?

Praktik Baik Pemrograman

Pada saat menuliskan suatu ekspresi, gunakanlah pengelompokan dengan tanda kurung untuk membuat ekspresi tidak ambigu dan mudah untuk dibaca. Pastikan ekspresi tersebut ditulis dengan benar sebelum kamu menjalankan program. Biasakan untuk menguji (dalam pikiran kamu) sambil menulis, tidak menunggu program dijalankan.

Aktivitas 10.2-04-U **Latihan Ekspresi**

Buatlah solusi program untuk menyelesaikan problem-problem berikut. Kamu dapat menggunakan diagram alir, pseudokode, pemrograman visual, atau bahasa pemrograman untuk menghasilkan solusi dari problem tersebut sesuai dengan kondisi di tempat kamu.



Problem 1: Menghitung Luas Tanah (Tingkat Kesulitan: ★★)

Deskripsi Soal:

Pak Algor memiliki sebidang tanah berbentuk segitiga siku-siku. Dia ingin mengetahui berapa luas tanah yang dia miliki. Bantulah Pak Algor menghitung dengan membuat program untuk menghitung luas tanahnya.

Format Masukan:

Dua buah bilangan bulat yang merupakan alas dan tinggi segitiga dari tanah Pak Algor.

Format Keluaran:

Sebuah bilangan riil, dengan dua digit di belakang desimal yang merupakan luas tanah Pak Algor. Akhiri keluaran dengan karakter *newline*.

Masukan	Keluaran
20 40	400.00

Problem 2: Menghitung Luas Persegi (Tingkat Kesulitan: ★★)

Simak kembali Diagram Alir 1: Menghitung Luas Persegi pada bagian Algoritma di bab ini. Buatlah program berdasarkan diagram alir tersebut. Panjang sisi yang diberikan ialah sebuah bilangan riil dengan nilai maksimum 1.000 cm.

Problem 3: Hasil Bagi dan Sisa Pembagian (Tingkat Kesulitan: ★★★)

Deskripsi Soal:

Buatlah sebuah program untuk menampilkan hasil dan sisa pembagian dari dua buah bilangan bulat positif a dan b .

Format Masukan:

Dua buah bilangan positif a dan b yang dipisahkan oleh karakter spasi. Keduanya bernilai paling besar 10 miliar.

Format Keluaran:

Dua buah bilangan bulat yang ditulis di baris berbeda. Baris pertama ialah hasil pembagian, baris kedua ialah sisa pembagian. Hasil pembagian dibulatkan ke bawah.

Masukan	Keluaran
1000 3	333 1



Problem 4: Benar atau Salah? (Tingkat Kesulitan: ★ ★ ★)

Ekspresi yang memuat operator logika, relasional, dan kesamaan dapat kamu telusuri tanpa menjalankan program. Berikut ini diberikan beberapa ekspresi yang perlu kamu periksa nilainya, jika diketahui nilai $a = 1$, $b = 2$, dan $c = 3$.

No	Ekspresi	Hasil
1	$(a < b) \mid \mid (b > c)$	
2	$(a >= b) \mid \mid (b != c)$	
3	$(b == a) \&\& (c > a)$	
4	$(b >= a) \&\& ((b < c) \mid \mid (c > a))$	

3. Struktur Kontrol Percabangan

Pernahkah kamu ingin pergi ke sebuah tempat tertentu dengan menggunakan moda transportasi? Keputusan menggunakan sebuah moda transportasi untuk bepergian biasanya bergantung pada sebuah keadaan tertentu. Apabila kondisi hujan, kamu akan lebih memilih menggunakan mobil daripada menggunakan sepeda motor. Jika mau ke luar pulau, kamu lebih memilih menggunakan pesawat terbang. Apabila cuaca sedang cerah dan jarak yang ditempuh dekat, kamu akan memilih menggunakan sepeda motor.

Komputer merupakan alat yang membantu banyak aktivitas manusia. Saat ini, komputer merupakan bagian yang tidak terpisahkan dari kehidupan sehari-hari. Pada dasarnya, komputer menjalankan perintah dari manusia. Perintah-perintah tersebut dituangkan secara tertulis dalam sebuah aturan tertentu yang disebut kode program. Penyusunan sebuah kode program membutuhkan kemampuan algoritma, logika, dan bahasa pemrograman. Aktivitas membuat program merupakan latihan untuk menuliskan sebuah kode program sederhana dalam rangka memberikan instruksi ke komputer untuk melakukan sebuah aksi.

Kamu akan belajar memahami konsep pemrograman sederhana khususnya yang berkaitan dengan sebuah struktur kontrol percabangan atau kondisional. Apa itu kondisional? Secara sederhana, kondisional dalam dunia matematika dikenal dalam bentuk pernyataan "jika maka" (*if else statement*).



Pernyataan ini dibuat untuk mengekspresikan sebuah aksi berdasarkan sebuah kondisi tertentu. Sebagai contoh, ketika kamu diminta untuk mengklasifikasikan sebuah bilangan merupakan bilangan ganjil atau genap, dengan mudah kamu membuat sebuah aturan sebagai berikut.

- Jika bilangan tersebut habis dibagi 2, maka bilangan tersebut termasuk bilangan genap.
- Jika bilangan tersebut tidak habis dibagi 2, maka bilangan tersebut termasuk bilangan ganjil.

Proses tersebut merupakan salah satu ilustrasi dari sebuah pernyataan kondisi.

Dalam kehidupan sehari-hari, manusia perlu mengambil keputusan. Keputusan tersebut dapat berupa keputusan sederhana hingga keputusan yang sangat kompleks. Banyak permasalahan yang di dalamnya terkandung pengambilan keputusan sehingga keputusan pun juga digunakan dalam menyelesaikan suatu permasalahan komputasi. Pada bagian ini, kamu akan melihat bagaimana kamu dapat menerapkan suatu bentuk keputusan sederhana di dalam program. Pada bahasa pemrograman C, dan juga banyak bahasa pemrograman lainnya, keputusan dapat diimplementasikan dalam bentuk struktur kontrol percabangan. Ada beberapa variasi dari struktur ini, yaitu pernyataan *if-else*, pernyataan *switchcase*, dan pernyataan yang bersarang.

a. Struktur Kontrol Percabangan *If - Else*

Ada beberapa variasi penggunaan struktur kontrol percabangan *If - Else*. Bentuk umum dari pernyataan *if* ialah sebagai berikut.

```
if (kondisi) {  
    <pernyataan>;  
    <pernyataan>;  
    ...  
}
```

Bagian kondisi dapat diisi dengan ekspresi yang menghasilkan nilai benar atau salah. Apabila kondisi menghasilkan nilai benar, semua pernyataan yang berada di dalam struktur kontrol percabangan tersebut akan dieksekusi oleh program. Sekarang, perhatikan program berikut, dan lakukan penelusuran untuk memeriksa keluaran dari program tersebut.



```

/*
 * Program Membandingkan Bilangan
 */

#include <stdio.h>
int main(){
    int a = 1, b = 1;
    if (a == b) {
        printf("a sama dengan b \n");
    }
    return 0;
}

```

Program tersebut menggunakan struktur percabangan pada baris 5-7. Ekspresi yang digunakan pada bagian kondisi ialah $a==b$, sedangkan pernyataan yang dieksekusi jika kondisi benar terdapat pada baris 6. Karena hanya terdapat satu pernyataan yang menjadi bagian dari struktur kontrol tersebut, kamu dapat menghilangkan tanda kurung kurawal sehingga program dapat ditulis menjadi:

```

/*
 * Program Membandingkan Bilangan
 */

#include <stdio.h>

int main(){
    int a = 1, b = 1;
    if (a == b)
        printf("a sama dengan b \n");
    return 0;
}

```

Struktur kontrol percabangan dapat menambahkan blok *else* yang akan dieksekusi apabila kondisi bernilai salah.



```
if (kondisi)
    <pernyataan>;
else
    <pernyataan>;
```

Misalnya, untuk mengecek apakah suatu bilangan merupakan bilangan ganjil atau genap, kamu dapat memanfaatkan struktur *if-else* sebagai berikut:

```
/*
 * Program Cek Ganjil-Genap
 */

#include <stdio.h>

int main(){
    int bilangan;
    scanf("%d", &bilangan);
    if (bilangan % 2 == 0)
        printf("Bilangan Genap\n");
    else
        printf("Bilangan Ganjil\n");
    return 0;
}
```

Apabila kondisi makin kompleks, struktur *if-else* ini dapat dikembangkan kembali menjadi:

```
if (kondisi ke-1)
    <statement>
else if (kondisi ke-2)
    <statement>
.....
else if (kondisi ke-n )
    <statement>
else
    <statement>
```



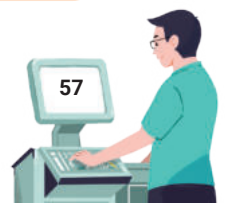
b. Struktur Kontrol Percabangan *Switch-Case*

Struktur kontrol percabangan yang memiliki cabang banyak dapat dibuat lebih sederhana menggunakan struktur *switch-case*. Bentuk umum dari struktur ini ialah sebagai berikut.

```
switch(switch_expr)
{
    case (constant expr1) :<statement>;
        <statement>;
        break;
    case (constant expr2) :<statement>;
        <statement>;
        break;
    ....
    default      : <statement>;
        <statement>;
        break;
}
```

Sebagai contoh, perhatikan kode program berikut.

```
#include <stdio.h>
int main()
{
    int bilangan, sisaPembagian;
    scanf("%d", &bilangan);
    sisaPembagian = bilangan % 4;
    switch (sisaPembagian)
    {
        case 0: printf("Habis Dibagi\n"); break;
        case 1: printf("Sisa Satu\n"); break;
        case 2: printf("Sisa Dua\n"); break;
        case 3: printf("Sisa Tiga\n"); break;
    }
    return 0;
}
```



Pada program di atas, struktur *switch-case* memeriksa nilai yang ada pada variabel sisa Pembagian. Karena nilai tersebut merupakan sisa pembagian sebuah bilangan dengan empat, hanya ada empat kemungkinan nilai, yaitu 0 sampai 3. Setiap kemungkinan nilai tersebut diperiksa melalui empat buah struktur *case* yang akan mencetak kalimat ke layar yang sesuai dengan sisa pembagian yang diperoleh.

c. Struktur Kontrol Percabangan Bersarang

Sebuah struktur kontrol dapat menjadi bagian dari suatu struktur kontrol lain. Hal ini disebut *nested* atau bersarang. Pada contoh berikut, diberikan sebuah kode program yang memiliki struktur kontrol percabangan bersarang. Telusurilah program tersebut jika program diberi masukan 1000 dan 10.

```
/*  
 * Program dengan IF bersarang  
 */  
  
#include <stdio.h>  
int main()  
{  
    int a, b;  
    scanf("%d %d", &a, &b);  
    if (b!=0)  
        if (a/b>10)  
            printf("1\n");  
    else  
        printf("-1\n");  
    return 0;  
}
```





Aktivitas 10.2-05-U

Latihan Struktur Percabangan

Buatlah solusi program untuk menyelesaikan problem-problem berikut.

Kamu dapat menggunakan diagram alir, pseudokode, pemrograman visual, atau bahasa pemrograman untuk menghasilkan solusi dari problem tersebut sesuai dengan kondisi di tempat kamu.

Problem 1: Membagi Bilangan (Tingkat Kesulitan: ★★)

Buatlah sebuah program dari Diagram Alir 3: Membagi Bilangan yang tersedia pada bagian Algoritma di awal bab ini.

Problem 2: Bilangan Bulat Positif (Tingkat Kesulitan: ★★)

Deskripsi Soal:

Buatlah program untuk mengecek apakah sebuah bilangan bulat ialah bilangan bulat positif.

- Jika bilangan bulat tersebut merupakan bilangan bulat positif, cetaklah “Bilangan Bulat Positif”.
- Jika bilangan bulat tersebut bukan merupakan bilangan bulat positif, jangan cetak apa pun.

Format Masukan:

Sebuah bilangan bulat n . Nilai n berada pada rentang $-100 < n < 100$.

Format Keluaran:

Satu baris kalimat sesuai pada deskripsi soal.

Masukan	Keluaran
3	Bilangan Bulat Positif
-10	

Problem 3: Jenis Bilangan Bulat (Tingkat Kesulitan: ★★★)

Deskripsi Soal:

Buatlah program untuk apakah sebuah bilangan bulat ialah bilangan termasuk bilangan bulat positif, negatif atau nol. Jika bilangan bulat tersebut merupakan bilangan bilangan bulat positif, cetak “Bilangan



Bulat Positif”. Jika bilangan bulat tersebut merupakan bilangan bulat negatif, cetak “Bilangan Bulat Negatif”. Jika bilangan bulat tersebut merupakan bilangan bulat nol, cetak “Bilangan Bulat Nol”.

Format Masukan:

Sebuah bilangan bulat n . Nilai n berada pada rentang $-100 < n < 100$.

Format Keluaran:

Satu baris kalimat sesuai pada deskripsi soal.

Masukan	Keluaran
3	Bilangan Bulat Positif
-5	Bilangan Bulat Negatif

Problem 4: Nama Bulan (Tingkat Kesulitan: ★★)

Deskripsi Soal:

Buatlah sebuah program yang menerima masukan bilangan bulat yang berada pada rentang 1–12, dan akan mencetak nama bulan yang sesuai dengan bilangan bulat tersebut. Apabila bilangan berada di luar rentang tersebut, cetak kalimat “Tidak ada bulan yang sesuai”.

Format Masukan:

Sebuah bilangan bulat n . Nilai n berada pada rentang $-100 < n < 100$.

Format Keluaran:

Satu baris kalimat sesuai pada deskripsi soal.

Masukan	Keluaran
1	Januari
3	Maret
13	Tidak ada bulan yang sesuai

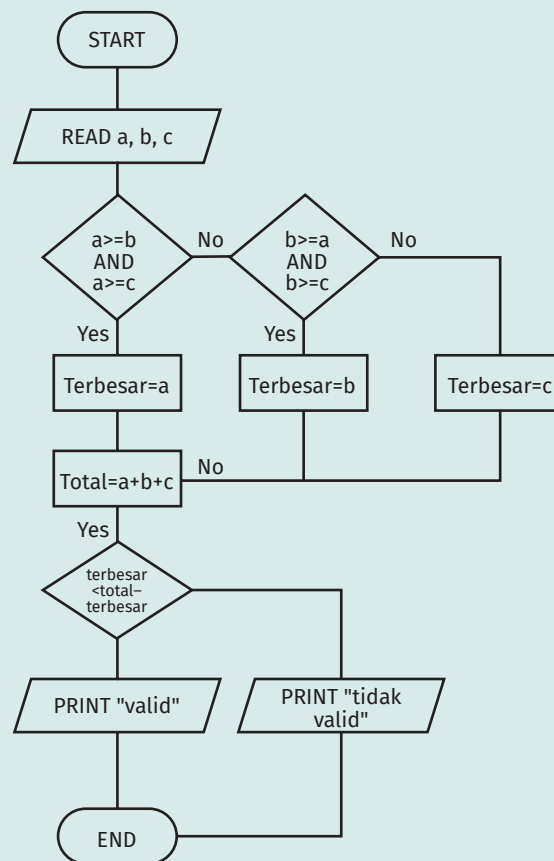
Problem 5: Mengecek Sisi Segitiga (Tingkat Kesulitan: ★★★)

Tahukah kamu bahwa sebuah segitiga hanya dapat dibangun apabila sisi terpanjangnya lebih kecil daripada total panjang kedua sisi lainnya? Jika syarat ini tidak dipenuhi, tidak ada segitiga yang terbentuk.



Agria sedang membuat program untuk menghitung luas segitiga yang menerima masukan berupa tiga buah bilangan bulat yang merupakan panjang sisi segitiga tersebut. Akan tetapi, Agria menyadari bahwa dia harus terlebih dahulu memastikan ketiga panjang sisi yang dimasukkan benar-benar dapat membentuk sebuah segitiga. Oleh karena itu, dia merancang sebuah algoritma dalam bentuk diagram alir berikut untuk mengecek apakah ketiga bilangan tersebut dapat membentuk segitiga.

Tugas kamu ialah membantu Agria dengan membuat program berdasarkan diagram alir tersebut.



Problem 6: Belajar Membuat Kasus Uji (Tingkat Kesulitan: ★★)

Perhatikan kembali diagram alir pada Problem 5. Diagram alir tersebut terlihat memiliki beberapa kemungkinan aliran, bergantung pada nilai masukan yang diberikan. Saat kamu mengecek program



yang kamu buat dengan suatu kasus uji, kasus uji yang diberikan haruslah meliputi semua kemungkinan aliran tersebut. Sekarang, buatlah kasus uji sedemikian rupa sehingga semua kemungkinan aliran pada diagram alir di atas dapat dicek.

Ayo, Renungkan!

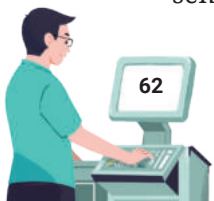
Jawablah pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal.

1. Pada bagian ini, kamu mendapatkan banyak konsep baru tentang program. Seperti apa perasaanmu saat ini?
2. Apakah kamu bereksperimen dengan contoh-contoh yang diberikan di buku? Jika ya, pengetahuan paling menarik apa yang kamu temukan dari hasil eksperimen tersebut?
3. Dari latihan yang telah kamu kerjakan, terutama dari problem 1 dan 5, apa salah satu contoh penggunaan dari struktur kontrol percabangan?
4. Apakah kamu sudah memahami proses membuat kasus uji untuk menguji sebuah program, yang harus meliputi semua kemungkinan aliran pada diagram alir? Mengapa kamu harus memastikan semua kemungkinan aliran dicek?
5. Kesalahan apa yang sering kamu lakukan saat menulis kode program dengan menggunakan struktur kontrol percabangan?

4. Struktur Kontrol Perulangan

Kamu akan belajar memahami konsep dan cara kerja perulangan dalam pemrograman. Setelah menyelesaikan aktivitas ini, kamu diharapkan mampu menerapkan perulangan dan menghubungkannya dengan kejadian sehari-hari.

- Pernahkah kamu melakukan aktivitas yang sama berulang kali? Eits... tidak perlu jauh-jauh, untuk melangkah saja, kita mengulang pergerakan kaki kanan dan kiri. Apakah kamu selalu berpikir sebelum melakukan setiap langkah? Pertanyaan terakhir tersebut dapat ditanyakan pada semua kegiatan perulangan yang kamu lakukan.



- Membaca buku, — kamu membalikkan halaman buku setiap selesai membaca halaman tersebut. Berangkat ke sekolah, — kamu melalui jalur yang terencana setiap hari. Bermain *game*, — kamu melakukan hal yang sama berulang-ulang.
- Tidak hanya itu, perulangan juga merupakan inti dalam berhitung. Jika kamu menghitung 5×3 , apa saja langkah yang kamu lakukan untuk mendapatkan hasilnya?

Salah satu keunggulan program komputer daripada manusia ialah kemampuannya untuk mengolah data yang berukuran besar atau melaksanakan suatu aksi berulang kali dalam periode waktu yang lama tanpa merasa bosan atau lelah. Hal ini dimungkinkan dengan adanya suatu kontrol perulangan. Pernyataan perulangan atau *loop* merupakan struktur program untuk keperluan iterasi, yaitu memproses satu atau beberapa pernyataan secara berulang (*looping*) berdasarkan kondisi tertentu. Program C menyediakan tiga bentuk pernyataan *loop*, yaitu:

- *for loop*
- *while loop*
- *do...while loop*

a. Struktur Kontrol Perulangan *For-Loop*

Pernyataan ini umumnya digunakan untuk memproses pernyataan secara berulang-ulang, dengan jumlah perulangan yang dilakukan telah diketahui sebelumnya. Misalnya, berjalan sebanyak n langkah ke depan, atau mencetak barisan dari suku pertama hingga suku ke- n . Struktur kontrol perulangan *for* ialah sebagai berikut.

```
for (expr1; expr2; expr3)
{
<statement>;
...
}
```

Struktur di atas akan dijalankan melalui proses berikut.

- Ekspresi *expr1* akan dieksekusi ketika program menjalankan struktur *for* tersebut. Ekspresi ini biasanya berisi inisialisasi suatu variabel *counter* yang digunakan untuk menghitung jumlah perulangan yang telah dilakukan.



- Ekspresi `expr2` ialah suatu ekspresi bernilai benar atau salah (*boolean*) yang akan dicek sebelum pernyataan di dalam blok struktur dieksekusi. Apabila ekspresi ini bernilai benar, pernyataan akan dieksekusi. Sebaliknya, apabila ekspresi bernilai salah, pernyataan tidak akan dieksekusi dan perulangan berakhir. Dengan kata lain, pada bagian ini, kamu menuliskan sebuah pernyataan yang merupakan kondisi berhenti (*stopping criteria*) untuk memastikan perulangan yang kamu buat memiliki langkah yang terbatas (dipastikan berhenti).
- Ekspresi `expr3` merupakan sebuah pernyataan yang dijalankan setelah semua pernyataan di dalam struktur *for* dieksekusi. Biasanya, pernyataan ini dibuat untuk mengubah nilai variabel *counter* yang akan makin mendekati kondisi berhenti (memastikan nilai *counter* akan konvergen ke kondisi berhenti).

Walaupun ketiga ekspresi tersebut bersifat opsional (tidak harus ada), kamu disarankan untuk menuliskan ketiga ekspresi tersebut saat membuat program dengan jelas dan lengkap. Untuk saat ini, kamu perlu memahami teknik untuk menulis ketiga ekspresi tersebut dengan baik.

Perhatikan contoh kode program menulis bilangan bulat berikut. Pada contoh tersebut, program akan menulis bilangan bulat dari 0 hingga kurang dari *n*. Telusurilah kode program tersebut dengan teknik penelusuran yang telah kamu pelajari pada saat menelusuri suatu diagram alir!

```
/*
 * Program Menulis Bilangan Bulat sebanyak n kali
 */

#include <stdio.h>
int main()
{
    int i, n;

    scanf("%d", &n);
    for (i=0; i < n; i=i+1) // i = i+1 dapat juga ditulis i++
    {
        printf("%d\n", i);
    }
    return 0;
}
```



Keluaran program tersebut saat diberi masukan 5 ialah:

0
1
2
3
4

Pada contoh tersebut, variabel *counter* yang digunakan ialah *i* yang nilainya diinisialisasi (diberi nilai awal) dengan nilai 0. Pada bahasa C, lazimnya *counter* dimulai dari nilai 0, berbeda dengan proses pencacahan yang biasa kamu lakukan dari 1 di dunia nyata. Hal ini terkait dengan beberapa aspek teknis di bahasa C yang dimulai dari 0. Misalnya, di kelas XI nanti, kamu akan menggunakan struktur data *array* yang dimulai dari indeks ke-0. Perlu diingat bahwa inisiasi dengan 0 ini hanyalah suatu kebiasaan masyarakat pemrogram dalam bahasa C, dan kamu tetap dapat melakukan pencacahan mulai dari 1. Kamu dapat mencoba mengubah kode program di atas sehingga *counter* berjalan dari 1. Selain *counter* yang berjalan menaik (*ascending*), kamu juga dapat membuat suatu *counter* yang berjalan menurun (*descending*). Contoh ini disajikan misalnya pada diagram alir keempat dan kelima pada bagian Algoritma.

Pernyataan yang ada pada *expr1* hingga *expr3* dapat ditulis menjadi deretan instruksi yang dipisahkan dengan tanda koma. Misalnya, terdapat pada contoh program berikut.

```
1  for (i=0, j=0; i<3; ++i, j++)  
2  {  
3      printf("%d %d", i, j);  
4  }
```

Urutan pengerjaan akan sama seperti pada contoh sebelumnya. Akan tetapi, ada dua pernyataan yang akan dijalankan pada *expr1* hingga *expr3*.

Praktik Baik Pemrograman

Pada saat merancang sebuah struktur perulangan *for*, kamu perlu memastikan agar kondisi berhenti pasti akan tercapai (konvergen). Apabila kondisi berhenti tidak pernah tercapai, akibatnya, struktur ini akan berjalan terus-menerus



dan menyebabkan terjadinya suatu perulangan yang tidak terbatas (*infinite loop*). Apabila hal ini terjadi, program akan dibekukan oleh sistem operasi, bahkan akan dihentikan. Perhatikan contoh berikut.

```
/*  
 * Program dengan Perulangan tak Terbatas  
 */  
  
#include <stdio.h>  
  
int main()  
{  
    int i, n;  
    scanf("%d", &n);  
    for(i=0; i<n; i--)  
        printf("%d\n", n)  
    return 0;  
}
```

Dapat dilihat bahwa pada kode tersebut, nilai *counter* akan berkurang dan tidak akan pernah melebihi nilai *n* jika *n* diisi dengan suatu bilangan bulat positif.

Ayo, Cari Tahu!

Lakukan penelusuran pada kode program tersebut dengan nilai $n = 3$.

b. Struktur Kontrol Perulangan *While*

Saat merancang perulangan, kamu bisa jadi tidak dapat menentukan berapa kali perulangan akan dilakukan. Akan tetapi, kamu mengetahui kondisi berhentinya. Misalkan, instruksi berikut pada dunia nyata, “berjalan luruslah sampai ujung jalan, kemudian belok kiri”. Instruksi tersebut tidak memberikan gambaran jelas jumlah langkah yang akan kamu lakukan. Namun, secara naluriah, kamu mengetahui kapan kamu harus berhenti berjalan lurus, lalu berbelok ke arah kiri.



Pada program, suatu struktur kontrol *while* dikenal untuk melakukan perulangan seperti pada contoh di atas. Struktur kontrol tersebut dapat ditulis sebagai berikut. Pernyataan akan dieksekusi terus-menerus selama ekspresi kondisi bernilai benar.

```
while (ekspresi kondisi) {  
    <pernyataan>;  
    ...  
}
```

Sebagai contoh, misalnya kamu akan menulis kode program untuk membaca dan menuliskan kembali bilangan bulat positif. Hal ini terus dilakukan hingga program membaca nilai -1. Program tersebut dapat kamu lihat di bawah ini.

```
/*  
 * Program Baca Tulis Bilangan  
 */  
#include <stdio.h>  
int main()  
{  
    int bilangan;  
    scanf("%d", &bilangan);  
    while (bilangan != -1) {  
        printf("%d\n", bilangan);  
        scanf("%d", &bilangan);  
    }  
    return 0;  
}
```

Ayo, Cari Tahu!

Lakukan penelusuran pada kode program ini dengan masukan yang diberikan ialah 1 2 3 4 -1



c. Struktur Kontrol Perulangan *Do – While*

Struktur kontrol *do-while* memiliki perilaku yang mirip dengan *while*, yaitu kamu hanya mengetahui kondisi berhenti dari perulangan tersebut. Perbedaannya ialah struktur *do-while* dipastikan akan dikerjakan setidaknya satu kali. Bentuk umum pernyataan *do .. while* ialah sebagai berikut.

```
do {  
    <pernyataan>;  
} while (ekspresi kondisi);
```

Salah satu contoh penggunaan struktur *do-while* ialah ketika kamu menulis sebuah program interaktif yang akan meminta pengguna memasukkan kembali suatu nilai hingga nilai tersebut memenuhi suatu syarat. Hal ini akan sering kamu alami ketika kamu diminta untuk mengisi ulang (*retry*) saat menggunakan sebuah program atau mengisi sebuah formulir elektronik.

Misal, program berikut akan terus meminta pengguna memasukkan nilai sampai pengguna tersebut memasukkan bilangan bulat positif.

```
/*  
 * Program Verifikasi Masukan Pengguna  
 */  
  
#include <stdio.h>  
  
int main()  
{  
    int bilangan;  
    do  
    scanf("%d\n", &bilangan);  
    while (!(bilangan > 0));  
  
    printf("Anda telah memasukkan bilangan bulat positif\n");  
    return 0;  
}
```



d. Struktur Kontrol Perulangan Bersarang

Sama seperti struktur kontrol percabangan, kamu dapat meletakkan struktur kontrol perulangan secara bersarang. Misalnya, pada contoh program berikut yang akan mencetak suatu pola berbentuk persegi menggunakan karakter asterisk “*”.

```
/*
 * Program dengan FOR Bersarang
 */

#include <stdio.h>
const int PANJANG = 2;
const int TINGGI = 3;

int main()
{
    int i, j;
    for (i = 0; i < TINGGI; i++) {
        for (j = 0; j < PANJANG; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

Ayo, Cari Tahu!

Lakukanlah penelusuran pada kode program tersebut!

Pada contoh di atas, *counter* pada struktur *for* terluar dan terdalam tidak saling berkaitan. Pada beberapa kasus nantinya, kedua *counter* tersebut bisa saja saling berkaitan. Misalnya, pada program berikut.

```
/*
 * Program dengan Dua Counter
 */
```



```
#include <stdio.h>
int main()
{
    int i, j, m, n;
    scanf("%d %d", &m, &n);
    for (i=1; i<=m; i++) {
        for (j=i; j<=n; j++) {
            printf("%d", j);
            if (j==n)
                printf("\n");
            else
                printf(" ");
        }
    }
    return 0;
}
```

Pada program tersebut, kamu dapat melihat bahwa nilai awal *counter* j akan berpengaruh pada nilai *counter* i. Saat dijalankan, keluaran dari program tersebut ialah:

```
1 2 3 4 5
2 3 4 5
3 4 5
```

Tentunya, tidak ada batasan jumlah struktur yang kamu buat secara bersarang. Kamu pun juga dapat memadukan struktur perulangan dengan struktur percabangan sehingga menghasilkan program yang lebih kompleks.



Aktivitas 10.2-06-U Latihan Struktur Perulangan

Buatlah solusi program untuk menyelesaikan problem-problem berikut. Kamu dapat menggunakan diagram alir, pseudokode, pemrograman visual, atau bahasa pemrograman untuk menghasilkan solusi dari problem tersebut sesuai dengan kondisi di tempatmu.



Problem 1: Menghitung Mundur (Tingkat Kesulitan: ★★)

Buatlah kode program berdasarkan Diagram Alir 4 pada bagian Algoritma untuk mencetak bilangan secara hitung mundur.

Problem 2: Menghitung Rataan (Tingkat Kesulitan: ★★★)**Deskripsi Soal:**

Buatlah sebuah program yang akan menghitung rata-rata dari n buah bilangan.

Format Masukan:

Baris pertama ialah sebuah bilangan bulat positif n yang menunjukkan banyaknya data, baris berikutnya berisi n buah bilangan bulat. Nilai n maksimal 1000, dan besarnya bilangan yang harus dihitung rata-ratanya berada pada rentang -1 miliar hingga 1 miliar.

Format Keluaran:

Nilai rata-rata dari n buah bilangan masukan. Nilai rata-rata tersebut dituliskan sebagai bilangan riil dengan dua angka di belakang titik desimal.

Masukan	Keluaran
10 20 30	20.00

Problem 3: Mencari Bilangan Terbesar (Tingkat Kesulitan: ★★★)

Buatlah kode program berdasarkan Diagram Alir 5 pada bagian Algoritma untuk mencari bilangan terbesar dari sekumpulan bilangan yang diberikan.

Problem 4: Membuat Mesin Sortir Kembang Kol (Tingkat Kesulitan: ★★★★★)**Deskripsi Soal:**

Kamu akan membantu seorang petani kembang kol untuk menyortir kembang kol yang telah dipanen berdasarkan ukurannya. Kembang kol tersebut akan dikelompokkan menjadi berukuran kecil (< 50 gram per buah), berukuran sedang (50-200 gram per buah), dan berukuran jumbo (> 200 gram per buah). Selama ini, petani tersebut menyortir kembang kol menggunakan tenaga manusia. Karena kamu telah memiliki kemampuan untuk membuat program yang dapat melakukan hal tersebut secara otomatis, kamu pun mengajak petani tersebut untuk menerapkan *future practice* dengan menerapkan otomatisasi. Bantulah



petani tersebut dengan program buatan kamu! Ini akan menjadi langkah awal bagi petani tersebut untuk menjadi seorang petani modern.

Format Masukan:

Program kamu akan membaca berat dari setiap kembang kol dalam satuan gram. Berat setiap kembang kol ditulis dalam bilangan riil. Apabila semua kembang kol telah tersortir, program kamu akan membaca nilai -1.

Format Keluaran:

Tiga buah bilangan bulat yang merupakan jumlah kembang kol pada setiap kategori, yaitu kecil, sedang, dan jumbo. Setiap bilangan ditulis pada baris yang berbeda.

Masukan	Keluaran
100.0 20.5 300.1 40.1 -1	2 1 1

Problem 5: Menggambar Segitiga (Tingkat Kesulitan: ★★★★★)

Deskripsi Soal:

Buatlah sebuah program untuk menggambar sebuah segitiga dengan menggunakan karakter asterisk (*).

Format Masukan:

Sebuah bilangan bulat n yang menyatakan ukuran segitiga yang akan dibuat. Nilai n berada pada rentang 1 hingga 20.

Format Keluaran:

Bentuk segitiga berukuran $n \times n$ dengan menggunakan karakter asterisk (*) seperti pada contoh keluaran.

Masukan	Keluaran
7	<pre> ***** ***** ***** **** *** ** * </pre>

Problem 6: Memperbaiki Program (Tingkat Kesulitan: ★★★)



Deskripsi Soal:

Buatlah sebuah program untuk menggambar sebuah pola X dengan menggunakan karakter asterisk (*).

Format Masukan:

Sebuah bilangan bulat n yang menyatakan ukuran pola yang akan dibuat. Nilai n berada pada rentang 1 hingga 20.

Format Keluaran:

Pola huruf X yang sesuai dengan ukuran yang dimasukkan. Perhatikan perbedaan untuk banyak baris ganjil dan genap pada contoh keluaran.

Masukan	Keluaran
5	*__* _**_ __*_ _**_ *__*
6	*___* _*__* __**_ _*__* *___*

Pada soal ini, kamu telah diberikan salah satu program yang dibuat untuk menyelesaikan permasalahan di atas. Akan tetapi, program tersebut tidak dapat dijalankan dan ditulis dengan tidak rapi! Pelajarilah kode program di bawah ini, dan perbaiki program tersebut hingga dapat berjalan dengan benar. Sesuaikan pula penulisannya dengan mengikuti praktik baik yang telah kamu pelajari hingga saat ini.

5. Fungsi

Selama ini, kamu telah sering bertemu dengan fungsi. Di matematika, kamu membuat suatu persamaan menggunakan fungsi seperti $y = f(x) = x + 1$. Fungsi $f(x)$ menerima sebuah masukan x yang disebut sebagai fungsi daerah dan menghasilkan nilai y yang merupakan fungsi wilayah. Fungsi sangat berhubungan dengan kemampuan abstraksi yang telah kamu pelajari di Berpikir Komputasional sehingga program yang kamu tulis dapat ditulis dengan lebih baik.



Sejauh ini, kamu juga telah menggunakan beberapa fungsi dalam berlatih pemrograman. Pada diagram alir, kamu telah mengenal sebuah simbol subprogram untuk memberikan abstraksi dari suatu proses lain yang kamu gunakan dalam solusimu. Pada program bahasa C, struktur *main* merupakan sebuah fungsi yang akan dieksekusi oleh sistem operasi ketika program dijalankan. Selain itu, kamu pun telah menggunakan fungsi seperti *printf* dan *scanf*.

Pada hakikatnya, fungsi pada program melambangkan suatu kumpulan pernyataan yang memiliki tujuan tertentu. Tujuan tersebut direpresentasikan oleh nama dari fungsi tersebut. Misalnya, *scanf* yang memiliki fungsi untuk membaca (*scan*) nilai dari pengguna. Fungsi juga dapat menerima parameter-parameter, dan juga dapat mengembalikan suatu nilai. Dengan membungkus kumpulan instruksi tadi ke dalam suatu fungsi, kamu dapat menggunakan kembali fungsi tersebut di berbagai lokasi dalam programmu.

a. Membuat Fungsi

Saat membuat suatu fungsi baru, kamu perlu menentukan tiga hal berikut: nama fungsi yang merepresentasikan tujuan dari fungsi, parameter yang dimasukkan ke dalam fungsi, serta nilai yang dikembalikan. Ketiga informasi ini disebut prototipe dari fungsi. Adapun pernyataan-pernyataan yang ada di dalam fungsi tersebut disebut implementasi dari fungsi.

Misal, kamu akan membuat sebuah fungsi untuk menghitung luas lingkaran. Maka, kamu perlu menetapkan ketiga hal tersebut dan menghasilkan prototipe fungsi berikut. Dari prototipe tersebut, kamu dapat melihat bahwa fungsi *hitungLuasLingkaran* memerlukan sebuah parameter dengan tipe data *float* yang merupakan radius dari lingkaran. Saat dipanggil, fungsi ini akan mengembalikan sebuah nilai *float* yang merupakan luas lingkaran.

```
float hitungLuasLingkaran(float radius);
```

Jika dituliskan lengkap dengan implementasinya, fungsi tersebut dapat ditulis menjadi:



```
float hitungLuasLingkaran(float radius){
    float luas;
    luas = 3.14 * radius * radius;
    return luas;
}
```

Pada kode program tersebut, kata kunci *return* digunakan untuk mengakhiri fungsi dan mengembalikan suatu nilai. Dalam hal ini, nilai yang dikembalikan ialah luas lingkaran.

Perlu diingat bahwa prototipe dari fungsi-fungsi berikut akan dianggap sebagai fungsi yang berbeda karena memiliki parameter yang berbeda dari tipe data. Hal ini disebut sebagai *overloading*.

```
float hitungLuasLingkaran(float radius);
float hitungLuasLingkaran(int radius);
```

Kamu juga dapat membuat fungsi dengan jumlah parameter lebih dari satu, misalnya untuk menghitung luas persegi panjang berikut:

```
float hitungLuasPersegiPanjang(float panjang, float lebar);
```

Tentunya, kamu juga dapat membuat fungsi yang tidak memiliki parameter masukan, seperti yang kamu lakukan pada saat membuat fungsi **main()** pada program.

b. Memanggil Fungsi

Setelah dibuat, fungsi dapat dipanggil (*function call*) di dalam kode program. Perhatikan kode program berikut, yang akan memanggil fungsi *hitungLuasLingkaran* yang telah dibuat.

```
/*
 * Program Hitung Luas Lingkaran
 */

#include <stdio.h>

float hitungLuasLingkaran(float radius){
    float luas;
```



```

    luas = 3.14 * radius * radius;
    return luas;
}

int main(){
    float radius, luas;
    scanf("%f", &radius);
    luas = hitungLuasLingkaran(radius);
    printf("%f\n", luas);
    return 0;
}

```

Kamu juga dapat memanggil suatu fungsi di fungsi lain yang kamu buat. Misalnya, kamu ingin membuat fungsi untuk menghitung luas permukaan bola. Kamu dapat menulisnya menjadi:

```

/*
 * Program Hitung Luas Bola
 */

#include <stdio.h>

float hitungLuasLingkaran(float radius){
    float luas;
    luas = 3.14 * radius * radius;
    return luas;
}

float hitungLuasBola(float radius){
    float luas;
    volume = 4.0 * hitungLuasLingkaran(radius);
    return luas;
}

int main(){
    float radius, luasBola;
    scanf("%f", &radius);
    luasBola = hitungLuasBola(radius);
    printf("%f\n", luasBola);
}

```



```
    return 0;
}
```

c. Variabel Lokal pada Fungsi

Pada contoh-contoh di atas, kamu akan menemukan adanya deklarasi variabel dengan *identifier* yang sama. Misalnya, variabel luas kamu temukan di dalam fungsi *hitungLuasLingkaran* dan *hitungLuasBola*. Kedua variabel tersebut disimpan pada alamat memori yang berbeda yang hanya dapat diakses di dalam fungsi tempat variabel tersebut berada. Dengan kata lain, keduanya ialah variabel lokal yang tidak saling berhubungan.

d. Fungsi Rekursif

Pada persamaan di mata pelajaran Matematika, kamu mungkin pernah menemukan suatu fungsi yang ditulis secara rekursif seperti berikut.

$$f(n) = \begin{cases} 3 & , n = 0 \\ 2f(n-1) & , n < 0 \end{cases}$$

Pada contoh tersebut, terlihat bahwa $f(n)$ akan bernilai 3 jika nilai $n = 0$. Jika tidak, $f(n)$ akan memanggil fungsi $f(n-1)$ di dalamnya. Suatu fungsi yang memanggil dirinya sendiri disebut sebagai *fungsi rekursif*. Suatu fungsi rekursif harus memiliki dua buah bagian, yang disebut langkah basis dan langkah rekursif.

Langkah basis adalah bagian fungsi yang tidak memanggil dirinya sendiri. Pada contoh di atas, langkah basis dilakukan ketika n bernilai 0. Saat langkah basis ini dipanggil, rekursi akan selesai. *Langkah rekursif* adalah bagian fungsi yang memanggil dirinya sendiri. Pada contoh di atas, langkah rekursif dilakukan ketika n bernilai > 0 .

Pada saat merancang sebuah fungsi rekursif, kamu perlu memastikan agar langkah rekursif *harus* mencapai langkah basis. Pada contoh di atas, apabila nilai $n > 0$, langkah rekursif dipastikan akan mencapai langkah basis. Perhatikan bahwa parameter n selalu berkurang 1. Akan tetapi, jika n bernilai negatif, langkah rekursif tidak akan mencapai langkah basis. Oleh karena itu, penting bagi kamu untuk menspesifikasikan dengan jelas rentang parameter fungsi rekursif, langkah rekursif, dan langkah basis yang digunakan.



Fungsi rekursif di atas dapat ditulis seperti berikut dalam bentuk kode program.

```
/*  
 * Program dengan Fungsi Rekursif  
 */  
  
#include <stdio.h>  
int f(int n){  
    if (n == 0)           // Langkah Basis  
        return 3;  
    else  
        return 2 * f(n-1) + 4; // Langkah Rekursif  
}  
  
int main(){  
    printf("%f\n", f(5));  
    return 0;  
}
```

Praktik Baik Pemrograman

Gunakan fungsi untuk melakukan abstraksi. Kumpulkan fungsi-fungsi yang telah kamu buat agar dapat digunakan kembali untuk membuat program dengan lebih cepat. Kumpulan fungsi ini dapat kamu satukan menjadi sebuah pustaka atau *library*. Apabila pustaka tersebut memiliki manfaat yang besar dan dibutuhkan oleh banyak orang, kamu dapat membuat pustaka tersebut menjadi publik.



Aktivitas 10.2-07-UP Latihan Fungsi

Buatlah solusi program untuk menyelesaikan problem-problem berikut. Kamu dapat menggunakan diagram alir, pseudokode, pemrograman visual, atau bahasa pemrograman untuk menghasilkan solusi dari problem tersebut sesuai dengan kondisi di tempatmu.

1. Buatlah kode program dari Diagram Alir 2: Menghitung Luas Permukaan Kubus.



2. Buatlah sebuah fungsi untuk menghitung luas dan keliling bangun datar, seperti persegi panjang, lingkaran, atau segitiga.
3. Buatlah sebuah fungsi untuk menghitung luas permukaan bangun ruang seperti balok, kerucut, bola, dan limas.
4. Buatlah sebuah fungsi rekursif untuk menghitung faktorial.



Aktivitas 10.2-08-UP

Membuat Kasus Uji

Misalkan, kamu akan membuat suatu algoritma untuk menghitung akar dari persamaan kuadrat dengan menggunakan rumus berikut:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Jawablah pertanyaan-pertanyaan berikut.

1. Apakah yang menjadi masukan, proses, dan keluaran dari algoritma tersebut?
2. Apa saja kondisi yang dapat menyebabkan rumus di atas tidak dapat dikerjakan?

Berdasarkan jawaban kamu, lengkapi tabel rancangan kasus uji berikut.

Masukan			Luaran / Perilaku Program
a	b	c	
1	3	-4	-4 dan 1
...	...	...	...
...	...	...	...
...	...	...	Tidak dapat dihitung karena melibatkan akar bilangan negatif.
...	...	...	Tidak dapat dihitung karena melibatkan pembagian dengan nol.

Dari hasil di atas, menurut kamu, apakah diperlukan adanya pengecekan terhadap nilai-nilai yang dimasukkan pada formula untuk memastikan



algoritma yang kamu rancang dapat berjalan dan menghasilkan keluaran yang benar?

C. Memilih Algoritma untuk Masalah di Kehidupan Nyata

Dalam kehidupan sehari-hari, kamu akan menemui berbagai permasalahan yang perlu kamu selesaikan. Untuk mencari solusi yang terbaik dari permasalahan tersebut, kamu perlu memilih dari berbagai pilihan langkah yang dapat digunakan. Begitu juga dengan algoritma yang digunakan dalam menyelesaikan permasalahan komputasional. Dalam menyelesaikan permasalahan tersebut, kamu perlu mencari dan memilih algoritma yang paling tepat dan sesuai untuk menyelesaikannya.

1. Pencarian (*Searching*)

Hidup ialah pencarian yang tiada henti. Mari, kita berpikir ke pengalaman “mencari” dalam kehidupan sehari-hari, misalnya seperti berikut ini.

- Pernahkah kamu merasa kebingungan saat mencari sebuah buku di lemari bukumu? Atau, bahkan, di perpustakaan sekolahmu? Saat kamu meminta bantuan kepada petugas, mengapa dia dapat menemukan buku yang kamu cari dengan waktu yang lebih singkat?
- Suatu hari, kamu kehilangan baju yang kamu butuhkan dan kamu mencarinya. Apa strategimu supaya baju tersebut cepat ditemukan?
- Kamu mengingat sebuah potongan lirik lagu, tetapi tidak ingat judul lagu tersebut. Bagaimana kamu dapat menemukan lagu tersebut dengan cepat?

Apa itu *mencari*? Mencari adalah usaha menemukan “sesuatu” yang dapat berupa benda, angka, konsep, informasi yang memenuhi kriteria tertentu dalam suatu ruang pencarian. Masalah pencarian sangat umum ditemukan di dalam kehidupan, termasuk dalam dunia komputasi. Ketika melakukan suatu pencarian, kamu harus menemukan suatu benda atau objek yang memenuhi kriteria tertentu dari sekumpulan benda atau objek lain. Beberapa contoh dari masalah pencarian yang sering kamu temui ialah sebagai berikut.

- Mencari buku dengan judul tertentu di rak buku perpustakaan.
- Mencari pakaian batik seragammu di lemari yang berisi semua pakaian yang kamu miliki.
- Mencari dokumen atau web tertentu dengan mesin pencari seperti Google.



Mencari benda nyata gampang, tinggal kamu lihat dan cocokkan dengan mata. Namun, mencari informasi atau konsep yang tidak kelihatan? Hmmmm... Tidak mudah!



(a)



(b)

Gambar 2.2 Pencarian (a) buku di perpustakaan, (b) informasi di internet.

Sumber: (a) Dokumen Kemdikbudristek, (b) Anthony Shkraba/Pexels.com

Masalah pencarian dapat dibuat dalam bentuk yang lebih formal agar dapat diterapkan pada banyak kasus. Elemen pada masalah pencarian seperti berikut.

1. Sekumpulan benda atau objek.
2. Kriteria dari benda atau objek yang dicari.
3. Pengecekan benda atau objek, untuk memeriksa apakah ia memenuhi kriteria pencarian.

Pertanyaan selanjutnya ialah bagaimana *strategi* untuk mencari. Tentunya, akan ada banyak cara yang dapat kita lakukan. Misalnya, kamu dapat mengambil pakaian secara acak dan mengecek apakah pakaian tersebut ialah seragam batik. Jika tidak, kamu akan mengembalikan pakaian tersebut kembali ke lemari dan kembali mengambil sebuah pakaian secara acak. Tentu ada cara lain, misalnya dengan memeriksa pakaian dari yang berada paling atas ke paling bawah. Atau, jika pakaian tersimpan dalam keadaan tergantung, kamu dapat mengecek dari pakaian paling kiri ke paling kanan.

Tentunya, ada banyak strategi lain yang dapat kamu gunakan. Ada strategi yang lebih baik daripada strategi yang lain bergantung pada keadaan benda atau objek tersebut saat pencarian dilakukan. Tentunya, kamu akan lebih mudah mencari suatu buku dengan judul tertentu di lemari perpustakaan yang tersusun



rapi dengan aturan tertentu dibandingkan dengan mencarinya di sebuah lemari yang berantakan.

2. Pengurutan (*Sorting*)

Saat merapikan sesuatu, misalnya koleksi buku, kamu menyusun buku tersebut dengan menggunakan suatu aturan. Misalnya, jika kamu memiliki koleksi buku cerita berseri, kemungkinan besar, kamu akan menyusunnya secara berurut dari volume pertama hingga volume yang terbaru. Atau, ketika sedang berbaris, kamu diminta untuk membentuk barisan berdasarkan tinggi badan. Hal-hal tersebut merupakan sebuah proses pengurutan atau *sorting*. Proses pengurutan akan menjadi bagian yang tidak terpisahkan dari program komputer atau aplikasi yang sering digunakan. Pada aktivitas ini, kamu akan melihat bagaimana proses pengurutan dapat dilakukan dengan menggunakan berbagai strategi. Pelajarilah strateginya!

Pengurutan ialah suatu permasalahan klasik pada komputasi yang dilakukan untuk mengatur agar suatu kelompok benda, objek, atau entitas diletakkan mengikuti aturan tertentu. Urutan yang paling sederhana misalnya mengurutkan angka secara terurut menaik atau menurun.

Biasanya, masalah pengurutan terdiri atas sekumpulan objek yang disusun secara acak yang harus diurutkan. Setelah itu, secara sistematis, posisi objek diperbaiki dengan melakukan pertukaran posisi dua buah objek. Hal ini dilakukan secara terus-menerus hingga semua posisi objek benar.

Misal, kamu memperoleh 5 buah angka acak berikut:



Kamu dapat membuat angka tersebut terurut menaik dengan melakukan satu kali pertukaran, yaitu dengan menukar nilai 4 dengan nilai 3. Terdapat 2 langkah penting dalam melakukan sebuah pengurutan. Langkah pertama ialah melakukan *pembandingan*. Untuk melakukan pengurutan, dipastikan ada dua buah nilai yang dibandingkan. Pembandingan ini akan menghasilkan bilangan yang lebih besar dari, lebih kecil dari, atau memiliki nilai sama dengan sebuah bilangan lainnya. Langkah kedua ialah melakukan penempatan bilangan setelah melakukan pembandingan. Penempatan bilangan ini dilakukan



setelah didapatkan bilangan lebih besar atau lebih kecil (bergantung pada pengurutan yang digunakan).

Terdapat beberapa teknik (algoritma) untuk melakukan pengurutan seperti *bubble sort*, *insertion sort*, *quick sort*, *merge sort*, dan *selection sort*. Penjelasan untuk setiap teknik sebagai berikut.

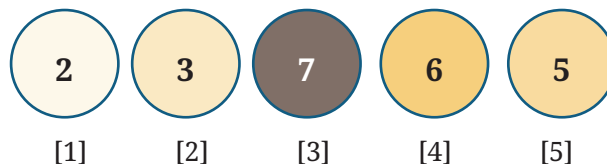
a. *Bubble Sort*

Algoritma *bubble sort* merupakan algoritma yang sudah cukup tua dalam dunia pengurutan. Sesuai namanya, algoritma ini mengilustrasikan bilangan yang diurutkan sebagai sebuah gelembung (*bubble*) yang akan terus naik sampai mencapai tempat atau urutan yang semestinya. *Bubble sort* bekerja dengan menukar elemen yang berdekatan secara berulang hingga mencapai urutan yang semestinya. Jika ada n data bilangan, berikut ialah langkah-langkah yang terdapat pada algoritma *bubble sort* (urut menaik).

1. Bandingkan nilai data ke-1 dan data ke-2. Jika data ke-1 lebih besar daripada data ke-2, tukar posisinya.
2. Ulangi langkah no. 1 pada bilangan berikutnya, yaitu data ke-2 dan ke-3, hingga data ke- $(n-1)$ dan ke- n (jika selesai sampai data ke- n , satu putaran selesai dan nilai data tertinggi akan berada di ujung kanan).
3. Ulangi langkah no. 1 dan 2, untuk n data yang dikurangi 1, yaitu dari $n-1$, $n-2$, dst sampai $n=2$ (sampai habis).

Contoh Eksekusi *Bubble Sort*

Terdapat sebuah deret bilangan seperti berikut: 2, 3, 7, 6, 5. Urutkan bilangan tersebut secara menaik dengan menggunakan algoritma *bubble sort*.



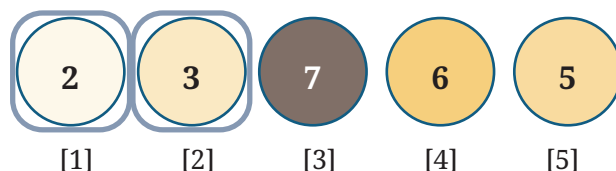
Proses Iterasi (Putaran) Pertama (Empat Kali Perbandingan)

Ambil bilangan pertama dan kedua, lalu bandingkan. Pada kasus ini, bilangan pertama ialah 2 dan bilangan kedua ialah 3. Karena hasil akhir



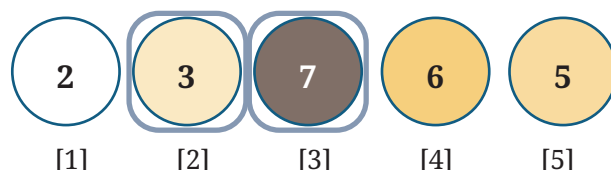
yang diharapkan ialah bilangan yang terurut naik (*ascending*), bilangan yang lebih besar harus berada di sebelah kanan.

Langkah pertama, bandingkan bilangan *pertama* dan *kedua*, yaitu 2 dan 3. Didapatkan 2 lebih kecil dari 3. Maka, urutan bilangan tersebut tetap (2,3).



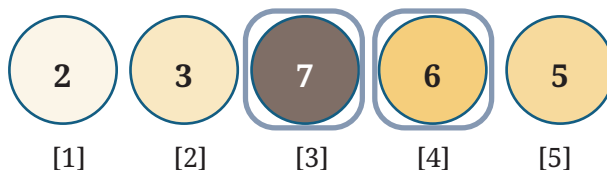
(2, 3, 7, 6, 5) menjadi (2, 3, 7, 6, 5), (tetap, tidak ada pertukaran posisi).

Langkah selanjutnya ialah membandingkan nilai bilangan *kedua* dan *ketiga*, yaitu 3 dan 7. Karena 3 lebih kecil daripada 7, urutan tetap.

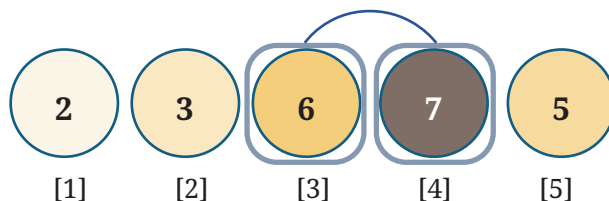


(2, 3, 7, 6, 5) tetap menjadi (2, 3, 7, 6, 5).

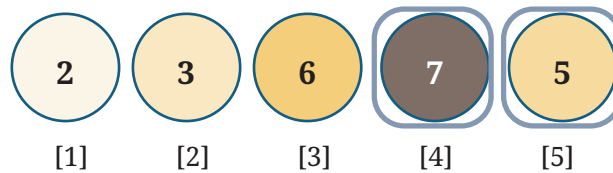
Langkah selanjutnya ialah membandingkan bilangan *ketiga* dan *keempat*. Didapatkan bahwa 7 lebih besar daripada 6. Oleh karena itu, urutan akan ditukar.



(2, 3, 7, 6, 5) menjadi (2, 3, 6, 7, 5).

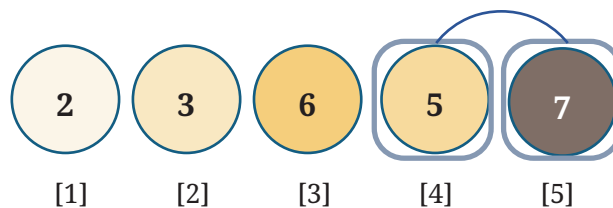


Langkah selanjutnya ialah membandingkan bilangan *keempat* dan *kelima*. Didapatkan fakta bahwa 7 lebih besar daripada 5. Oleh karena itu, urutan akan ditukar.



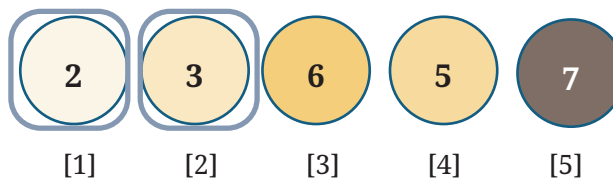
(2, 3, 6, 7, 5) menjadi (2, 3, 6, 5, 7).

Sampai tahap ini, urutan (2, 3, 7, 6, 5) telah menjadi (2, 3, 6, 5, 7). Tentu belum memenuhi pengurutan yang diinginkan. Oleh karena itu, akan dilakukan proses pengurutan kembali mulai dari bilangan pertama, dengan cara seperti di atas. Satu proses perbandingan bilangan pertama sampai dengan bilangan terakhir disebut sebagai satu *iterasi*.

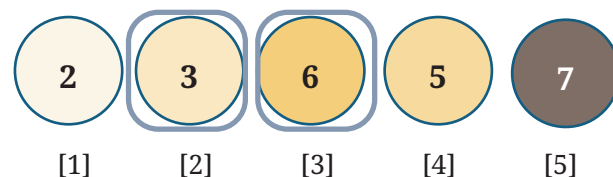


Proses Iterasi Kedua (Tiga Kali Perbandingan)

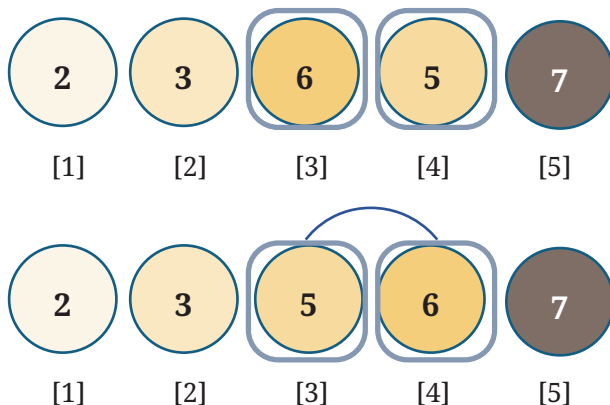
(2, 3, 6, 5, 7) menjadi (2, 3, 6, 5, 7), (tetap, tidak ada pertukaran)



(2, 3, 6, 5, 7) menjadi (2, 3, 6, 5, 7), (tetap tidak ada pertukaran)



(2, 3, 6, 5, 7) menjadi (2, 3, 5, 6, 7), terjadi **pertukaran**.



Proses Iterasi Ketiga (Dua Kali Perbandingan)

(2, 3, 5, 6, 7) menjadi (2, 3, 5, 6, 7), (tetap, tidak ada pertukaran)

(2, 3, 5, 6, 7) menjadi (2, 3, 5, 6, 7), (tetap, tidak ada pertukaran)

Proses Iterasi Keempat (Satu Kali Perbandingan)

(2, 3, 5, 6, 7) menjadi (2, 3, 5, 6, 7), (tetap, tidak ada pertukaran)

Proses pengurutan akan berhenti di tahap ini. Hasil pengurutan secara menaik (*ascending*) sudah dipenuhi. Jika banyak bilangan dan urutan awalnya berbeda, banyak iterasi yang dibutuhkan dapat berbeda.

b. *Insertion Sort*

Algoritma *insertion sort* ialah salah satu algoritma yang digunakan untuk permasalahan pengurutan dalam lis (daftar objek). Sesuai namanya, *insertion sort* mengurutkan sebuah lis dengan cara menyisipkan elemen satu per satu sesuai dengan urutan besar kecilnya elemen hingga semua elemen menjadi lis yang teratur. Misalnya, dalam kasus mengurutkan elemen lis dari yang terkecil hingga terbesar (*ascending*), tahap pertama ialah membaca suatu elemen dengan elemen yang berdekatan. Apabila elemen yang berdekatan dengan elemen saat ini lebih kecil, elemen yang lebih kecil akan ditukar dengan elemen yang lebih besar dan dibandingkan kembali dengan elemen-elemen sebelumnya yang sudah teratur. Apabila elemen saat ini sudah lebih besar daripada elemen sebelumnya, iterasi berhenti. Hal ini dijalankan satu per satu hingga semua lis menjadi teratur.



Berikut ini langkah-langkah algoritma *insertion sort*.

1. Asumsikan ialah nomor iterasi.
2. Ambil bilangan ke- $i+1$.
3. Bandingkan dengan bilangan sebelumnya.
4. Jika bilangan sebelumnya lebih besar, lakukan penukaran, lalu bandingkan lagi dengan bilangan sebelumnya.

Lakukan terus-menerus hingga ditemukan bilangan sebelumnya yang lebih kecil atau hingga sudah sampai di awal lis.

Ilustrasi *Insertion Sort*

Terdapat sebuah deret bilangan seperti berikut: 2, 3, 7, 6, 5 yang direpresentasikan dengan menggunakan kartu. Urutkan bilangan tersebut secara menaik dengan menggunakan algoritma *insertion sort*.

2	3	7	6	5
---	---	---	---	---

Proses Iterasi Pertama

Langkah pertama, bandingkan bilangan *pertama* dan *kedua*, yaitu 2 dan 3. Didapatkan 2 lebih kecil daripada 3. Maka, urutan bilangan tersebut tetap (2,3).

(2, **3**, 7, 6, 5) tetap (2, **3**, 7, 6, 5).

2	3	7	6	5
---	---	---	---	---

Catatan: bilangan yang dicetak tebal ialah bilangan yang telah diperiksa.

Proses Iterasi Kedua

Pada iterasi selanjutnya, ambil bilangan *ketiga*, yaitu 7. Lalu, bandingkan dengan bilangan sebelumnya. Karena 3 lebih kecil daripada 7, urutan tetap.

(2, 3, **7**, 6, 5) tetap (2, 3, **7**, 6, 5).

2	3	7	6	5
---	---	---	---	---

Proses Iterasi Ketiga

Pada iterasi selanjutnya, ambil bilangan *keempat*, yaitu 6. Lalu, bandingkan dengan bilangan sebelumnya. Didapatkan bahwa 7 lebih besar daripada



6. Oleh karena itu, selanjutnya, bandingkan dengan bilangan-bilangan sebelumnya, lalu menukarnya apabila bilangan tersebut lebih besar. Pertama, bandingkan 6 dengan 7. Apakah 6 lebih kecil daripada 7? Karena iya, tukar 6 dengan 7. Lalu, bandingkan lagi dengan bilangan sebelumnya, yaitu 3. Apakah 6 lebih kecil daripada 3? Karena 6 tidak lebih kecil daripada 3, maka 6 sudah berada pada posisi yang benar, yaitu sebelum 7 dan setelah 3.

Proses memindahkan 6 di antara 3 dan 7 ini biasa disebut penyisipan (*insertion*) sehingga nama algoritma ini disebut *insertion sort*.

(2, 3, 7, 6, 5) menjadi (2, 3, 6, 7, 5).

2	3	6	7	5
---	---	---	---	---

Proses Iterasi Keempat

Pada iterasi selanjutnya, kamu mengambil bilangan kelima, yaitu 5. Didapatkan bahwa 7 lebih besar daripada 5. Oleh karena itu, selanjutnya, kamu akan membandingkan dengan bilangan-bilangan sebelumnya, lalu menukarnya apabila bilangan tersebut lebih besar. Pertama, kamu akan membandingkan 5 dengan 6. Apakah 5 lebih kecil daripada 6? Karena iya, kamu akan menukar 5 dengan 6. Setelah itu, kamu akan mengecek dengan bilangan sebelumnya lagi, yaitu 3. Apakah 5 lebih kecil daripada 3? Karena 5 tidak lebih kecil daripada 3, maka 5 sudah pada posisi seharusnya, yaitu setelah 3 dan sebelum 6. Terjadi lagi proses penyisipan kartu 5 di antara 3 dan 6. Akibatnya, urutan berubah dari (2, 3, 6, 7, 5) menjadi (2, 3, 5, 6, 7).

2	3	5	6	7
---	---	---	---	---

Karena semua elemen telah terurut, proses pengurutan berhenti.

c. *Selection Sort*

Selection sort merupakan algoritma pengurutan yang cukup sederhana. Algoritma ini mencari (menyeleksi) bilangan terkecil/terbesar, bergantung pada urut naik atau turun, dari daftar bilangan yang belum terurut. Kemudian, bilangan tersebut diletakkan dalam daftar bilangan baru yang dijaga keterurutannya. Algoritma ini membagi daftar bilangan menjadi dua bagian, yaitu bagian terurut dan bagian yang belum terurut. Bagian yang terurut di sebelah kiri dan bagian yang belum terurut di sebelah kanan.



Awalnya, semua elemen bilangan dalam daftar ialah bagian yang belum terurut, dan bagian yang terurut kosong.

Berikut ini langkah-langkah yang terdapat pada algoritma *selection sort*.

1. Cari bilangan terkecil yang ada pada bagian belum terurut.
2. Tukar bilangan tersebut dengan bilangan pertama bagian belum terurut, lalu masukkan ke bagian terurut.
3. Ulangi langkah 1 dan 2 sampai bagian yang belum terurut habis.

Ilustrasi urutan *selection sort* dapat dilihat pada tabel berikut.

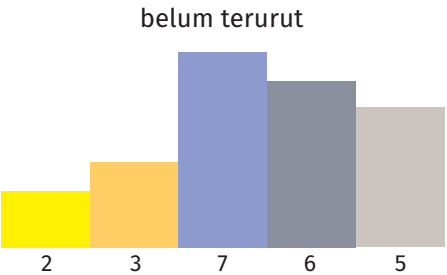
Bagian Terurut	Bagian yang Belum Terurut	Nilai Terkecil dari Bagian Belum Terurut
0	(2,3,7,6,5)	2
(2)	(3,7,6,5)	3
(2,3)	(7,6,5)	5
(2,3,5)	(6,7)	6
(2,3,5,6)	(7)	7
(2,3,5,6,7)	0	

Secara rinci, algoritma *selection sort* yang dikaitkan dengan pemrograman dijelaskan sebagai berikut.

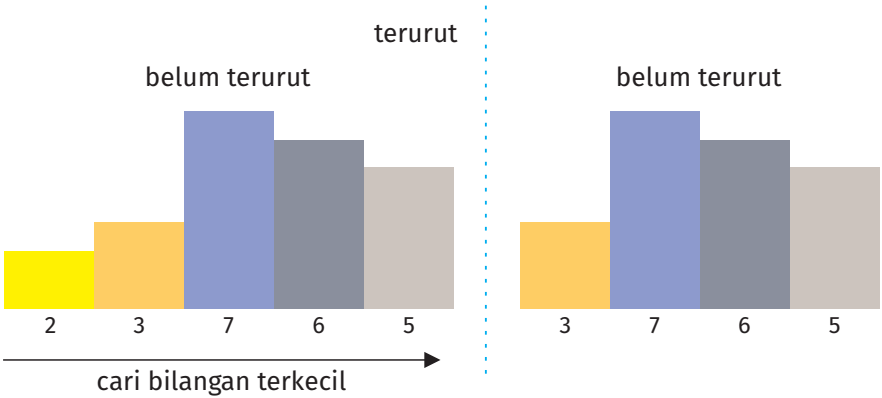
Terdapat sebuah daftar bilangan tidak terurut seperti berikut: 2, 3, 7, 6, 5. Urutkan bilangan tersebut secara menaik dengan menggunakan algoritma *selection sort*.

Proses Iterasi Pertama

Data Awal:



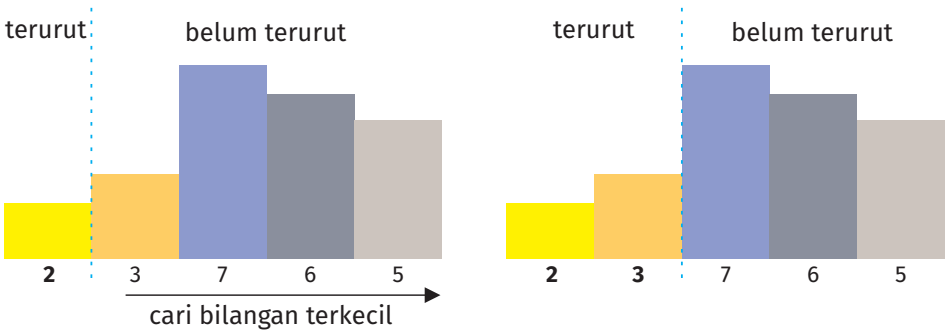
Cari bilangan terkecil di bagian belum terurut: ditemukan 2 sebagai bilangan terkecil.



Tukar bilangan 2 dengan bilangan pertama bagian belum terurut. Geser batas bagian yang sudah terurut ke kanan sehingga 2 menjadi bagian yang sudah terurut. Dalam ilustrasi ini, angka yang dicetak tebal menunjukkan *bilangan yang sudah terurut*.

Proses Iterasi Kedua

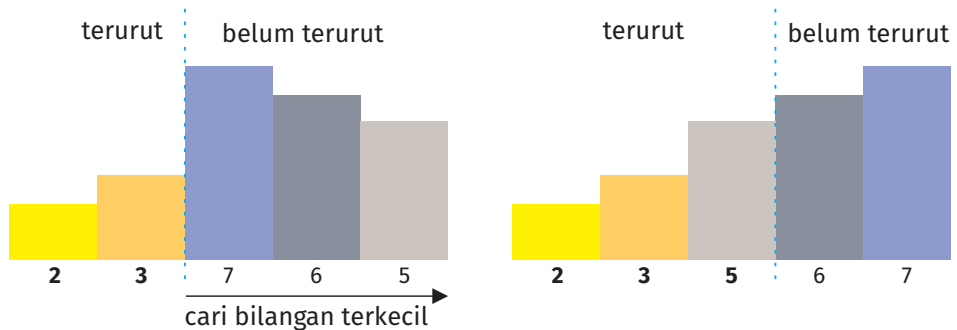
Cari bilangan terkecil di bagian belum terurut, ditemukan angka 3 sebagai bilangan terkecil.



Tukar bilangan 3 dengan bilangan pertama bagian belum terurut. Geser batas bagian yang sudah terurut ke kanan sehingga 3 menjadi bagian yang sudah terurut.

Proses Iterasi Ketiga

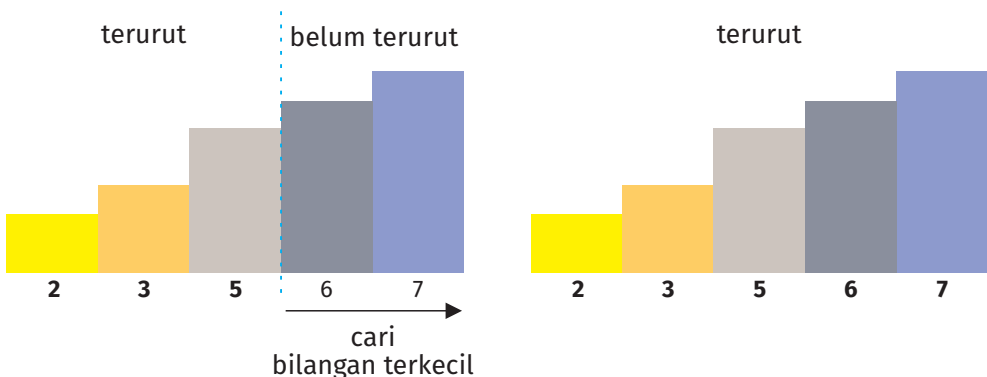
Cari bilangan terkecil di bagian belum terurut, ditemukan angka 5 sebagai bilangan terkecil.



Tukar bilangan 5 dengan bilangan pertama bagian belum terurut, yaitu 7. Geser batas bagian yang sudah terurut ke kanan sehingga 5 menjadi bagian yang sudah terurut.

Proses Iterasi Keempat

Cari bilangan terkecil di bagian belum terurut, ditemukan angka 6 sebagai bilangan terkecil. Tukar bilangan 6 dengan bilangan pertama bagian belum terurut. Di bagian akhir karena data tinggal dua, setelah proses penukaran, algoritma telah selesai dilaksanakan.





Aktivitas 10.2-09-U

Mencari dan Mengurutkan

Identifikasilah hal-hal di kehidupan kamu yang melibatkan proses mencari, proses mengurutkan, atau keduanya sekaligus. Algoritma pengurutan apakah yang selama ini kamu gunakan?



Aktivitas 10.2-10-UP

Tebak Angka

Untuk memahami masalah pencarian, kamu akan bermain tebak angka. Pada saat bermain, pahami permainan tersebut dan identifikasi aspek-aspek masalah pencarian pada permainan tersebut. Carilah strategi terbaik untuk menemukan angka yang dimiliki oleh temanmu dengan jumlah pengecekan sesedikit mungkin.

Skenario Permainan

Pada permainan ini, kamu harus berpasangan dengan salah seorang teman. Temanmu akan memilih sebuah angka bilangan bulat antara 1 – 100 (inklusif, angka 1 dan 100 juga boleh dipilih), dan angka tersebut akan dia rahasiakan. Tugas kamu ialah menemukan angka tersebut.

Untuk menemukan angka tersebut, kamu harus mengecek apakah angka tebakanmu ialah angka yang dimiliki oleh temanmu. Kamu hanya dapat mengecek angka satu per satu dengan menyebutkan angka tebakan kamu tersebut.

Setiap kali kamu menebak, temanmu harus menjawab satu dari tiga kemungkinan berikut.

- “Benar”, apabila angka yang kamu tebak sama dengan angka yang dimiliki temanmu.
- “Angka milikku lebih kecil”, apabila angka yang dimiliki temanmu lebih kecil dari tebakanmu.
- “Angka milikku lebih besar”, apabila angka yang dimiliki temanmu lebih besar dari tebakanmu.



Tentu saja, kamu dapat menebak angka apa pun. Namun, cari strategi yang membuat kamu dapat dengan cepat (atau dengan kata lain jumlah tebakan sesedikit mungkin) menemukan angka yang dipilih oleh temanmu.

Catatlah angka-angka yang kamu tebak dan jumlah tebakan yang kalian lakukan di lembar kerja yang disediakan. Lakukan permainan ini minimal sebanyak dua kali. Pada permainan berikutnya, kalian dapat bertukar peran.

Ilustrasi

Berikut ini ilustrasi dari permainan di atas. Pada permainan ini, Andi dan Binti bermain secara berpasangan. Andi memilih angka, sedangkan Binti harus menebak angka tersebut.

Mula-mula, Andi memilih sebuah angka antara 1 s.d. 100 (inklusif) dan

Binti mencatat tebakannya di lembar kerjanya

Binti: “50?”

Andi: “Angka milikku lebih kecil.”

Binti: “30?”

Andi: “Angka milikku lebih besar.”

Binti: “40?”

Andi: “Angka milikku lebih kecil.”

Binti: “35?”

Andi: “Benar!”

Pada percakapan di atas, terlihat bahwa Binti dapat menebak angka yang dipilih Andi dalam empat kali tebakan. Percakapan tersebut dicatat dalam lembar kerja, pada permainan ke-0.

Kamu dapat memanfaatkan tabel berikut yang digunakan untuk mencatat angka yang ditebak dan berapa kali penebakan dilakukan. Pada permainan ke-0, diberikan contoh isian dari ilustrasi yang diberikan di atas. Terlihat bahwa Andi memilih angka 49. Binti memerlukan sebanyak 4 kali penebakan untuk menebak dengan benar angka yang dipilih oleh Andi, yaitu menebak secara berurutan 50, 40, 30, dan 35 .



Jawabannya tentu bergantung pada tebakan berikutnya.

Siapa Andi?

Siapa Binti?

Permainan ke-	Tebakan ke-																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	50	40	30	35														
1																		
2																		
3																		

Apa yang kamu diskusikan?

Setelah bermain, saatnya kamu memikirkan makna permainan tersebut, dan cara kamu bermain. Beberapa poin diskusi yang akan kamu lakukan ialah seperti berikut.

1. Apakah permainan ini merupakan masalah pencarian?
2. Apabila Binti menjalankan strategi yang tepat, berapa kali jumlah maksimal tebakan yang benar-benar dia perlukan?
3. Strategi pencarian seperti apa yang kamu lakukan untuk menebak sesedikit mungkin?
4. Apakah strategimu berbeda dengan strategi yang dilakukan temanmu? Jika berbeda, apa perbedaannya?
5. Strategi paling bagus apa yang dapat kalian temukan untuk menemukan angka dengan jumlah tebakan paling sedikit?
6. Adakah cara lain untuk “mencari” angka yang ditebak?

Apa yang kamu lakukan?

Tuliskan algoritma Tebak Angka dalam bahasa Indonesia. Masukkan dalam Buku Kerja Siswa.

Tantangan

Simpan tantangan ini sampai kamu sudah dapat memprogram dari modul Pemrograman. Jika kamu terpaksa melakukan permainan tebak angka sendiri karena belajar di rumah, ayo, lakukan hal ini bersama sebuah program komputer! Program komputer akan menjadi Andi, angka yang dipikirkan akan ditentukan secara acak oleh komputer. Kamu dapat membuat program komputer yang



menjadi Andi jika kamu sudah belajar modul pemrograman. Catat hal ini sebagai salah satu hal yang akan kamu lakukan!



Aktivitas 10.2-11-UP Bermain Kartu

Skenario Permainan

1. Untuk aktivitas ini, kamu memerlukan 10 kartu yang masing-masing bertuliskan angka 1 sampai 10.
2. Aktivitas dapat dilakukan secara mandiri atau berkelompok.
3. Kamu akan diberikan sebuah kartu bertuliskan angka dari 1 - 10.
4. Kesepuluh kartu tersebut kamu kocok dan letakkan dalam bentuk barisan di atas meja. Kartu diletakkan tertutup.
5. Kamu harus dapat mengurutkan semua kartu secara menaik. Kartu yang berada di paling kiri barisan harus yang paling kecil.
6. Untuk mengurutkan, kamu harus melakukan serangkaian pertukaran kartu. Pertukaran dilakukan dengan membuka dua buah kartu. Apabila diperlukan, kamu dapat menukar posisi kedua kartu tersebut.
7. Kamu diminta untuk menyusun algoritma pertukaran yang dapat dilakukan untuk memastikan semua kartu dalam posisi terurut. Kamu dapat memilih untuk menggunakan salah satu dari tiga algoritma pengurutan yang disampaikan pada bagian konsep.

Apa yang kamu diskusikan?

Setelah bermain, saatnya memikirkan permainan tersebut dan cara kamu bermain. Berikut ini beberapa poin yang penting untuk didiskusikan.

1. Apakah permainan tadi merupakan masalah pengurutan?
2. Strategi pengurutan seperti apa yang kamu lakukan untuk melakukan pengecekan dan pertukaran sesedikit mungkin?
3. Apakah strategi kamu berbeda dengan strategi yang dilakukan oleh temanmu? Jika berbeda, apa perbedaannya?



4. Strategi paling bagus apa yang dapat kamu temukan untuk mengurutkan dengan banyaknya pertukaran paling sedikit?
5. Adakah kondisi yang membuat kamu melakukan banyak sekali pertukaran untuk mengurutkan kartu secara menaik?

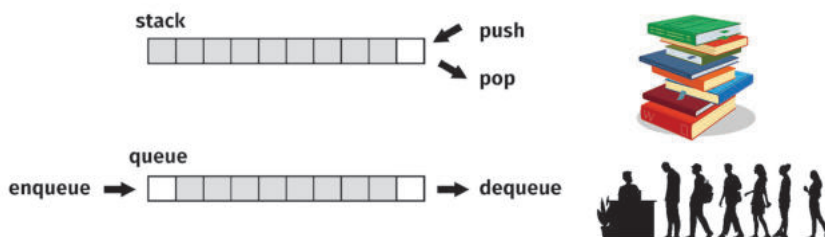
Ayo, Renungkan!

1. Apakah kamu sudah pernah melakukan permainan ini?
2. Saat mengurutkan kartu, apakah kamu senang?
3. Apakah kamu paham bahwa mengurutkan kartu itu suatu proses pengurutan?
4. Apakah kamu berhasil menemukan cara yang paling cepat untuk mengurutkan kartu tersebut?
5. Apakah kamu merasa ada masalah lain yang serupa dengan permainan tadi?
6. Pelajaran paling berkesan apa yang kamu dapatkan dari permainan ini?

Jawablah pertanyaan tersebut dalam Lembar Refleksi pada Buku Catatan, dan jangan lupa mencatat kegiatan dalam Jurnal.

D. Memilih Struktur Data untuk Masalah di Kehidupan Nyata

Berikut ini kamu akan belajar tentang hal-hal yang kamu temukan dalam keseharian, misalnya tentang tumpukan dan antrean.



Gambar 2.3 Ilustrasi struktur data antrean serta tumpukan dan contohnya di dunia nyata.

Kita akan mempelajari dua buah konsep cara penyimpanan data/objek dalam sebuah struktur yang akan menentukan urutan pemrosesan data/objek tersebut, yaitu *stack* (tumpukan) dan *queue* (antrean). Kedua konsep ini memiliki prosedur yang berbeda dalam menyimpan dan mengeluarkan data. Kedua konsep tersebut masing-masing memiliki peranan yang berbeda dan digunakan pada situasi yang berbeda pula.

Bayangkan sebuah loket di sebuah rumah sakit. Para pasien yang akan berobat diminta untuk mendaftar lebih dahulu di loket penerimaan dan mengisi formulir pendaftaran. Setelah formulir tersebut diisi, para pasien akan mengembalikan formulir ke loket dan menunggu dipanggil oleh petugas. Kebetulan, di pagi hari, dokter yang bertugas belum datang sehingga para pasien harus menunggu. Ketika sang dokter tiba, petugas loket akan memanggil para pasien satu per satu untuk mendapat layanan.

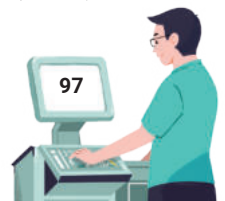
Perhatikan bagaimana urutan pasien itu dipanggil oleh petugas loket.

- Misalkan, petugas loket menumpuk formulir-formulir tersebut. Formulir yang baru diterima diletakkan *di atas* formulir yang sudah diterima sebelumnya. Kemudian, ketika memanggil pasien, petugas tersebut memanggil dengan urutan mulai dari formulir yang berada *di atas* tumpukan. Menurut kamu apakah urutan tersebut adil/sesuai dengan yang diharapkan para pasien? Mengapa?
- Bagaimana cara petugas menyusun tumpukan formulir dan/atau cara urutan memanggil para pasien dari tumpukan formulir sedemikian rupa sehingga pasien yang datang dan mengisi formulir lebih dulu, akan *dipanggil lebih dulu* juga (dan sebaliknya)?

Dalam dunia komputasi/informatika, terkadang, kamu perlu untuk menyimpan data/objek dalam suatu urutan tertentu, untuk kemudian/sewaktu-waktu diambil/dikeluarkan kembali, mungkin untuk diproses lebih lanjut atau untuk tujuan-tujuan lain. Ada dua cara utama kamu dapat melakukan penyimpanan ini.

1. Antrean (*Queue*)

Pada metode ini, objek-objek disimpan dalam metode penyimpanan yang berupa sebuah antrean sehingga objek yang *pertama/lebih dahulu datang*, juga akan *lebih dahulu keluar/selesai*, layaknya sebuah antrean di loket, pintu masuk, dll. Prinsip ini disebut juga sebagai prinsip *First In First Out (FIFO)*.



Dalam sebuah antrean orang, misalnya, jelas orang yang pertama datang akan berada di depan antrean, dan harus menjadi yang pertama yang mendapat pelayanan.

2. Tumpukan (*Stack*)

Pada metode ini, objek-objek disimpan dalam metode penyimpanan yang menyerupai sebuah tumpukan (misal: tumpukan piring). Dengan demikian, objek yang *pertama/lebih dahulu disimpan*, justru akan menjadi yang terakhir keluar. Prinsip ini disebut juga *Last In First Out (LIFO)*. Dalam tumpukan piring, misalnya, piring pertama yang diletakkan, akan berada di posisi paling bawah, dan jika kamu ambil piring satu per satu dari tumpukan itu, tentunya piring yang berada di posisi paling bawah tersebut akan menjadi yang terakhir diambil.

Baik dalam kehidupan sehari-hari maupun dalam dunia informatika, kedua konsep urutan penyimpanan data tersebut memiliki peran dan kegunaan masing-masing. Ada permasalahan-permasalahan/situasi di mana antrean (FIFO) lebih cocok digunakan, dan sebaliknya ada juga permasalahan-permasalahan di mana tumpukan (LIFO) lebih tepat diterapkan. Untuk lebih memahami kedua konsep ini dan bagaimana mereka digunakan, mari, kamu lakukan beberapa aktivitas di bawah ini.

Aktivitas 10.2-12-U Penggunaan *Stack* dan *Queue* secara Tepat

Pada aktivitas ini, kamu akan membaca beberapa skenario kondisi, baik dalam dunia sehari-hari maupun dalam dunia informatika. Tugas kamu ialah memikirkan, pada setiap kondisi/skenario tersebut, manakah yang lebih tepat digunakan/lebih relevan menggambarkan situasi tersebut, apakah *stack* ataukah *queue*. Berikan penjelasan mengapa kamu memilih jawaban tersebut!

1. Di persimpangan jalan, terdapat lampu merah. Apabila lampu merah menyala, mobil-mobil yang datang ke persimpangan tersebut harus berhenti dulu. Ketika lampu berubah menjadi hijau, semua mobil perlahan-lahan berjalan kembali dalam urutan tertentu. Manakah yang lebih tepat menggambarkan situasi tersebut?



2. Ketika menjelajah web/internet, kamu menggunakan sebuah *browser* (misal Firefox, Chrome dll). Terdapat sebuah fitur yang memungkinkan kamu untuk bergerak dari satu halaman yang sudah kamu kunjungi ke halaman lainnya, yaitu dengan menekan tombol *Back* dan *Forward*. Misalnya, kamu mengunjungi halaman A, kemudian B, lalu C. Jika kamu kemudian menekan tombol *Back*, dari halaman C, kamu akan kembali ke halaman B. Jika kamu tekan lagi tombol *Back* (pada saat ada di B), kamu akan kembali ke A. Jika kemudian kamu tekan tombol *Forward*, kamu akan kembali halaman B. Jika kamu tekan sekali lagi tombol *Forward*, kamu akan kembali ke halaman C. Oleh karena itu, aplikasi *browser* tersebut harus menyimpan (dan mengingat) semua halaman yang sudah pernah kamu kunjungi sebelumnya (biasa disebut sebagai Riwayat atau *History*). Bentuk penyimpanan yang manakah (*stack* atau *queue*) yang paling tepat digunakan untuk menyimpan Riwayat pada *browser*?



3. Mesin *printer* bertugas untuk mencetak dokumen yang dikirimkan dari sebuah komputer. Satu buah *printer* dapat terhubung ke beberapa buah komputer sekaligus, dan semuanya dapat mengirim perintah kepada *printer* tersebut untuk mencetak dokumen yang berbeda-beda. *Printer* tersebut tentunya hanya dapat mencetak satu buah dokumen dalam satu waktu tertentu, dan mungkin membutuhkan beberapa detik/menit untuk menyelesaikan proses cetak satu dokumen. Oleh karena itu, ketika *printer* sedang sibuk mencetak sebuah dokumen dari sebuah komputer, kemudian datang permintaan mencetak dari beberapa komputer yang lain (yang berbeda), *printer* tersebut harus menyimpan dokumen-dokumen yang baru datang tersebut agar nanti dapat dicetak ketika proses pencetakan yang sedang berjalan saat ini sudah selesai. Manakah yang lebih tepat digunakan, *stack* atau *queue* untuk penyimpanan dokumen-dokumen yang sedang “menunggu giliran” untuk dicetak tadi?



4. Pada sebuah aplikasi pengolah dokumen, biasanya terdapat fasilitas untuk melakukan *Undo* dan *Redo*. Operasi *Undo* akan membatalkan langkah/tindakan terakhir yang kamu lakukan saat mengedit dokumen (misal, jika kamu menyadari ada kesalahan pada langkah terakhir kamu), sedangkan *Redo* digunakan untuk mengulang kembali operasi yang baru saja dibatalkan dengan sebuah *Undo*. Proses *Undo* dan *Redo* ini dapat dilakukan sampai dengan operasi pertama setelah sebuah dokumen dibuka/disimpan. Misalnya, terjadi rangkaian kejadian berikut.
- Budi membuka dokumen A.
 - Budi menambahkan judul pada dokumen A.
 - Budi menulis sebuah paragraf pada dokumen A.
 - Budi menambahkan sebuah tabel pada dokumen A.
 - Budi menyisipkan sebuah gambar pada dokumen A.

Apabila kemudian Budi menekan tombol *Undo*, operasi terakhir (yaitu penambahan gambar) akan dibatalkan sehingga gambar tersebut akan hilang dari dokumen. Jika kemudian Budi menekan tombol *Undo* sekali lagi, operasi terakhir sebelum itu (yaitu menambahkan tabel) juga akan dibatalkan sehingga tabel tersebut akan hilang dari dokumen. Jika kemudian Budi menekan tombol *Redo*, operasi *Undo* yang terakhir (yaitu yang menghilangkan tabel) akan dibatalkan sehingga tabel tersebut akan muncul kembali.

Jelas bahwa aplikasi perlu untuk menyimpan data-data berupa tindakan/operasi apa saja yang dilakukan oleh pengguna dari awal sampai akhir, serta efeknya terhadap dokumen, agar dapat memberikan fungsionalitas *Undo* dan *Redo* tersebut. Manakah di antara *stack* dan *queue* yang lebih tepat digunakan untuk menyimpan operasi-operasi tersebut?



Lembar Kerja Siswa

Untuk setiap kasus di atas, lakukan analisis penggunaan *stack* dan *queue* dengan mengisi LKS ini.

Persoalan	Stack	Queue	Saya Pilih.....
Persimpangan lampu merah			
Penjelajahan Internet			
Antrean permintaan <i>print</i> dokumen dalam sebuah komputer			
<i>Undo Redo</i>			



Aktivitas 10.2-13-UP Simulasi *Stack*

Pada aktivitas ini, kamu akan bermain dengan satu orang siswa lainnya. Satu orang harus berperan menjadi *Pemberi Perintah* dan satu lagi harus berperan sebagai *Simulator*. Permainan dimulai dengan Pemberi Perintah memberikan sebuah perintah simulasi (yang akan dijelaskan di bawah), dan kemudian Simulator harus menjalankan simulasi dan memberikan jawaban yang benar. Jawaban tersebut kemudian harus diperiksa oleh Pemberi Perintah dan kemudian harus dinyatakan apakah jawaban tersebut benar atau salah. Setelah itu, kedua orang bertukar peran: Simulator harus menjadi Pemberi Perintah dan Pemberi Perintah menjadi Simulator. Lakukan pertukaran ini sampai beberapa kali. Orang yang berhasil mendapatkan jawaban benar sebanyak mungkin akan menjadi pemenangnya.

Berikut ini format/bentuk perintah dan bentuk jawaban yang diinginkan. Kita asumsikan ada sebuah *stack* yang mampu menyimpan nilai-nilai bilangan. Setiap perintah simulasi berisi kumpulan dari 2 buah perintah:

- PUSH X
- POP



Perintah PUSH digunakan untuk menyimpan nilai ke dalam *stack*. Perintah ini harus diikuti oleh sebuah bilangan bulat X yang akan disimpan ke dalam *stack*. Perintah POP digunakan untuk mengeluarkan angka yang *berada di atas tumpukan* saat ini. Jika saat ini tumpukan kosong, perintah POP tidak memberikan efek apa-apa. Berikut ini contoh sebuah perintah simulasi dan hasilnya.

Perintah	Isi Stack	Hasil POP
PUSH 5	5	
PUSH 3	5,3	
PUSH 2	5,3,2	
PUSH 4	5,3,2,4	
POP	5,3,2	4
PUSH 6	5,3,2,6	
POP	5,3,2	6
POP	5,3	2
POP	5	3

Ketika seorang Simulator menerima sebuah perintah simulasi, dia harus memberikan jawaban berupa daftar *bilangan yang akan dikeluarkan* dari *stack*, sesuai dengan urutan perintah simulasi yang dia terima. Misalnya, pada contoh di atas, Simulator harus memberikan jawaban berupa:

4
6
2
3

Tentunya, banyaknya angka pada jawaban harus sama dengan banyaknya perintah POP yang diberikan oleh Simulator.



Lembar Kerja Siswa

Untuk permainan peran ini, dapat dipakai LKS berikut ini.

Pemberi Perintah	Catatan Simulator	Isi <i>Stack</i>	Hasil POP

Aktivitas 10.2-14-UP Simulasi *Queue*



Pada aktivitas ini, kamu akan melakukan simulasi operasi pada sebuah *queue*. Serupa dengan aktivitas sebelumnya, aktivitas ini dijalankan oleh dua orang yang akan bertugas sebagai Pemberi Perintah dan Simulator. Pemberi Perintah akan memberikan kumpulan perintah yang berisi operasi pada *queue*, sedangkan Simulator harus memberikan jawaban berupa *rangkaian isi queue* yang dihasilkan dari setiap perintah yang diberikan.

Format perintah sebagai berikut.

- ENQUEUE X: memasukkan sebuah bilangan bulat ke dalam *queue*.
- DEQUEUE: membuang/mengeluarkan bilangan yang berada pada posisi pertama antre.



Untuk setiap perintah, Simulator harus menuliskan *apa isi queue* setiap kali perintah tersebut selesai dijalankan.

Sebagai contoh pemberi perintah memberikan perintah-perintah sebagai berikut.

Perintah	Simulator Menulis Isi Queue Setelah Setiap Perintah Dijalankan	Hasil DEQUEUE
ENQUEUE 5	5	
ENQUEUE 3	5, 3	
DEQUEUE	3	5
ENQUEUE 4	3, 4	
DEQUEUE	4	3

Jika Simulator harus memberikan 5 baris jawaban berupa isi dari *queue* setelah setiap perintah dijalankan, hasilnya:

1. 5
2. 5, 3
3. 3
4. 3, 4
5. 4

Lembar Kerja Siswa

Untuk permainan peran ini, dapat dipakai LKS.

Pemberi Perintah	Catatan Simulator	Isi Queue	Hasil Enqueue



Ayo, Renungkan!

1. Apakah kamu dapat memahami dengan baik perbedaan dari konsep *stack* dan *queue*?
2. Jika diberikan sebuah kondisi di dunia nyata/informatika, dapatkan kamu menentukan apakah *stack* atau *queue* yang lebih relevan diterapkan sebagai metode penyimpanan?
3. Dapatkah kamu mencari contoh-contoh lain penerapan *stack* dan *queue* dalam kehidupan sehari-hari?
4. Apakah kamu dapat memainkan permainan simulasi *stack* dan *queue* di atas dengan baik? Apakah permainan tersebut membantu proses pemahaman kamu terhadap kedua konsep tersebut?



Aktivitas 10.2-15-UP Studi Kasus: Survei Desa

Perhatikan Problem 2 pada Aktivitas 10.2-06-UP: Latihan Struktur Perulangan. Pada problem tersebut, kamu hanya diminta untuk menghitung rata-rata dari maksimum 1.000 nilai. Sekarang, misalkan kamu akan memodifikasi program tersebut untuk mensurvei usia dari warga suatu desa. Tugas kamu tidak hanya sekadar mencetak rata-rata dari usia tersebut, tetapi juga mencetak nilai tengah (median) dari usia penduduk di desa tersebut. Menurut kamu, perbedaan paling dasar apa yang akan terjadi di program yang kamu buat?

Rancanglah program tersebut, baik dalam bentuk diagram alir, pseudokode, atau kode program yang untuk mencapai tujuan di atas. Untuk bentuk diagram alir atau pseudokode, kamu dapat menggunakan blok program *sort()* untuk mengurutkan data. Jika menggunakan kode program, kamu dapat menggunakan fungsi pengurutan bawaan bahasa pemrograman yang digunakan.



Pilihlah satu jawaban yang tepat.

1. Mengapa suatu algoritma perlu direpresentasikan dalam berbagai bentuk seperti bentuk naratif, diagram alir, atau pseudokode?
 - A. Untuk Mengubah algoritma menjadi program.
 - B. Menyusun algoritma dengan menggunakan simbol matematika.
 - C. Menggambarkan algoritma agar dapat dipahami oleh orang lain.
 - D. Menyederhanakan algoritma menjadi lebih pendek.
 - E. Membuatnya dapat dieksekusi oleh komputer.
2. Diagram alir dapat digunakan untuk merepresentasikan atau menggambarkan sebuah algoritma karena
 - A. lebih efisien daripada pseudokode
 - B. lebih mudah dipahami oleh mesin
 - C. memerlukan lebih sedikit langkah
 - D. memungkinkan sebuah algoritma divisualisasikan
 - E. mengasah aspek kreativitas
3. Benar atau salah pernyataan berikut:
pseudokode dapat dijalankan oleh komputer karena telah ditulis dengan menggunakan bahasa pemrograman.
4. Apa yang dimaksud dengan empat elemen generik dalam bahasa pemrograman prosedural?
 - A. Variabel, ekspresi, struktur kontrol percabangan, dan struktur kontrol perulangan.
 - B. Kata kunci, pernyataan, operator, dan angka.
 - C. Fungsi, kelas, objek, dan metode.



- D. Tipe data, pengulangan, kondisi, dan perbandingan.
 - E. Perkalian, pembagian, penjumlahan, dan pengurangan.
5. Mengapa pemahaman empat elemen generik dalam bahasa pemrograman prosedural penting dalam pembuatan program?
- A. Merupakan bagian utama dari bahasa pemrograman.
 - B. Hanya berlaku dalam bahasa pemrograman.
 - C. Digunakan dalam semua bahasa pemrograman.
 - D. Tidak memengaruhi pembuatan program.
 - E. Merupakan sebuah ketentuan baku dalam pemrograman.
6. Saat kamu merancang suatu algoritma untuk menyelesaikan suatu permasalahan, kegiatan apa yang akan kamu lakukan?
- A. Membuat program segera tanpa menentukan tujuan atau format masukan/keluaran.
 - B. Membaca teori pemrograman terlebih dahulu.
 - C. Menentukan tujuan program, format masukan, dan format keluaran terlebih dahulu.
 - D. Membaca semua aktivitas di buku sebelum membuat program.
 - E. Mencari kode program serupa di Internet.
7. Apa yang dimaksud dengan variabel dalam bahasa pemrograman?
- A. Sebuah perintah untuk menghentikan eksekusi program.
 - B. Sebuah struktur kontrol untuk melakukan iterasi.
 - C. Nama dari suatu lokasi memori yang digunakan untuk menyimpan data.
 - D. Sebuah tipe data untuk menyimpan nilai karakter.
 - E. Memori komputer.



8. Apa yang dimaksud dengan struktur kontrol keputusan dalam bahasa C?
- A. Sebuah tipe data yang digunakan untuk menyimpan nilai logika.
 - B. Sebuah pernyataan yang digunakan untuk menginisialisasi variabel.
 - C. Sebuah struktur yang digunakan untuk mengatur alur program berdasarkan kondisi tertentu.
 - D. Sebuah tipe data untuk menyimpan angka desimal.
 - E. Suatu struktur data yang memungkinkan penyimpanan data yang bercabang atau bersifat hirarkis.
9. Kamu ingin mencari nomor telepon temanmu. Namun, kamu tidak tahu di mana kamu menyimpannya. Kamu memutuskan untuk mencari nomor tersebut satu per satu dari awal buku telepon. Teknik pencarian ini mirip dengan
- A. *bubble sort*
 - B. *quick Sort*
 - C. *linear/sequential search*
 - D. *binary search*
 - E. *merge sort*
10. Kamu memiliki daftar angka dalam urutan acak dan ingin mengurutkannya dari angka terkecil ke angka terbesar dengan memindahkan elemen-elemen yang lebih kecil ke posisi terdepan daftar. Teknik pengurutan ini mirip dengan algoritma
- A. *bubble sort*
 - B. *quick Sort*
 - C. *insertion search*
 - D. *selection search*
 - E. *merge sort*



11. Ketika kamu mengantre di kasir supermarket, konsep apa di dalam komputer, yang paling mirip dengan cara antrean ini?
- A. *stack*
 - B. *queue*
 - C. *array*
 - D. *linked list*
 - E. *sorting*
12. Pada program pengolah kata, aksi *undo* dan *redo* menggambarkan struktur data
- A. *stack*
 - B. *queue*
 - C. *array*
 - D. *linked list*
 - E. *tree*



Pengayaan

Jika ingin belajar lebih mendalam tentang materi di atas, kamu dapat mengunjungi tautan berikut ini.

PENCARIAN (*SEARCHING*)

- Search Algorithm: <https://buku.kemdikbud.go.id/s/cmxxkmb>
- Binary Search: <https://buku.kemdikbud.go.id/s/3dyxlp>

PENGURUTAN (*SORTING*)

- Sorting Algorithm: <https://buku.kemdikbud.go.id/s/fkijda>
- <https://visualgo.net/en>





Refleksi

Pada bab ini, kamu telah mengenal konsep algoritma dan struktur data. Berangkat dari apa yang telah kamu pelajari, hal apa saja yang dapat kamu bawa ke dalam kehidupanmu sehari-hari? Adakah hal-hal yang kamu pikir dapat menjadi lebih cepat dan ringkas dilakukan dengan menerapkan konsep algoritma dan struktur data?



Teknologi dan Budaya Digital

“
Bagaimana suatu komputer dapat bekerja hingga menghasilkan suatu informasi yang berguna bagi kita?
”





Tujuan Pembelajaran

Setelah mempelajari bab ini, kamu akan mampu memahami dinamika input, proses, dan *output* pada komputer, pemanfaatan perkakas teknologi digital dengan menghargai hak atas kekayaan intelektual, mengenal profesi bidang Informatika, serta digitalisasi budaya Indonesia.

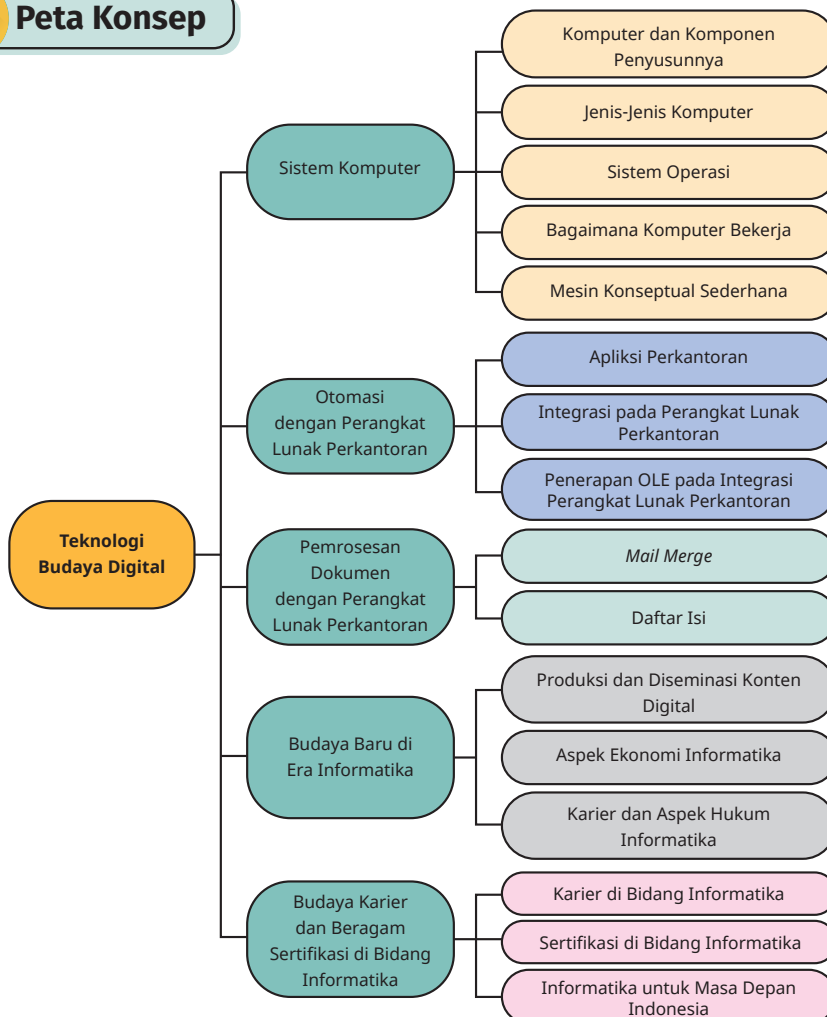


Kata Kunci

teknologi digital, budaya digital, sistem komputer, perangkat lunak, perangkat keras



Peta Konsep



Saat ini, informatika sangat memengaruhi kehidupanmu sehingga kamu harus menumbuhkan nilai-nilai berbudaya dalam berinteraksi di dunia digital. Informatika tumbuh dari sejarah perkembangan komputer di masa lampau dengan tokoh-tokoh jeniusnya. Informatika menghasilkan produk yang dapat berdampak pada aspek ekonomi maupun hukum. Di masa depan, kamu tidak boleh tertinggal untuk tahu teknologi di bidang Informatika ini, untuk membantu menyelesaikan permasalahan yang penting. Pada bab ini, kamu akan mempelajari bagaimana teknologi komputer bekerja secara umum, baik dari perangkat keras maupun perangkat lunaknya. Selain itu, kamu juga akan belajar mengenai dampak dari teknologi digital komputer terhadap kehidupan sosial manusia, melalui penggunaan teknologi komputer di dunia kerja serta berbagai jenis peluang karier yang terkait dengan teknologi komputer, pembuatan dan penyebaran konten digital, serta aspek-aspek hukum dari teknologi komputer.

Sebelum kamu memulai mempelajari berbagai hal tersebut, jawablah beberapa pertanyaan berikut.

1. Apa saja komponen-komponen dalam sebuah sistem komputer?
2. Bagaimana peran dari setiap komponen dalam sebuah sistem komputer? Bagaimana mereka dapat bekerja sama menghasilkan fungsi yang diinginkan?
3. Apakah yang dimaksud dengan sistem operasi? Apa perbedaannya dengan perangkat lunak lainnya?
4. Perangkat lunak perkantoran apa saja yang kamu ketahui? Apa saja kegunaannya?
5. Apa arti dari fitur *mail merge* pada perangkat lunak perkantoran? Bagaimana cara penggunaan fitur ini?
6. Apakah kamu tahu arti lisensi pada perangkat lunak? Apa saja jenis lisensi yang ada?
7. Apa saja jenis-jenis karier dalam bidang teknologi komputer yang kamu ketahui?

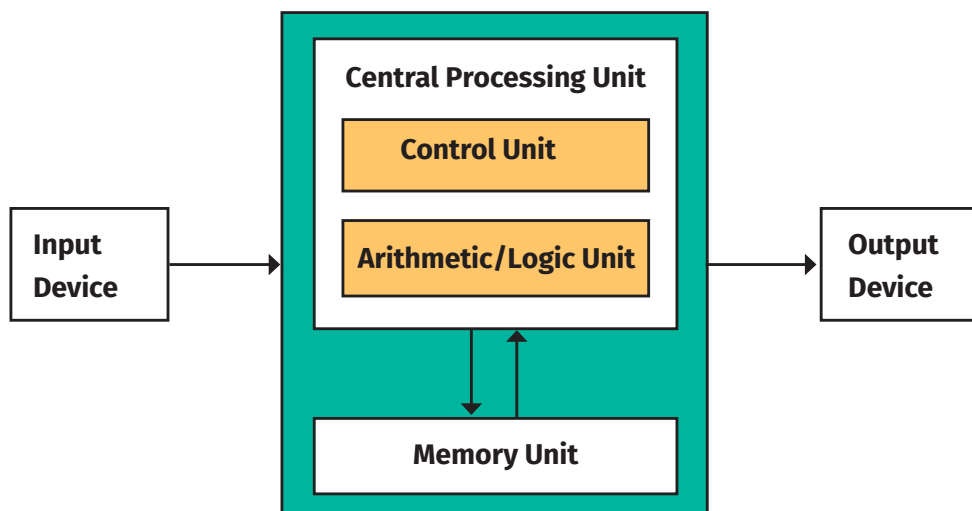


A. Sistem Komputer

Pada bagian ini, kamu akan belajar tentang komputer dan komponen penyusunnya, jenis-jenis komputer, peran sistem operasi, dan prosedur komputer bekerja.

1. Komputer dan Komponen Penyusunnya

Secara umum, komputer adalah peralatan elektronik yang mampu menerima masukan data, mengolah data, dan memberikan hasil keluaran dalam bentuk informasi, baik itu berupa gambar, teks, suara maupun video. Secara sederhana, sebuah komputer menerima masukan dari peranti masukan, memproses masukan tersebut, dan menghasilkan *output*. Agar sebuah sistem komputer dapat berjalan, setiap komponennya harus disusun menggunakan suatu arsitektur tertentu. Kebanyakan komputer saat ini menggunakan arsitektur Von Neuman. Arsitektur Von Neuman itu sendiri gambaran umumnya diberikan pada Gambar 3.1. Pada arsitektur ini, masukan melalui *input device* akan diproses oleh *Central Processing Unit* (CPU) untuk menghasilkan *output* yang direpresentasikan melalui perangkat keluaran (*Output Device*).



Gambar 3.1 Diagram blok konseptual komputer yang mengacu pada arsitektur Von Neuman.

Sumber: Dokumen Kemdikbudristek (2021)

Sistem komputer terdiri atas beberapa komponen utama, yaitu: perangkat keras (*hardware*), perangkat lunak (*software*), dan pengguna. Semua komponen tersebut saling mendukung sehingga komputer dapat beroperasi. Perangkat keras



komputer membutuhkan perangkat lunak agar komputer dapat dihidupkan dan difungsikan. Perangkat keras yang tidak disertai perangkat lunak (program), akan menjadikan komputer hanyalah sebuah mesin yang tidak berguna. Hal ini dikarenakan perangkat lunak diciptakan untuk memberikan instruksi fungsionalitas pada mesin komputer sehingga menghasilkan sebuah mesin yang memiliki fungsi dan dapat berjalan sesuai instruksi program. Melekatkan sebuah perangkat lunak pada perangkat keras komputer sering dikenal sebagai *meng-install*.

a. Perangkat Keras (*Hardware*)

Perangkat keras komputer (*hardware*) adalah komponen fisik pada komputer yang dapat disentuh, dilihat atau dipindahkan. Perangkat keras dapat berupa alat input misalnya *mouse*, *keyboard*, *scanner*. Dapat juga berupa tempat penyimpanan seperti *harddisk*, *Solid State Drive* (SSD). Ada juga yang berfungsi sebagai alat pemroses seperti *Central Processing Unit* (CPU) atau lebih dikenal dengan prosesor. Ada pula perangkat keras sebagai alat penyimpanan sementara ketika komputer sedang menyala berupa *Random Access Memory* (RAM) atau sering dikenal sebagai memori. Terakhir, ada juga perangkat keras yang berfungsi sebagai alat keluaran, contohnya *printer* dan monitor.

b. Perangkat Lunak (*Software*)

Perangkat lunak komputer (*software*) tidak terlihat secara fisik, tetapi berfungsi dan dapat dioperasikan oleh pengguna melalui antarmuka yang disediakan. Fungsinya ialah untuk menjembatani pengguna dengan perangkat keras. Perangkat lunak ialah kode-kode program yang dibuat menggunakan bahasa pemrograman. Kode-kode tersebut merupakan kumpulan perintah atau instruksi untuk menjalankan tugas tertentu sesuai dengan keinginan pengguna, atau untuk mengendalikan kerja perangkat keras. Jika sebuah sistem komputer diibaratkan manusia, perangkat keras ialah “otak” dan perangkat lunak ialah “pikiran”. Contoh perangkat lunak ialah sistem operasi, aplikasi (*app*), *driver*, dan lainnya.



c. Pengguna

Pengguna ialah orang yang menggunakan atau mengoperasikan komputer. Ketika kamu berkesempatan sedang mengoperasikan komputer, pada saat itu pulalah, kamu berperan sebagai pengguna.

2. Jenis-Jenis Komputer

Berdasarkan ukurannya, komputer dibagi menjadi beberapa jenis, yaitu antara lain seperti berikut.

a. *Microcomputer* (Komputer Mikro)

Komputer mikro merupakan komputer yang memiliki ukuran paling kecil dibandingkan dengan jenis komputer lainnya dan menggunakan *microprocessor* sebagai CPU atau unit pemrosesan utama. Contoh dari komputer mikro antara lain *ultrabook*, *game konsol*, *smartphone*, dan tablet. Karena ukuran yang kecil dan harga yang lebih murah dibandingkan dengan jenis komputer lainnya, komputer mikro paling banyak digunakan dalam kehidupan sehari-hari.

Beberapa komputer bahkan dalam bentuk papan tunggal SBC (*single board circuit*) yang berukuran kecil, misalnya yang populer ialah *raspberry pi* dan *arduino*. Raspberry Pi (Gambar 3.2), sering disingkat dengan nama Raspi, ialah SBC yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris (<https://www.raspberrypi.org/>).

Arduino ialah platform elektronik *open-source* berdasarkan perangkat keras dan perangkat lunak yang mudah digunakan, ditujukan untuk membuat proyek interaktif. Papan arduino dapat membaca input, dan menghasilkan sinyal *output* yang mengaktifasi motor, menyalakan LED atau lainnya. Arduino dapat diprogram dengan mudah. Karena kemudahan dan harganya yang murah, arduino dapat ditemui mulai dari papan 8-bit sederhana hingga produk untuk aplikasi IoT, perangkat yang dapat dikenakan, pencetakan 3D, dan *embedded system* (<https://www.arduino.cc/>).





Gambar 3.2 Contoh *Microcomputer* Raspberry Pi 4

Sumber: Michael H. ("Laserlicht")/Wikimedia Commons

b. Komputer Personal (PC, *Personal Computer*)

Komputer personal atau *Personal Computer* (PC) pada Gambar 3.3 memiliki ukuran yang lebih besar daripada komputer mikro dan memiliki kemampuan penyimpanan dan pengolahan data yang lebih besar dibandingkan dengan komputer mikro. Komputer ini dibuat untuk penggunaan personal. PC dapat berbentuk *desktop* PC (dirancang untuk diletakkan di meja), atau untuk dapat dijinjing dan dibawa-bawa (laptop).



Gambar 3.3 Contoh *Personal Computer*

Sumber: Everaldo Coelho/Wikipedia.org (2007)



c. *Mini PC*

Merupakan komputer “peralihan” dari komputer personal ke komputer mini yang dipakai di industri. Biasanya, komputer dipakai untuk industri kecil atau personal untuk keperluan profesional atau industri kecil. Contoh bentuk *Mini PC* diberikan pada Gambar 3.4.



Gambar 3.4 Contoh *Mini PC*, Coffee Lake-U-based Bean Canyon Intel NUC8i5BEK2

Sumber: Michael H ("Laserlicht")/Wikimedia Commons

d. *Minicomputer*

Berbeda dengan komputer personal, komputer mini (Gambar 3.5) berukuran lebih besar, dan mempunyai kapasitas memori maupun pemroses yang lebih besar. Komputer mini dipakai menunjang kebutuhan pengolahan informasi perusahaan skala menengah. Saat ini, komputer mini kurang populer dan makin sedikit digunakan karena perusahaan lebih praktis untuk menyewa komputer di *cloud* yang memudahkan pemeliharannya.



Gambar 3.5 Contoh minicomputer PDP-8e minicomputer system.

Sumber: Computer History Museum/Public Domain



e. Komputer *Mainframe*

Komputer *Mainframe* (Gambar 3.6) berukuran lebih besar dibandingkan dengan komputer dan biasanya digunakan oleh perusahaan-perusahaan besar sebagai *server* (peladen).



Gambar 3.6 Salah Satu Contoh Komputer Mainframe IBM z Systems z13

Sumber: Agiorgio/Wikimedia Commons

f. *Supercomputer*

Dibandingkan dengan komputer lainnya, *supercomputer* memiliki ukuran yang paling besar dan memiliki kapasitas pengolahan data dan kinerja yang paling kuat (Gambar 3.7). Superkomputer memiliki kemampuan untuk melakukan triliunan perintah atau instruksi per detik yang dapat dihitung dalam FLOPS (*Floating Point Operation per Second*). Sama seperti *minicomputer* dan *mainframe*, pengguna superkomputer biasanya ialah perusahaan atau organisasi besar misalnya NASA yang menggunakannya dalam meluncurkan dan mengendalikan pesawat dan roket.



Gambar 3.7 Sierra/ATS-2 Supercomputer

Sumber: US Department of Energy/Wikimedia Commons



Catatan: selain “komputer” yang disebutkan di atas, beberapa gawai (*gadget*), yaitu perangkat elektronik kecil yang berfungsi khusus, ada yang termasuk “komputer” karena terdiri atas perangkat keras, sistem operasi, dan perangkat lunak. Contohnya ponsel pintar (*smart phone*) dan tablet, yang makin populer dan menjadi perlengkapan sehari-hari di era digital ini. Selain fungsi utamanya untuk berkomunikasi, ponsel pintar bahkan sudah menjadi “asisten” pribadi.

3. Sistem Operasi

Seperti telah disebutkan sebelumnya salah satu perangkat lunak ialah berupa sistem operasi. Sistem operasi ialah perangkat lunak yang mendasari operasi dasar dari suatu komputer. Sistem operasi berfungsi sebagai perantara antara perangkat keras (*hardware*) komputer dan aplikasi perangkat lunak (*software*) yang dijalankan oleh pengguna. Fungsi utama dari sistem operasi melibatkan manajemen sumber daya dan menyediakan antarmuka untuk berinteraksi dengan perangkat keras. Sistem operasi yang banyak dipakai saat ini ialah Windows, MacOS, dan Linux. Perangkat lunak lain yang menjadi sistem operasi ponsel pintar ialah Android.

Beberapa peran utama dari Sistem Operasi ialah seperti berikut.

- Menyediakan antarmuka bagi pengguna
- Mengendalikan *input* dan *output*
- Mengelola beragam perangkat keras yang terhubung dengan komputer
- Mengelola pemuatan ataupun penjalanan sebuah perangkat lunak
- Mengelola *file* yang ada di komputer (*save, copy, sort, delete*)
- Menangani interupsi dan kesalahan (*error*)
- Mengelola penggunaan prosesor
- Mengelola penggunaan *memory*
- Mengelola keamanan sistem dari akses yang tidak berwenang, misalkan dengan menyediakan sistem *login*
- Menangani komunikasi dengan perangkat jaringan

Kamu tentu pernah melakukan *multitasking*, yaitu mengerjakan beberapa pekerjaan sekaligus, misalnya sambil merajut, nonton TV, bahkan sesekali menengok *handphone*. Sambil menyapu, kamu mendengarkan musik bahkan ikut bernyanyi. Manusia mempunyai kemampuan *multitasking* (Gambar 3.8),



walaupun untuk beberapa kondisi, perlu dilakukan dengan hati-hati, misalnya sangat berbahaya menonton video sambil menyetir mobil. *Multitasking* yang tidak dikendalikan dengan baik, belum tentu menambah efisiensi dan hasilnya belum tentu baik. Misalnya, belajar sambil menonton sepak bola dapat memecah perhatian sehingga kamu tidak belajar dengan baik. Ketika kamu sedang belajar sambil mendengarkan lagu, kamu berhenti ketika ibu memanggil untuk makan malam (ini yang disebut interupsi).



Gambar 3.8 Ilustrasi *Multitasking*

Sebuah komputer yang sedang melakukan *multitasking*, misalnya penggunaanya sedang menjalankan aplikasi pengolah kata, aplikasi pengolah lembar kerja, aplikasi presentasi, dan aplikasi Paint untuk menggambar. Pengguna dapat memindahkan sepotong teks dari satu aplikasi ke lainnya lewat *clipboard*. Diam-diam, jam yang tertulis di pojok layar juga sedang bekerja.

Seperti dijelaskan di atas, salah satu fungsi sistem operasi ialah menangani *multitasking*. Sistem Operasi tidak menangani *multitasking* seperti manusia karena komputer hanya mempunyai satu prosesor, dan prosesor itu yang menjalankan program (lihat aktivitas mesin superkonseptual tentang bagaimana CPU menjalankan program dengan langkah sangat rinci).

Sistem Operasi dapat melakukan *multitasking* dengan menjalankan algoritma *round robin* (RR). Ya, sebuah algoritma karena sistem operasi ialah sebuah program juga. Prinsip dari algoritma penjadwalan *round robin* dijelaskan sebagai berikut.



Round robin (RR) ialah salah satu algoritma yang digunakan oleh penjadwal proses (*process scheduler*) dalam sebuah sistem operasi. Pada algoritma RR, ditentukan suatu slot waktu (*time slice*) yang akan dialokasi ke setiap proses dalam porsi yang sama dan dalam urutan melingkar, menangani semua proses tanpa prioritas. Penjadwalan RR sederhana dan mudah diterapkan. Penjadwalan RR dapat diterapkan pada masalah penjadwalan lainnya, seperti penjadwalan paket data di jaringan komputer. Nama algoritma ini berasal dari prinsip RR, di mana setiap orang mengambil bagian yang sama dari sesuatu secara bergantian.

Agar proses dikerjakan secara adil, penjadwal RR memberikan setiap pekerjaan slot waktu atau penyisihan waktu CPU, dan menginterupsi pekerjaan belum diselesaikan saat itu. Pekerjaan dilanjutkan saat slot waktu berikutnya ditetapkan bagi proses itu. Jika proses selesai atau mengubah statusnya menjadi menunggu selama slot waktu yang diberikan, penjadwal memilih proses pertama dalam antrian siap untuk dieksekusi. Dengan tidak ada banyak pekerjaan yang dilakukan, atau jika slot waktu relatif besar terhadap ukuran pekerjaan, proses yang menghasilkan pekerjaan besar akan lebih banyak dikerjakan daripada proses lainnya.

Misalnya, jika slot waktu ialah 100 milidetik (ms), dan *Job1* membutuhkan total waktu 250 ms untuk menyelesaikannya, penjadwal round robin akan menangguhkan pekerjaan setelah 100 ms dan memberikan waktu pada pekerjaan lain di CPU. Setelah pekerjaan lain memiliki bagian yang sama (masing-masing 100 ms), *Job1* akan mendapatkan alokasi waktu CPU lain dan siklus akan berulang. Proses ini berlanjut hingga pekerjaan selesai dan tidak membutuhkan waktu lagi di CPU.

- *Job1* = membutuhkan 250 ms untuk dapat diselesaikan.
- Alokasi pertama untuk *Job1* = 100 ms.
- Alokasi kedua = 100 ms.
- Alokasi ketiga = 100 ms, tetapi *Job1* selesai dan diakhiri pada 50 ms.
- Maka, waktu CPU untuk *Job1* = 250 ms.



Ada dua pendekatan algoritma untuk menyelesaikan *round robin scheduler*. Pertama (dengan algoritma ini, CPU tidak pernah berhenti)

- *Selama periode satu kuantum*: jika ada *job* selesai, hapus dari antrian, ambil berikutnya.
- *Di akhir satu kuantum*: antrekan kembali, ambil giliran berikutnya.

Kedua

- *Selama periode satu kuantum*: jika ada *job* selesai, hapus dari antrian, tunggu sampai akhir kuantum.
- *Di akhir satu kuantum*: antrekan kembali, ambil giliran berikutnya.

Kuantum=100 ms

Proses	Waktu Kedatangan	Waktu Eksekusi (ms)
P0	0	250
P1	50	170
P2	120	70
P3	170	100
P4	200	130
P5	350	50
	Total	770

Gambar 3.9a berikut menunjukkan waktu kedatangan dan waktu eksekusi dari beberapa proses, dengan slot 100 ms dan eksekusi dari proses-proses tersebut. Selanjutnya, Gambar 3.9b menjelaskan secara detail bagaimana setiap *Job* (P0 sampai dengan P5) dijadwalkan pengerjaannya.

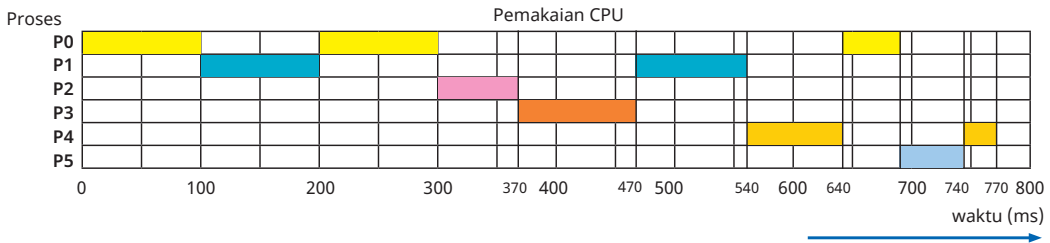
SIMULASI PENJADWALAN PROSES

Waktu	Antrian					
0	P0					
50	P0	P1				
100	P1	P0				
120	P1	P0	P2			
170	P1	P0	P2	P3		
200	P0	P2	P3	P1	P4	
300	P2	P3	P1	P4	P0	
350	P2	P3	P1	P4	P0	P5
370	P3	P1	P4	P0	P5	
470	P1	P4	P0	P5		
540	P4	P0	P5			
640	P0	P5	P4			
690	P5	P4				
740	P4					
770						

Selesai	Deskripsi Kejadian
	P0 datang, diproses.
	P1 datang masuk antrian.
	P1 diproses, P0 diantrikan kembali.
	P2 datang masuk antrian.
	P3 datang masuk antrian.
	P0 diproses, P4 datang masuk antrian.
	P2 diproses, P0 diantrikan kembali.
	P5 datang masuk antrian.
P2	P2 selesai, P3 diproses.
P3	P3 selesai, P1 diproses.
P1	P1 selesai, P4 diproses.
	P0 diproses, P4 diantrikan kembali.
P0	P0 selesai, P5 diproses.
P5	P5 selesai, P4 diproses.
P4	P4 selesai.

(a)





(b)

Gambar 3.9 (a) Simulasi Penjadwalan Proses, (b) Simulasi CPU

Sumber: Dokumen Kemdikbudristek (2021)

Interpretasi diagram!

Jawablah pertanyaan berikut ini.

1. Mengacu ke simulasi eksekusi tersebut, pekerjaan (proses) mana yang selesai terlebih dahulu dan mana yang selesai paling akhir?
2. Kesimpulan apa saja yang kamu dapatkan?
3. Dari dua algoritma yang dijelaskan di atas, algoritma mana yang dipilih?

Kamu juga dapat membuat diagram eksekusi yang lebih visual dengan *sticky notes*, untuk mempermudah interpretasinya.

Aktivitas 10.3-01-U Simulasi Multitasking

Pada aktivitas ini, kamu berlatih untuk menerapkan sistem *multitasking* dengan menggunakan simulasi dan permainan peran (*role play*).

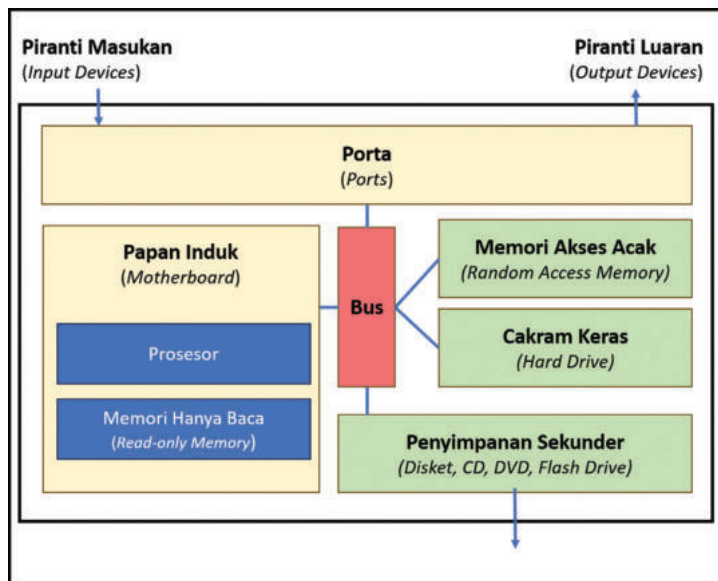
Langkah-Langkah

1. Setiap kelompok akan berbagi.
 - a) Berperan sebagai “*scheduler*” (penjadwal) yang merupakan bagian dari SO. *Scheduler* mengelola permintaan layanan dan antrian, mencatatnya dalam lembar catatannya. Setiap *job* yang datang akan memberi tahu sebelumnya berapa lama waktu eksekusi yang dibutuhkannya.
 - b) CPU (yang menjalankan *job*, menahannya selama mendapat giliran) dan melepaskannya saat kuantum waktu habis.

- c) Bus yang akan membawa dan mengembalikan “*job*” ke antrean yang dikelola *scheduler* berikut statusnya (selesai, atau masih perlu berapa lama lagi).
2. Guru akan memberikan sebuah tabel yang isinya waktu kedatangan beberapa *job* untuk dilayani oleh sistem operasi.
3. Simulasikan pelaksanaan pekerjaan mulai kedatangan *job* yang pertama, sampai semua *job* selesai dikerjakan.

4. Bagaimana Komputer Bekerja

Komputer memiliki mekanisme untuk bekerja karena komputer memiliki banyak komponen yang harus melakukan interaksi satu sama lain. Secara ringkas, Gambar 3.10 menggambarkan diagram kotak arsitektur sederhana sebuah komputer di mana pemroses (kotak besar) menerima masukan dari perangkat masukan dan menghasilkan keluaran melalui *port*. Komponen pada pemroses (kotak besar) terdiri atas prosesor (CPU) yang berupa chip, ROM, RAM, Hard drive, CD ROM, dan *floppy drive* (sekarang tidak populer). CD ROM dan *floppy drive* ialah alat yang terhubung dengan media penyimpan eksternal, yaitu (CD dan *floppy disk*). Setiap komponen tersebut terhubung dengan bus.



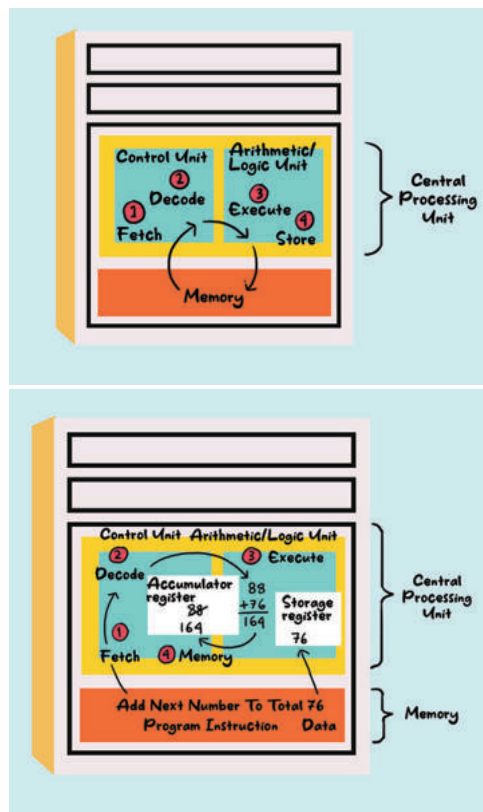
Gambar 3.10 Diagram Arsitektur Sederhana Komputer

Sumber: Dokumen Kemdikbudristek (2021)



Bagaimana CPU menjalankan instruksi program?

Mari, periksalah cara unit pengolah pusat (CPU), dalam hubungannya dengan memori, menjalankan program komputer. Kamu akan melihat bagaimana hanya satu instruksi dalam program yang dijalankan. Faktanya, kebanyakan komputer saat ini hanya dapat menjalankan satu instruksi pada satu waktu, meskipun mereka menjalaninya dengan sangat cepat. Banyak komputer pribadi dapat menjalankan instruksi dalam waktu kurang dari sepersejuta detik. Komputer yang dikenal sebagai superkomputer dapat menjalankan instruksi dalam waktu kurang dari sepermiliar detik.



Gambar 3.11 Siklus mesin pada komputer sebelum dan setelah bekerja.

Sumber: Dokumen Kemdikbudristek (2021)

Sebelum instruksi dapat dieksekusi, instruksi program dan data harus ditempatkan ke dalam memori dari perangkat input atau perangkat penyimpanan sekunder. Seperti yang ditunjukkan Gambar 3.11, setelah data dan instruksi yang diperlukan berada dalam memori, unit pemrosesan pusat melakukan empat langkah berikut untuk setiap instruksi.

1. Unit kontrol mengambil (mendapat) instruksi dari memori.
2. Unit kontrol menerjemahkan instruksi (memutuskan apa artinya) dan memerintahkan agar data yang diperlukan dipindahkan dari memori ke ALU (unit aritmatika/logika). Dua langkah pertama ini bersama-sama disebut waktu instruksi, atau waktu-I.
3. Unit aritmatika/logika menjalankan instruksi aritmatika atau logika. Artinya, ALU diberikan kendali dan melakukan operasi aktual pada data.
4. Unit aritmatika/logika menyimpan hasil operasi ini dalam memori atau *register*. Langkah 3 dan 4 bersama-sama disebut waktu eksekusi, atau waktu-E.

Unit kontrol selanjutnya memerintahkan memori untuk mengirimkan hasilnya ke perangkat keluaran atau perangkat penyimpanan sekunder. Kombinasi waktu-I dan waktu-E disebut siklus mesin. Gambar 3.11 (bawah) menunjukkan instruksi yang melalui siklus mesin.

CPU memiliki *clock* internal yang menghasilkan detak (*pulse*) dengan kecepatan tetap untuk menyinkronkan semua operasi komputer. Sebuah instruksi siklus mesin tunggal dapat terdiri atas sejumlah besar sub-instruksi, yang masing-masing harus mengambil setidaknya satu siklus *clock*. Setiap jenis CPU dirancang untuk memahami sekumpulan instruksi tertentu yang disebut *instruction set*. Sama seperti ada banyak bahasa berbeda yang dipahami orang, setiap jenis CPU memiliki *instruction set* yang dimengertinya. Oleh karena itu, belum tentu CPU yang dapat digunakan pada satu komputer dapat digunakan komputer yang lainnya.

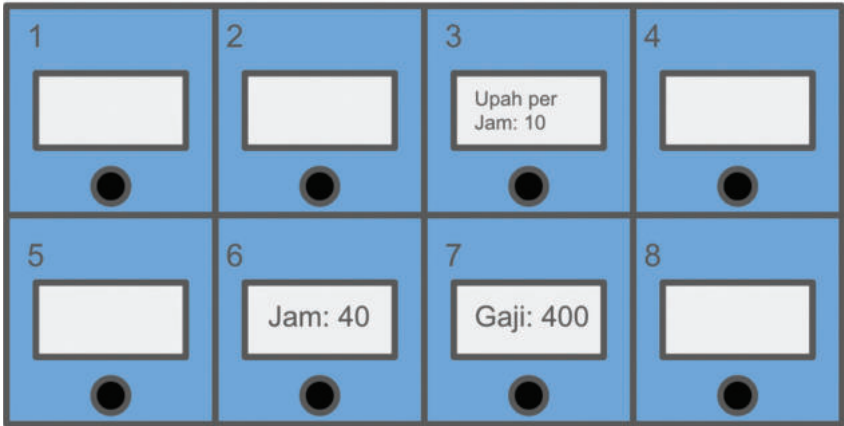
Pada memori, dapat tersimpan instruksi program dan juga data. Bagaimana *control unit* dapat membedakannya?

Lokasi dalam memori untuk setiap instruksi dan setiap bagian data diidentifikasi oleh sebuah alamat. Artinya, setiap lokasi memiliki nomor alamat, seperti *locker* (kotak penyimpanan) di perpustakaan atau di sekolah. Seperti kotak penyimpanan memiliki nomor yang tetap, tetapi isi kotak penyimpanan dapat berbeda di suatu waktu: dapat berisi tas, berisi buku, atau berisi *tumbler* tempat minum.

Seperti hal kotak penyimpanan, memori dapat berisi instruksi atau data. Instruksi lama dapat diganti dengan instruksi baru, data lama dapat diganti dengan data baru, tetapi memori tetap memiliki alamat yang sama. Tidak seperti



kotak penyimpanan, lokasi memori hanya dapat menampung sejumlah data dalam ukuran *byte*.



Gambar 3.12 Memory Address seperti Kotak Surat

Sumber: Dokumen Kemdikbudristek (2021)

Gambar 3.12 menunjukkan bagaimana program memanipulasi data dalam memori. Sebuah program penggajian, misalnya, dapat memberikan instruksi untuk meletakkan data tarif (gaji-per-jam) di lokasi kotak 3 dan jumlah jam kerja di lokasi kotak 6. Untuk menghitung gaji karyawan, instruksi untuk komputer, yaitu mengalikan data di lokasi kotak 3 dengan data di lokasi kotak 6 dan pindahkan hasilnya ke lokasi kotak 8. Pemilihan lokasi dapat dilakukan di mana saja yang belum digunakan. Pemrogram yang menggunakan bahasa pemrograman, tidak perlu tahu nomor alamat mesin yang sebenarnya karena setiap alamat data disebut dengan nama yang menjadi alamat simbolis. Dalam contoh ini, nama alamat simbolis adalah Tarif, Jam, dan Gaji.

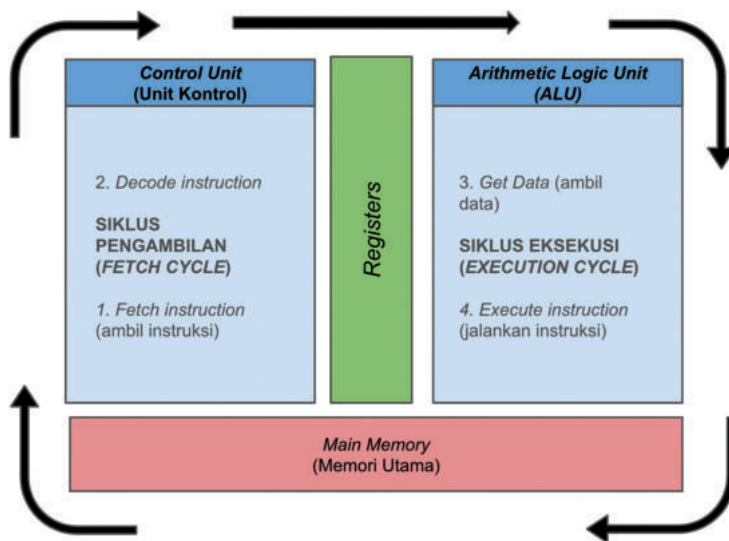
5. Mesin Konseptual Sederhana

Tahukah kamu, bahwa kamu dapat membuat abstraksi sebuah mesin komputer dengan menciptakan sebuah mesin konseptual sederhana. Kamu diberi sebuah mesin konseptual sederhana ciptaan Mr. ALGO dan menyimulasikan cara kerjanya, yang seperti cara kerja sebuah komputer, tetapi dengan lebih sederhana. Pada bagian ini, kamu akan memahami bahwa instruksi program dalam bahasa yang lebih dekat ke manusia harus diterjemahkan menjadi instruksi dalam bahasa mesin untuk dapat dijalankan. Ingat bahwa dengan struktur komputer yang terdiri atas input, *output*, memori, dan CPU, komputer hanya dapat *membaca* data dari perangkat masukan, *menulis* data ke perangkat keluaran,



menyalin data dari CPU ke memori atau dari memori ke CPU, dan *melakukan perhitungan* aritmatika dan logika. Dengan kemampuan tersebut, kamu dapat menuliskan program komputer yang beragam dan luar biasa daya gunanya!

Program komputer terdiri atas sekumpulan instruksi, dan instruksi yang dijalankan oleh mesin harus dikenali oleh CPU yang disebut bahasa mesin. Bagaimana sebuah komputer menjalankan sebuah program dalam bahasa mesin? Komputer akan menjalankan (mengeksekusi) perhitungan dengan langkah yang disebut *fetch execute cycle* (siklus ambil dan jalankan). *Fetch execute cycle* ialah operasi yang paling mendasar dalam komputer, yang juga disebut *fetch decode execute cycle*. Selama *fetch execute cycle*, mesin komputer akan mengambil instruksi dari memori dan menjalankan instruksi tersebut sesuai dengan jenis instruksinya. Mengapa disebut *fetch execute cycle*? Siklus untuk *fetching*, *decoding*, dan *executing* akan diulang-ulang oleh CPU selama mesin komputer hidup! Gambaran siklusnya ditunjukkan dalam Gambar 3.13. Gambar ini menunjukkan lebih detail bagian-bagian dari CPU.



1. CU akan mengambil instruksi (*fetch execute cycle*).
2. CU akan menerjemahkan instruksi tersebut harus melakukan apa, misalnya menyimpan data, menghitung, atau lainnya. Ingat bahwa komputer hanya dapat menghitung dan menyimpan/mengambil, atau mengirimkan data ke *input/output device*.
3. ALU akan mengambil data yang diperlukan untuk menjalankan instruksi, dan data yang sedang diproses disimpan dalam *register*.
4. ALU menjalankan instruksi.

Gambar 3.13 Siklus Ambil dan Jalankan (*Fetch Execute Cycle*)

Sumber: Dokumen Kemdikbudristek (2021)



Berikut ini gambaran Memori dan Register dari sebuah mesin ciptaan Mr. ALGO tersebut. Alamat pada mesin ini dinyatakan dalam kode Heksadesimal. Memori dibagi-bagi menjadi kotak-kotak, setiap kotak mempunyai alamat dan dapat berisi data. Misalnya, mesin ciptaan kamu memiliki kapasitas memori untuk menampung 4 data dengan alamat AAA1 s.d. AAA4 dan CPU mempunyai 2 register dengan alamat REG1 dan REG2 yang ilustrasikan seperti pada gambar di bawah ini. Sebetulnya, semua data akan disimpan dalam bentuk biner, seperti yang telah dipelajari di jenjang SMP. Namun demikian, untuk kemudahan membaca data semua ilustrasi, data tetap dituliskan dalam besaran desimal.

MEMORI				REGISTER	
AAA1	AAA2	AAA3	AAA4	REG1	REG2

Sekarang, kamu akan meniru siklus pengambilan instruksi (*fetch instruction cycle*) untuk menjalankan perhitungan aritmatika sederhana. Instruksi ini awalnya ditulis dalam bahasa yang lebih mudah dimengerti oleh manusia (bahasa tingkat tinggi).

```
X=200
Y=100
Jumlah= X
+ Y PRINT
Jumlah
```

Mesin Konseptual Sederhana akan mengeksekusi perintah tersebut dalam beberapa langkah. Hal itu karena data disimpan dalam memori (disimpan dalam variabel X dan Y), sedangkan proses perhitungan penjumlahan harus dilakukan oleh ALU yang merupakan bagian dari CPU. Misalnya, nilai dari variabel X disimpan dalam alamat AAA1 dan nilai dari variabel Y disimpan dalam AAA2. Nilai variabel Jumlah akan disimpan dalam alamat AAA4. Data harus dibawa ke register untuk dijumlahkan.

Mesin Konseptual Sederhana menjalankan beberapa instruksi bahasa mesin sebagai berikut untuk menjalankan program di atas.

```
SIMPAN 100 AAA1
SIMPAN 200 AAA2
SALIN AAA1 REG1
SALIN AAA2 REG2
TAMBAH REG1 REG2
SALIN REG2 AAA3
PRINT AAA3
```



Eksekusi dari perintah-perintah tersebut secara berturut-turut ditunjukkan dengan ilustrasi sebagai berikut.

Tabel 3.1 Contoh Implementasi Mesin Konseptual Sederhana

Instruksi	Penjelasan	Peta Memori dan CPU
SIMPAN 100 AAA1	Menyimpan data 100 ke dalam memori dengan alamat AAA1.	100 AAA1 AAA2 AAA3 AAA4 REG1 REG2
SIMPAN 200 AAA2	Menyimpan data 200 ke dalam memori dengan alamat AAA2.	100 AAA1 200 AAA2 AAA3 AAA4 REG1 REG2
SALIN AAA1 REG1	Menyalin data dari memori dengan alamat AAA1 ke register dengan alamat REG1.	100 AAA1 200 AAA2 AAA3 AAA4 100 REG1 REG2
SALIN AAA2 REG2	Menyalin data dari memori dengan alamat AAA2 ke register dengan alamat REG2.	100 AAA1 200 AAA2 AAA3 AAA4 100 REG1 200 REG2
TAMBAH REG1 REG2	Menambahkan data pada alamat REG1 dan data pada alamat REG2 dan mesin akan menyimpan hasilnya pada alamat REG2. *) Baca Penjelasan di bawah tabel.	100 AAA1 200 AAA2 AAA3 AAA4 100 REG1 300 REG2
SALIN REG2 AAA3	Menyalin data dari register REG2 ke memori AAA3.	100 AAA1 200 AAA2 300 AAA3 AAA4 100 REG1 300 REG2
PRINT AAA3	Mencetak data yang ada pada memori AAA3 ke layar monitor mesin untuk ditampilkan.	300

Mesin Konseptual Sederhana tersebut hanya mempunyai 2 register (REG1 dan REG2), dan dirancang agar jika melakukan operasi aritmatika dua buah bilangan, hasilnya disimpan pada register yang menyimpan operan ke-2.

Mungkin saja, ada orang yang menciptakan mesin konseptual lain yang mempunyai 3 register. Seandainya mesin mempunyai 3 register, dapat saja



hasil penjumlahan disimpan pada register ke-3. Jumlah register, dan bagaimana operasi dilakukan, serta hasil operasi aritmatika disimpan, itu ditentukan oleh penciptanya, merupakan spesifikasi mesin.

Kamu juga dapat menciptakan mesin konseptual yang sangat sederhana, yang hanya mempunyai 1 register saja untuk menampung data. Mesin ini biasanya disebut AKUMULATOR karena semua operan dan hasil perhitungan disimpan di register tunggal. Perintah TAMBAH dilakukan dengan menambahkan data yang diambil dari memori ke satu-satunya register itu. Perintah mesin misalnya seperti berikut.

Tabel 3.2 Contoh Implementasi Mesin Konseptual 1 Register

Instruksi	Hasil
LOAD <alamat>	Data yang disimpan pada alamat memori <i>disimpan</i> ke AKUMULATOR.
TAMBAH <alamat>	Data yang disimpan pada AKUMULATOR <i>ditambah</i> dengan data yang diambil dari alamat memori.
KURANG <alamat>	Data yang disimpan pada AKUMULATOR <i>dikurangi</i> dengan data yang diambil dari alamat memori.
KALI <alamat>	Data yang disimpan pada pada AKUMULATOR <i>dikalikan</i> dengan data yang diambil dari alamat memori.
BAGI <alamat>	Data yang disimpan pada AKUMULATOR <i>dibagi</i> dengan data yang diambil dari alamat memori.
OUTPUT	Data AKUMULATOR <i>dikirim</i> ke perangkat keluaran.
INPUT <data><alamat>	Alamat diisi dengan data.

Untuk menambahkan 100+200, instruksi yang dilakukan oleh mesin dengan satu AKUMULATOR yang hanya mempunyai 4 alamat memori AAA1, AAA2, AAA3, dan AAA4 diberikan sebagai berikut.



Tabel 3.3 Memori dalam Mesin dangn! Akumulator

Instruksi	Isi Memori dan CPU (Akumulator)												
INPUT 100 AAA1	<table><tr><td>100</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>AAA1</td><td>AAA2</td><td>AAA3</td><td>AAA4</td><td></td><td>AKUMULATOR</td></tr></table>	100						AAA1	AAA2	AAA3	AAA4		AKUMULATOR
100													
AAA1	AAA2	AAA3	AAA4		AKUMULATOR								
INPUT 200 AAA2	<table><tr><td>100</td><td>200</td><td></td><td></td><td></td><td></td></tr><tr><td>AAA1</td><td>AAA2</td><td>AAA3</td><td>AAA4</td><td></td><td>AKUMULATOR</td></tr></table>	100	200					AAA1	AAA2	AAA3	AAA4		AKUMULATOR
100	200												
AAA1	AAA2	AAA3	AAA4		AKUMULATOR								
LOAD AAA1	<table><tr><td>100</td><td>200</td><td></td><td></td><td></td><td>100</td></tr><tr><td>AAA1</td><td>AAA2</td><td>AAA3</td><td>AAA4</td><td></td><td>AKUMULATOR</td></tr></table>	100	200				100	AAA1	AAA2	AAA3	AAA4		AKUMULATOR
100	200				100								
AAA1	AAA2	AAA3	AAA4		AKUMULATOR								
TAMBAH AAA2	<table><tr><td>100</td><td>200</td><td></td><td></td><td></td><td>300</td></tr><tr><td>AAA1</td><td>AAA2</td><td>AAA3</td><td>AAA4</td><td></td><td>AKUMULATOR</td></tr></table>	100	200				300	AAA1	AAA2	AAA3	AAA4		AKUMULATOR
100	200				300								
AAA1	AAA2	AAA3	AAA4		AKUMULATOR								
OUTPUT	300												

Jika kamu perhatikan, instruksi-instruksi di atas merupakan rangkaian perintah yang dilakukan untuk menghitung penjumlahan 100+200. Instruksi yang diberikan bergantung pada spesifikasi mesinnya, dan mesin juga dapat mempunyai satu atau lebih register. Pencipta mesin yang akan merancang mesin-mesin komputer tersebut dan mewujudkannya menjadi perangkat elektronik yang dapat berfungsi sesuai spesifikasi. Tentu saja, gambaran pelaksanaan instruksi di atas ialah penyederhanaan karena yang dilakukan oleh komputer sebenarnya jauh lebih rinci. Data 100 dan 200 dalam desimal pun akan direpresentasi dalam bentuk biner seperti yang kamu telah pelajari di tingkat SMP.

Unit pembelajaran ini memberikan gambaran bahwa ekspresi yang ditulis dalam bahasa yang dimengerti manusia harus diterjemahkan sangat detail agar dapat dieksekusi oleh mesin ciptaan kamu. Komputer yang selama ini kamu gunakan sebenarnya mirip dengan mesin konseptual sederhana tersebut, hanya saja kecepatan melaksanakan instruksi sangat cepat sekali.





Aktivitas 10.3-02-U

Simulasi Eksekusi Perintah dalam Mesin Ciptaan Mr. ALGO

Pada aktivitas ini, kamu akan bermain untuk memahami lebih detail lagi tentang bagaimana data disimpan dan diproses dalam sebuah komputer konseptual

Langkah-Langkah

Bentuk kelompok yang terdiri atas 3-4 siswa.

Mr. ALGO menciptakan mesin konseptual sederhana dengan 5 alamat memori AEB1, AEB2, AEB3, AEB4, AEB5, dan CPU-nya memiliki 3 register REG1, REG2, dan REG3. Spesifikasi instruksi-instruksinya sebagai berikut.

Tabel 3.4 Contoh Instruksi 5 Alamat Memori 3 Register

Instruksi Mesin	Penjelasan
SIMPAN <data><alamat>	Menyimpan data ke alamat memori yang dituju.
SALIN <alamat1><alamat2>	Menyalin data dari alamat1 untuk kemudian disimpan pada alamat2. Alamat dapat berupa alamat memori atau register.
TAMBAH REG1 REG2	Melakukan penjumlahan data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3.
KALI REG1 REG2	Melakukan perkalian data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3.
KURANG REG1 REG2	Melakukan pengurangan data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3.



Instruksi Mesin	Penjelasan
<i>BAGI REG1 REG2</i>	Melakukan pembagian data pada REG1 dan REG2, kemudian hasilnya disimpan di REG3.
<i>PRINT <alamat></i>	Mencetak data yang ada di alamat memori untuk ditampilkan di monitor.

Tugas kamu ialah menerjemahkan operasi matematika berikut ke dalam instruksi-instruksi supaya dapat dieksekusi oleh mesin ciptaan Mr. ALGO tersebut.

- a) $3 + 8 \times 9$
- b) $(1 + 2) \times (8 - 5)$
- c) $2 \times 10 - 8 + 3$
- d) $3 + 16/2$

Dengan menggunakan *sticky note*, jangan lupa untuk menggambarkan peta dari memori dan CPU di setiap instruksi yang dieksekusi oleh mesin kamu ke dalam kertas berukuran A0.

Ayo, Renungkan!

Suatu hari, jika akan menjadi ahli perancang komputer, kamu akan berkenalan dengan *chips*, atau komponen elektronik yang membentuk mesin ciptaan. Tentu, kamu harus membuat bahasa yang dipahami oleh komponen-komponen itu dengan memanfaatkan *interface* yang tersedia. Menantang, bukan?



B. Otomatisasi dengan Perangkat Lunak Perkantoran

Dalam bagian ini, kamu akan mempelajari tentang aplikasi perkantoran, integrasi pada perangkat lunak perkantoran, dan penerapan OLE pada integrasi perangkat lunak perkantoran.

1. Aplikasi Perkantoran

Perangkat lunak perkantoran atau aplikasi *office* ialah kumpulan aplikasi yang digunakan untuk memudahkan pekerjaan perkantoran. Karena berupa aplikasi, fitur-fiturnya sudah tersedia dan dapat langsung digunakan oleh pengguna, tanpa harus membangunnya sendiri menggunakan bahasa pemrograman.

Secara umum, aplikasi *office* terbagi menjadi beberapa jenis menurut kegunaan dan fungsinya. Tiga di antaranya yang paling banyak digunakan ialah seperti berikut.

a. Perangkat Lunak Pengolah Kata (*Word Processor*)

Aplikasi ini digunakan untuk membuat dan mengolah dokumen, misalnya membuat surat, menyusun laporan, dan lain-lain. Contoh aplikasi pengolah kata ialah Microsoft Word, OpenOffice Word, Google Docs, dan lain-lain.

b. Perangkat Lunak Pengolah Data/Angka (*Spreadsheet*)

Aplikasi ini digunakan untuk mengelola data yang disajikan dalam bentuk tabel dua dimensi yang terdiri atas kolom dan baris. Fitur-fitur yang tersedia dapat mempermudah perhitungan data dan mengolah data secara statistik, sekaligus menampilkannya dalam bentuk diagram. Contoh aplikasi *spreadsheet* ialah Microsoft Excel, OpenOffice Spreadsheet, Google Sheet, dan lain-lain.

c. Perangkat Lunak Pembuat Presentasi

Aplikasi ini digunakan untuk membuat *slide* presentasi dengan mudah, cepat, dan menarik. Contoh perangkat lunak pembuat presentasi ialah Microsoft Office Powerpoint, OpenOffice Presentation, Google Slide, dan lain-lain.

2. Integrasi pada Perangkat Lunak Perkantoran

Mengintegrasikan artinya menyatukan beberapa objek, data atau komponen untuk membentuk sesuatu yang utuh dan bermakna, walaupun komponennya tadinya terpecah atau tidak ada hubungannya. Misalnya, kamu harus membuat sebuah laporan berdasarkan pengamatan yang datanya dibuat dan divisualisasi



dengan excel, dan sebelum laporan lengkap, kamu membuat proposal dalam bentuk *slides*. Saat kamu membuat laporan, untuk menghindari mengerjakan hal yang sama (mengetik ulang), kita perlu “membawa” potongan hasil kerja dengan MS Excel, MS Word dan MS PowerPoint ke dalam MS Word karena laporan akhir dibuat dengan MS Word.

Setiap jenis aplikasi perkantoran memiliki fungsi yang berbeda sehingga pengguna biasanya hanya menggunakan satu aplikasi sekali waktu, misalnya menggunakan aplikasi *word processor* untuk menyusun dokumen. Akan tetapi, ada kalanya, pengguna memerlukan lebih dari satu aplikasi untuk suatu keperluan. Sebagai contoh, saat diperlukan laporan yang memuat diagram, selain *word processor*, diperlukan juga *spreadsheet* untuk membuat diagram. Oleh karena itu, diperlukan integrasi antaraplikasi perkantoran.

Integrasi memungkinkan untuk mengaitkan data maupun fitur antaraplikasi sehingga data atau fitur pada satu aplikasi dapat digunakan di aplikasi yang lain. Integrasi aplikasi perkantoran bertujuan untuk menggabungkan aplikasi lain dalam satu aplikasi. Sebagai contoh, pada Microsoft Office, data dari Microsoft Excel dapat ditautkan dengan dokumen pada Microsoft Word dalam pembuatan surat, atau data di Microsoft Excel ditautkan dengan presentasi pada PowerPoint dalam menampilkan grafik dan tabel. Kedua contoh tersebut merupakan contoh implementasi dari integrasi antaraplikasi di Microsoft Office. Microsoft Excel digunakan sebagai sumber data atau objek untuk disajikan dalam dokumen atau presentasi.

Terdapat beberapa cara untuk mengintegrasikan data, teks, gambar antaraplikasi perkantoran, yaitu menggunakan: (a) teknik “Salin-Tempel” (*Copy Paste*) atau “Gunting-Tempel” (*Cut Paste*), (b) Menu yang tersedia di aplikasi, (c) teknik *Object Linking* dan *Embedding*.

a. Integrasi dengan Perintah *Cut*, *Copy*, and *Paste*

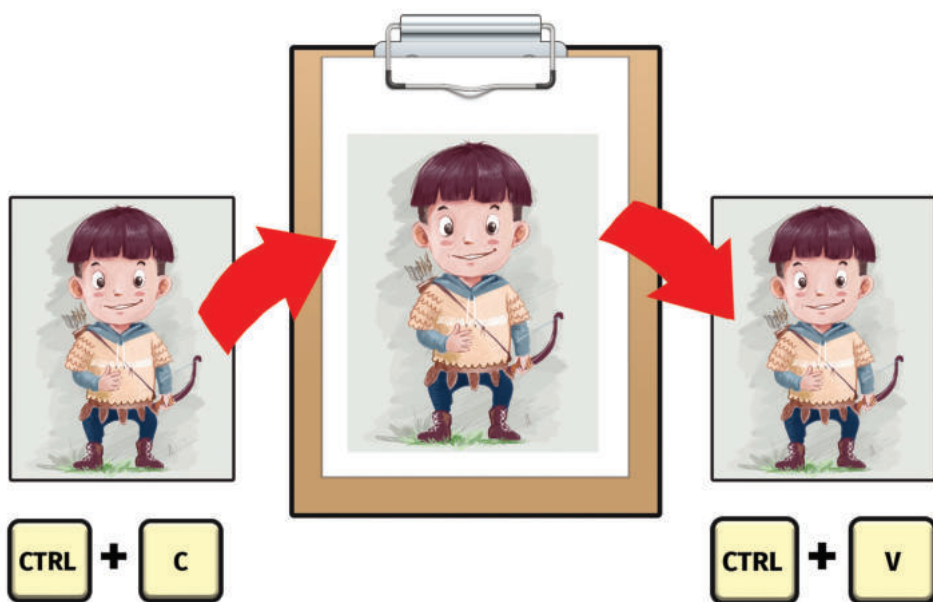
Perintah *Cut* (Potong), *Copy* (Salin), dan *Paste* (Tempel) pada aplikasi komputer saat ini diinspirasi dari praktik tradisional dalam pengeditan naskah yang diketik pada sebuah kertas, di mana orang akan memotong (*cut*) paragraf dari halaman dengan gunting dan menempelkannya ke halaman lain (*paste*). Praktik ini tetap berlangsung hingga 1980-an. Pada saat itu, toko alat tulis bahkan menjual “gunting pengeditan” dengan bilah cukup panjang yang mampu memotong halaman selebar 22 cm.



Saat ini, perintah potong dan tempel (*cut, copy, and paste*) sangat populer digunakan. Banyak aplikasi menyediakan cara unik untuk metode ini seperti: kombinasi tombol, menu tarik-turun (*pull-down menu*), men *pop-up*, dll.

Kalau mekanisme tradisional Potong dan Tempel menggunakan gunting, pada aplikasi komputer, perintah *Cut* memindahkan teks/grafik atau objek lain ke dalam *clipboard* atau *buffer* berupa tempat penyimpanan sementara. Perintah *Paste* akan memindahkan objek dari *clipboard* tersebut menuju ke dokumen tujuan. Perintah *Copy* akan menyalin teks/grafik atau objek lain yang disorot ke dalam *clipboard* dan akan memindahkan objek dari *clipboard* tersebut menuju ke dokumen tujuan.

Perintah “Potong dan Tempel” memiliki urutan cara yang sama, tetapi perintah untuk Potong, yaitu menggunakan Ctrl-x (⌘ + x untuk pengguna Macintosh). Contohnya ialah melakukan *Copy Paste* pada tabel dari aplikasi lembar kerja ke pengolah kata. Atau, melakukan penyalinan dari MS Word dan ditempel ke MS Excel atau MS PowerPoint. Artinya, kamu dapat membuka dua atau tiga aplikasi sekaligus, dan membawa potongan teks, tabel, gambar dari satu aplikasi ke aplikasi lainnya.

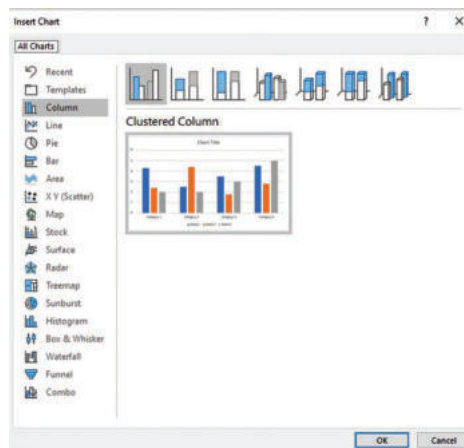


Gambar 3.14 Ilustrasi mengenai perintah potong dan tempel dengan analogi yang berasal dari dunia nyata.



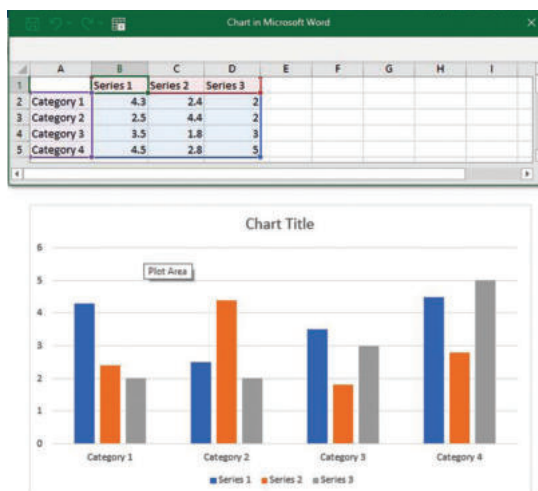
b. Fitur Integrasi pada Aplikasi

Integrasi konten selanjutnya dilakukan dengan menggunakan fitur integrasi yang tersedia di aplikasi. Kamu dapat menggunakan menu pada salah satu aplikasi yang secara otomatis membuka aplikasi lainnya. Misalnya, pada Microsoft Word, ketika dipilih menu *Insert Chart*, Microsoft Excel secara otomatis akan terbuka. Tentu, kamu juga membawa *chart* ke MS PowerPoint dan melakukan hal yang sama. Gambar 3.14 dan Gambar 3.15 menunjukkan contoh menu *Insert Chart* yang dapat menampilkan *sheet* yang dapat diisi data untuk membuat *chart*.



Gambar 3.15 *Insert Chart* pada MS Word

Sumber: Tangkapan layar Microsoft Word



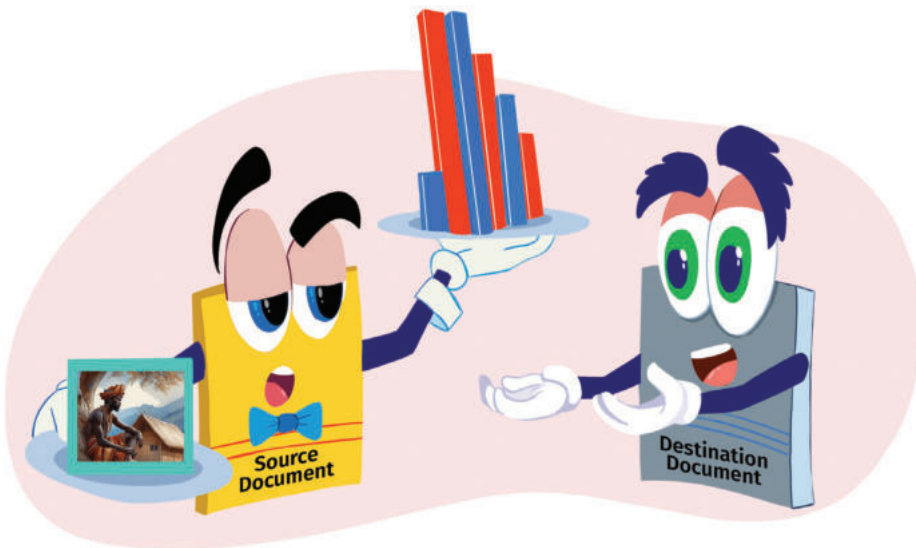
Gambar 3.16 Muncul *Sheet* seperti pada Microsoft Excel

Sumber: Tangkapan layar Microsoft Excel



c. *Object Linking and Embedding*

Menggunakan fasilitas *Object Linking* (untuk Microsoft Office, Gambar 3.17). Pada cara ini, objek yang berasal dari aplikasi perkantoran lain dapat disisipkan ke aplikasi perkantoran yang sedang dibuat. Perubahan pada data sumber akan mengakibatkan perubahan pada objek yang dihubungkan dengan fasilitas ini pada aplikasi lainnya.



Gambar 3.17 Ilustrasi Object Linking Antardua Dokumen

Sumber: Dokumen Kemdikburistek (2021)

Pertanyaan Pemahaman

Jawablah pertanyaan di bawah ini dalam buku catatan, dan jangan lupa mencatat kegiatan dalam Jurnal.

1. Tahukah kamu, mengapa potongan teks, tabel atau gambar dapat di-copy/paste dan dibawa antaraplikasi?
2. Dalam *Object Linking*, jika sumber diedit, otomatis semua objek yang terhubung akan berubah. Menurut kamu, mengapa ini terjadi? Mana yang lebih menguntungkan, melakukan copy/paste atau *Object Linking*?
3. Kelak, konsep ini akan kamu pelajari dalam pemrograman! Kedua konsep ini, yaitu (1) menyalin dan membawa salinannya, atau (2) hanya mengacu tanpa membawa objek yang disalin merupakan dua konsep yang penting dalam informatika. Konsep yang kedua akan menjadi dasar kamu belajar “pointer”.

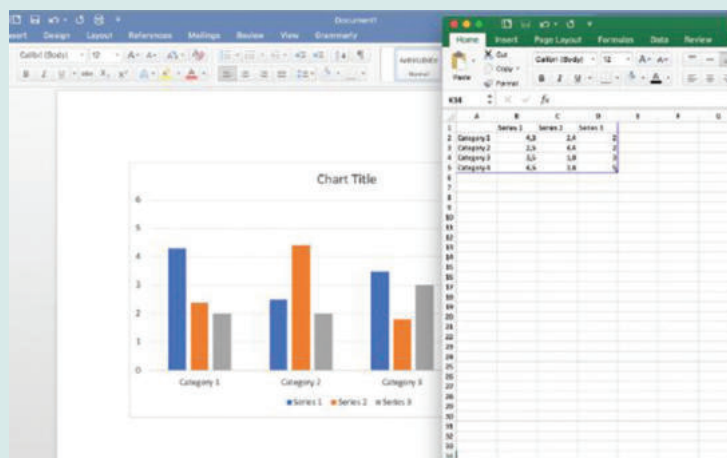


Aktivitas 10.3-03-P **Integrasi Word dan Excel**

Sering kali, kamu perlu untuk membuat laporan yang mengandung gambar dan diagram (*chart*). MS Word menyediakan editor gambar tetap terbatas dan tidak direlasikan dengan data. Jika *chart* berasal dari data yang diolah dan dibuat dengan MS Excel, kamu perlu “memindahkan” *chart* yang berasal dari data dan dibuat di MS Excel, menjadi potongan gambar pada MS Word. Kalau data berubah, kamu harus mengulangi proses yang sama.

Langkah-Langkah

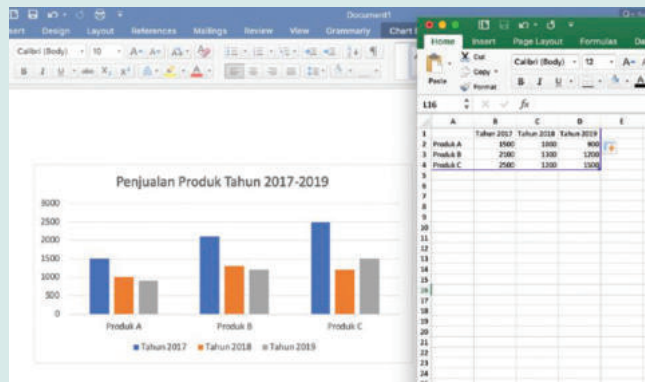
1. Buka aplikasi MS Word (tampilan menggunakan MS Office 2016).
2. Klik menu Insert Chart.
3. Pilih jenis diagram yang diinginkan. Maka, akan muncul tampilan diagram secara otomatis pada Word dan Excel yang memuat datanya. Berikut diagram word dan *dataset* dalam bentuk Excel.



Sumber: Tangkapan layar Microsoft Word dan Microsoft Excel

4. Lakukan perubahan pada data sesuai dengan yang diperlukan. Banyaknya baris maupun kolom dapat disesuaikan pula. Kemudian, tutup MS Excel.
5. Ubahlah judul diagram dengan melakukan *double click* pada Chart Title. Tulis judul sesuai dengan yang diinginkan. Berikut tampilan setelah diagram diberi judul.





Sumber: Tangkapan layar Microsoft Word dan Excel

Pertanyaan Pemahaman

Jawablah pertanyaan di bawah ini dalam buku catatan, dan jangan lupa mencatat kegiatan dalam Jurnal!

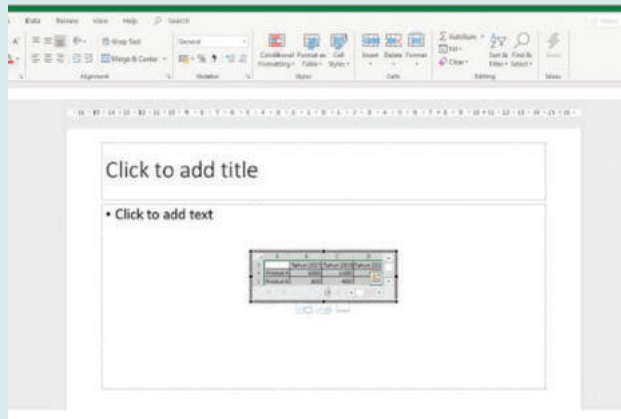
1. Apakah kamu menyadari, bahwa dengan melakukan hal di atas, kamu mengaktifkan (“memanggil”) MS Excel saat sedang memakai MS Word? Menurutmu, apa yang terjadi? Mengapa demikian?
2. Apa bedanya dengan misalnya, mengerjakan tabel dan gambar dengan menggunakan MS Excel, kemudian hasilnya dibawa dalam bentuk gambar (misalnya dengan “sniper” atau mengambil potongan layar menjadi gambar) sehingga tersimpan di *Clipboard*? Setelah itu, kamu membuka MS Word dan melakukan paste dalam dokumen MS Word.

Aktivitas 10.3-04-P Integrasi PowerPoint dan Excel

Dalam aktivitas ini, kamu akan membuat tabel excel sebagai bagian dari objek dalam sebuah *slide* PowerPoint.

Langkah-Langkah

1. Buka aplikasi PowerPoint.
2. Klik menu *Insert Table Excel Spreadsheet*. Perintah ini akan membuka *Worksheet* secara otomatis. Berikut *table spreadsheet* pada PowerPoint.



Sumber: Tangkapan layar Microsoft Excel

3. Buatlah tabel secukupnya. Meskipun yang terbuka hanya jendela kecil, tetapi fitur-fitur Excel, termasuk fungsi-fungsinya, tersedia secara lengkap seperti yang biasa terdapat pada aplikasi Excel. Peserta dapat mencoba fungsi-fungsi matematika yang biasa digunakan.
4. Apabila ingin dilakukan *update* pada data, lakukan *double click* pada tabel, akan muncul kembali *worksheet*. Kemudian, lakukan perubahan sesuai yang diinginkan.



Aktivitas 10.3-05-P

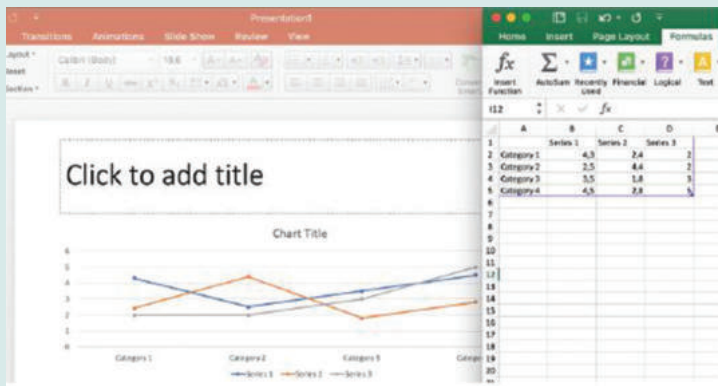
Membuat Diagram PowerPoint dengan Excel

Kamu harus menyiapkan presentasi dari data hasil pengamatan. Karena data pengamatannya tidak banyak, lebih praktis untuk langsung mengetikkan data dan membuat *chart* langsung pada *slides*.

Langkah-Langkah

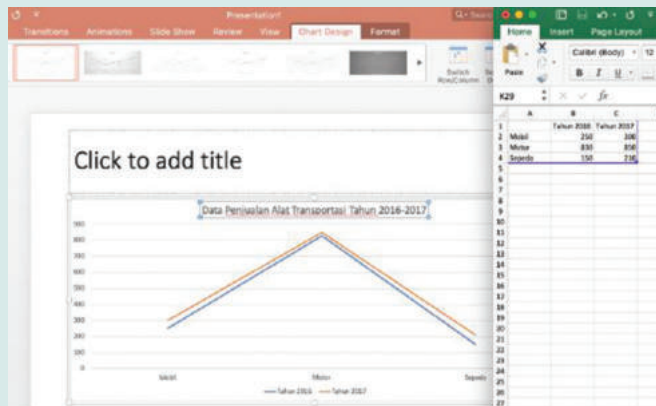
1. Buka aplikasi PowerPoint.
2. Klik menu Insert Chart.
3. Pilih jenis diagram yang diinginkan. Maka, akan muncul tampilan diagram secara otomatis pada PowerPoint dan tampilan *worksheet* Excel yang memuat datanya. Berikut diagram pada PowerPoint beserta *dataset* dalam bentuk Excel.





Sumber: Tangkapan layar Microsoft PowerPoint dan Excel

4. Lakukan perubahan pada data sesuai dengan yang diperlukan. Banyaknya baris maupun kolom dapat disesuaikan pula.
5. Tutup *worksheet*.
6. Ubahlah judul diagram dengan melakukan *double click* pada *Chart Title*. Tulis judul sesuai dengan yang diinginkan. Berikut tampilan setelah diagram pada aplikasi PowerPoint diberi judul.



Sumber: Tangkapan layar Microsoft PowerPoint dan Excel

Aktivitas 10.3-06-P

Membuat Video Presentasi dengan PowerPoint

Berkas presentasi yang telah dibuat dapat diekspor menjadi sebuah video. Hal ini dapat dilakukan dengan mengakses menu *File* → *Export* → *Create a Video*. Presentasikan videomu kepada teman-temanmu.

3. Penerapan OLE pada Integrasi Perangkat Lunak Perkantoran

Seperti telah disebutkan sebelumnya, jika menggunakan *Object Linking & Embedding* (OLE), suatu objek yang berasal dari aplikasi perkantoran lain dapat disisipkan ke aplikasi perkantoran yang sedang dibuat. Pada bagian ini, akan dilakukan berbagai aktivitas terkait proses integrasi menggunakan fitur OLE ini.



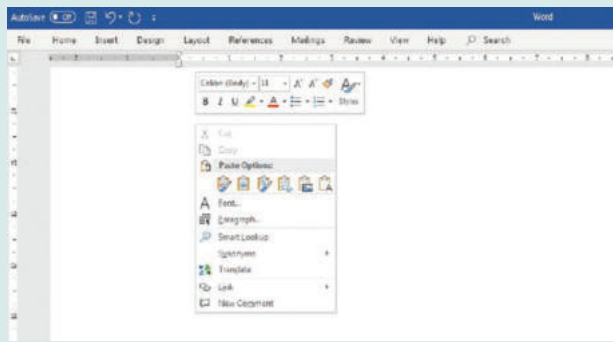
Aktivitas 10.3-07-P OLE antara Excel dan Word

Adakalanya, kamu perlu memasukkan data yang telah dibuat pada MS Excel ke suatu laporan yang disusun dengan menggunakan MS Word. Dalam hal ini, data telah ada sehingga tinggal dimasukkan saja. Tentunya, cara yang paling cepat ialah menggunakan *copy paste* biasa. Akan tetapi, dengan cara ini, data akan tersaji secara statis sehingga jika terjadi perubahan pada data sumbernya di Excel, data pada Word tidak berubah. Hal ini dapat diatasi dengan menggunakan fasilitas *Object Linking and Embedding* (OLE).

Langkah-Langkah

1. Buka aplikasi Excel.
2. Buatlah tabel pada Excel secukupnya.
3. Blok tabel yang sudah dibuat, kemudian *copy* tabel tersebut.
4. Buka Word.
5. Lakukan klik kanan.
6. Di bawah Paste Options, seperti terlihat pada gambar berikut, terdapat beberapa pilihan *paste*, pilih salah satu di antara Link & Keep Source Formatting (F) atau Link & Use Destination Styles (L). Perbedaan di antara keduanya hanyalah pada format tabelnya (misalnya font), yaitu menggunakan format sumber dari Excel (untuk pilihan Link & Keep Source Formatting) atau menggunakan format pada Word (untuk pilihan Link & Use Destination Styles).





Sumber: Tangkapan layar Microsoft Word

Selanjutnya, lakukan perubahan pada data sumber menggunakan perintah berikut ini.

7. Ubah data pada *file* Excel, misalnya dengan mengganti salah satu data pada tabel.
8. Perhatikan perubahannya pada Word di gambar berikut.

	Tahun 1	Tahun 2	Tahun 3
Kategori X	6,7	2,4	2
Kategori Y	2,5	3,4	5
Kategori Z	3,5	2,1	3

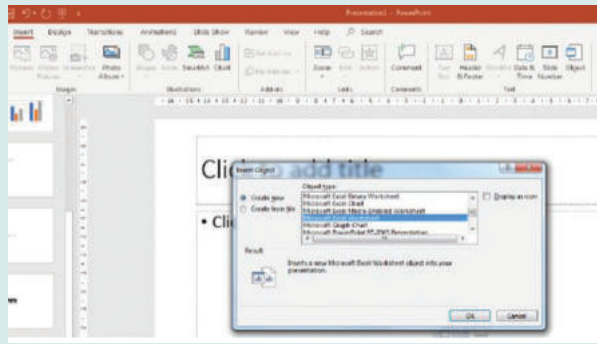
Sumber: Tangkapan layar Microsoft Word dan Excel

Aktivitas 10.3-08-P OLE antara PowerPoint dan Excel

Pada aktivitas ini, kamu akan menggunakan fitur OLE untuk menghubungkan tabel yang dibuat pada Excel ke slide pada PowerPoint.

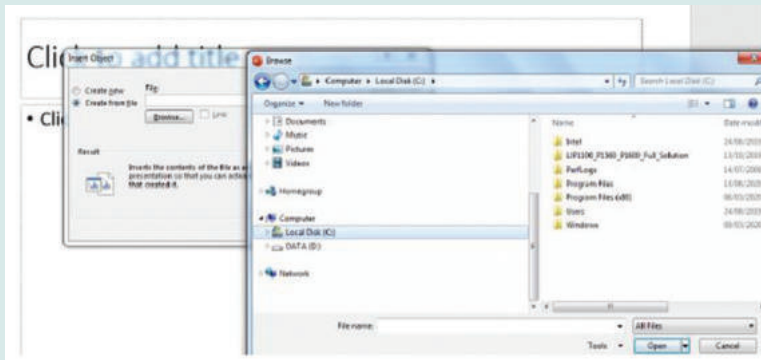
Langkah-Langkah

1. Buka aplikasi Excel. Buatlah tabel pada Excel secukupnya. Simpan *file* Excel.
2. Buka PowerPoint.
3. Pilih Insert Object. Perintah ini akan membuka jendela seperti pada gambar berikut.



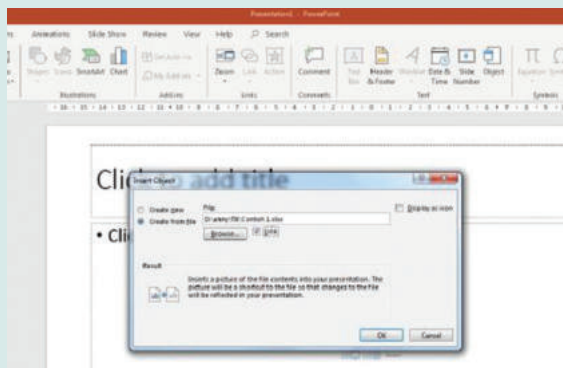
Sumber: Tangkapan layar Microsoft PowerPoint

4. Pilih *Create from file*, kemudian pilih *file* Excel yang telah dibuat sebelumnya pada langkah 1-3 seperti gambar berikut.



Sumber: Tangkapan layar Microsoft PowerPoint

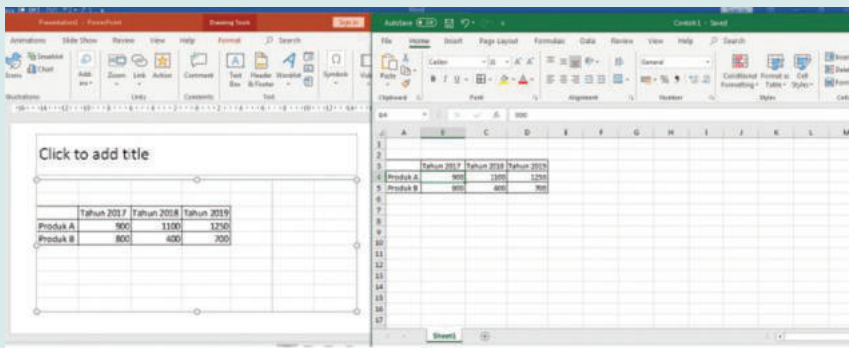
5. Beri tanda centang pada opsi *Link*, seperti gambar berikut.



Sumber: Tangkapan layar Microsoft PowerPoint

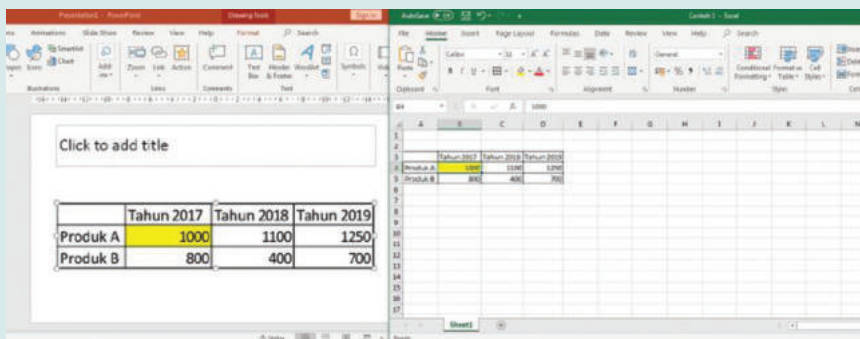
6. Klik *OK*. Maka, pada PowerPoint akan tampil tabel sesuai dengan tabel yang telah dibuat pada *file* Excel. Berikut tampilan tabel pada PowerPoint setelah object *file* Excel disisipkan.





Sumber: Tangkapan layar Microsoft PowerPoint dan Excel

7. Lakukan perubahan data pada *file* Excel, misalnya mengubah data seperti yang di *highlight* kuning. Data pada *slide* PowerPoint juga ikut berubah. Berikut tampilan tabel pada PowerPoint ketika data pada *file* Excel diubah.



Sumber: Tangkapan layar Microsoft PowerPoint dan Excel

C. Pemrosesan Dokumen dengan Perangkat Lunak Perkantoran

Dalam bagian ini, kamu akan mempelajari dua hal yang sering digunakan dalam memproses dokumen. Kedua hal itu ialah *mail merge* dan daftar isi.

1. Mail Merge

Mail Merge ialah fitur yang dapat kamu gunakan untuk menghasilkan surat, amplop, undangan, dan lain-lain secara berulang, tetapi terdapat beberapa komponen yang berbeda misalnya nama dan alamat yang dituju. Dalam hal ini, kita tidak perlu membuat dokumen sebanyak jumlah nama/tujuannya, tetapi hanya cukup menggunakan satu dokumen Word dan daftar nama/alamat yang dapat disimpan dalam daftar, basis data, atau *spreadsheet*.





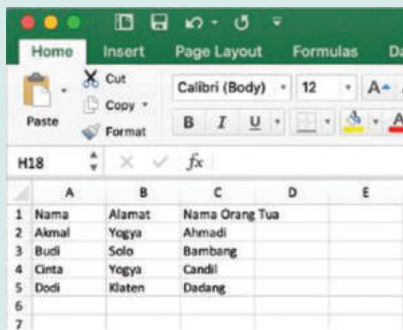
Aktivitas 10.3-09-P

Membuat Banyak Surat dengan *Mail Merge*

Pada aktivitas ini, kamu akan membuat banyak surat dengan fitur *mail merge*.

Langkah-Langkah

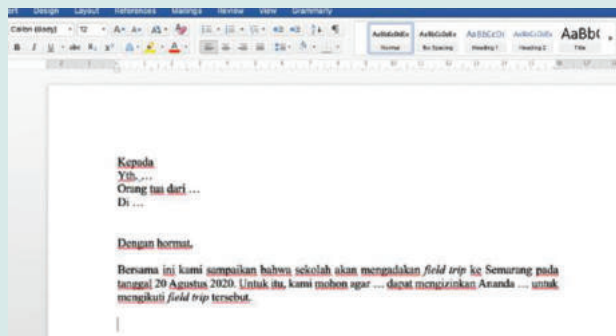
1. Buka Excel.
2. Buatlah tabel yang berisi data nama siswa, alamat, nama orang tua, kurang lebih seperti contoh di bawah ini.



	A	B	C	D	E
1	Nama	Alamat	Nama Orang Tua		
2	Akmal	Yogya	Ahmadi		
3	Budi	Solo	Bambang		
4	Cinta	Yogya	Candi		
5	Dodi	Klaten	Dadang		
6					
7					

Sumber: Tangkapan layar Microsoft Excel

3. Simpanlah *file* Excel tersebut.
4. Buka Word.
5. Buatlah dokumen surat kurang lebih seperti pada contoh berikut ini.

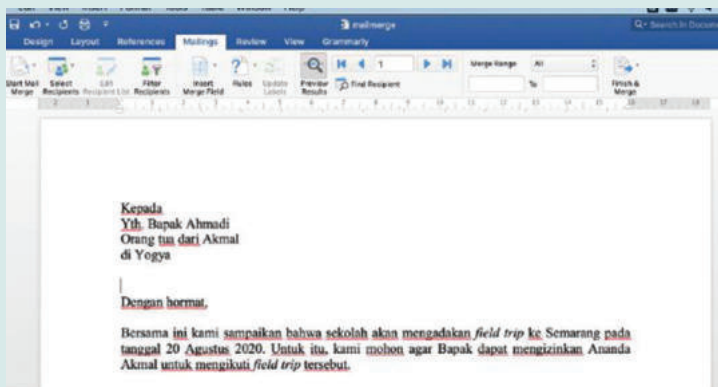


Sumber: Tangkapan layar Microsoft Word

6. Klik *Mailings* → *Select recipients* → *Use an existing list*.
7. Carilah *file* Excel yang telah dibuat sebelumnya pada langkah 1-3.
8. Klik *Open* → pilih *sheet* yang sesuai → klik *OK*.
9. Letakkan kursor pada sebelah kanan 'Yth.'.
10. Klik *Insert Merge Field* → pilih Hubungan.



11. Ulangi langkah 9 dan 10 untuk bagian-bagian lain dari dokumen surat sehingga diperoleh hasil seperti gambar berikut.
12. Untuk melihat hasilnya, klik Preview Result. Tampilan hasil setelah Mail Merge diterapkan ditunjukkan pada gambar berikut.



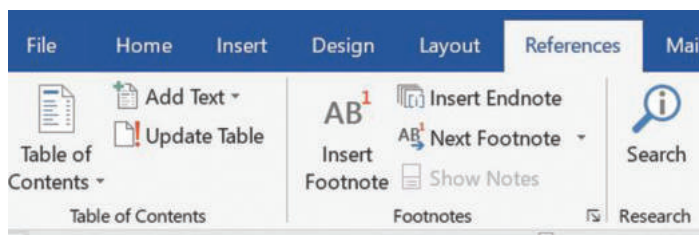
Sumber: Tangkapan layar Microsoft Word

13. Gerakkan panah ke kiri dan ke kanan untuk melihat tampilan data lainnya. Data pada setiap baris akan ditampilkan satu per satu.

2. Daftar Isi

Pernahkah kamu melihat daftar isi sebuah buku? Daftar isi terdiri atas apa saja? Andaikata kamu sedang menulis laporan, alangkah susahny jika harus mengetik judul bab satu per satu, kemudian mencatat nomor halamannya. Belum lagi jika teks berubah sehingga nomor halaman berubah. Adakah solusi?

Salah satu fasilitas yang tersedia di Microsoft Word ialah pembuatan Daftar Isi (*Table of Content*). Fiturnya ada pada salah satu pilihan *Reference* seperti gambar berikut.



Sumber: Tangkapan layar Microsoft Word

MS Word dapat membuat daftar isi secara otomatis dengan merujuk pada dokumen yang telah diberi *Styles* pada menu *Home* seperti gambar berikut.





Sumber: Tangkapan layar Microsoft Word

Daftar isi dibuat dari teks yang diberi *style* Heading 1, Heading 2, Heading 3, dst. Jadi, kamu harus menandai judul bab dan subbab yang akan ditampilkan pada daftar isi dengan *Style* Heading 1, Heading 2, atau Heading 3. Tampilan tingkat kerincian pada daftar isi dapat disesuaikan dengan kebutuhan.

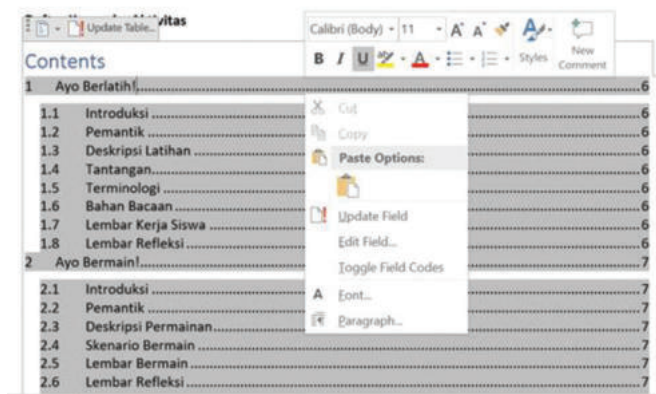
Pada aktivitas ini, kamu akan mengeksplorasi sendiri sebuah tutorial MS Word untuk melakukan otomatisasi pembuatan daftar isi laporan.

Langkah-Langkah

1. Posisikan kursor pada halaman di mana daftar isi akan dibuat (biasanya di awal).
2. Cari pilihan “*Table of Content*”, dan pilih format yang sesuai.
3. Daftar isi akan keluar secara otomatis.

Tutorial yang paling terpercaya tentunya diperoleh dari situs pengembang perangkat lunak, yaitu situs: <https://buku.kemdikbud.go.id/s/sts6y8> untuk membuat daftar isi.

Jika kamu mengubah teks sehingga judul bab tergeser dan nomor halaman berubah, daftar isi harus diperbarui. Caranya mudah. Posisikan kursor pada daftar isi, klik kanan Mouse, akan muncul menu “*Update Field*” seperti pada gambar berikut. Pilih menu tersebut, dan segera setelah itu, daftar isi secara otomatis berubah. Mudah, bukan? Perhatikan tampilan daftar isi setelah selesai dibuat berikut ini.



Sumber: Tangkapan layar Microsoft Word





Aktivitas 10.3-10-P

Membuat Daftar Isi Otomatis

Setelah paham tentang cara membuat daftar isi, kini, kamu akan berlatih menyesuaikan daftar isi dengan isinya.

Langkah-Langkah

Ambillah sebuah laporan yang pernah kamu buat sebelumnya. Berilah *styles* pada judul bab dan subbab pada laporan tersebut. Pastikan judul bab pada setiap bab diberi *style "Heading 1"* dan judul bab pada setiap subbab ialah *"Heading 2"*. Buatlah daftar isinya.

Setelah itu, tambahkan bab dari *file* yang lain dan tambahkan pada laporanmu, dan perbaharuilah daftar isimu. Mudah, bukan?

Lembar Kerja Siswa

Setelah berlatih membuat daftar isi, isilah lembar kerja berikut.

Tugas	Mudah	Agak Sulit	Sulit	Sangat Sulit
Saya berhasil membuat daftar isi laporan.				
Saya berhasil mengubah daftar isi secara otomatis setelah saya menambahkan satu bab.				
Saya berhasil mengubah format tampilan daftar isi.				
Ceritakanlah Pengalamanmu:				





Aktivitas 10.3-11-U Presentasi dengan Mading

Secara berkelompok, pilihlah satu produk seni dan budaya di tempat yang kalian sukai. Setelah itu, kalian akan membuat sebuah majalah dinding yang terstruktur dengan proses berikut.

Langkah-Langkah

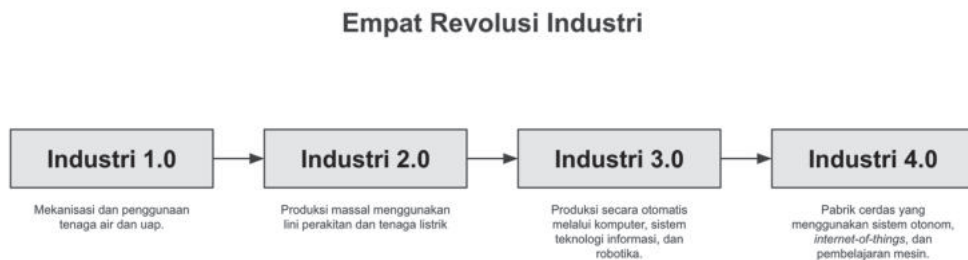
1. Pertama, kumpulkan media cetak di sekitar kalian yang terkait dengan tema tersebut, dapat berupa koran, majalah, atau poster yang tidak terpakai. Kemudian, kumpulkanlah beberapa gambar yang terkait dengan topik yang kalian pilih. Kalian juga dapat menggambar atau menambahkan foto yang kalian ambil sendiri.
2. Kedua, diskusikanlah secara berkelompok, bagaimana kalian akan menceritakan produk seni dan budaya tersebut. Setelah itu, tuliskanlah kerangka cerita kalian di selembar kertas.
3. Ketiga, siapkan karton atau buku gambar dengan ukuran yang cukup (misalnya A2). Setelah itu, tempelkanlah gambar-gambar yang telah kalian kumpulkan ke karton tersebut sesuai dengan urutan cerita yang telah kalian buat. Kalian dapat menambahkan tulisan atau hiasan lainnya untuk mempercantik mading kalian.
4. Tempelkanlah hasil karya kalian di dinding atau media lain yang disiapkan oleh guru. Setelah itu, secara bergiliran, presentasikanlah cerita kalian ke teman kalian.
5. Buatlah laporan yang menceritakan proses yang kalian lakukan dalam menghasilkan presentasi tersebut. Tuliskan juga refleksi kalian terhadap proses apa saja yang kalian lakukan. Identifikasi hal-hal yang efektif dan tidak efektif dalam proses pengerjaan aktivitas ini. Kemudian, berdasarkan refleksi tersebut, apa yang akan kalian lakukan berbeda setelah aktivitas ini?



D. Budaya Baru di Era Informatika

Sejalan dengan berkembangnya Informatika, terjadi pergeseran budaya. Berikut ini kamu akan belajar tentang hal yang dalam keseharian dekat dengan kamu. Hal tersebut ialah tentang konten digital, aspek ekonomi informatika, serta karier dan aspek hukum Informatika.

1. Produksi dan Diseminasi Konten Digital



Gambar 3.18 Era Revolusi Industri

Sumber: Dokumen Kemdikbudristek (2021)

Saat ini, manusia hidup di era digital yang sering disebut Industri 4.0. Dalam Industri 4.0, banyak hal dilakukan oleh manusia dengan memanfaatkan mesin cerdas berbasis komputer dan internet. Masyarakat sekarang ini disebut Masyarakat 5.0 (*Society 5.0*) karena hidup di dunia fisik sekaligus di dunia siber (maya) saat melakukan kegiatan dalam jaringan (daring). Banyak kegiatan dilakukan manusia secara fisik maupun secara daring dengan pemanfaatan teknologi komputer dan internet. Kamu tentu dapat mengingat pengalaman kegiatanmu yang didukung teknologi komputer dan internet, mulai dari berkomunikasi dengan media sosial, bermain *game online*, memesan makanan, memesan transportasi, mendaftar sekolah, belajar secara daring, dan sebagainya.

Dunia yang seperti ini menjadi makin cepat berubah, banyak hal menjadi tidak pasti, persoalan yang dihadapi manusia makin rumit, dan beragam cara untuk melakukan hal yang sama. Sebagai contoh, dulu orang harus pergi ke toko atau ke pasar untuk berbelanja. Penjual juga harus mempunyai bangunan kios atau toko. Pemilik toko mempekerjakan pegawai untuk mengurus gudang, menjadi kasir, dan menangani berbagai pekerjaan lain. Pembeli juga harus membawa uang tunai untuk membayar. Sekarang, orang dapat berbelanja dari rumah, tidak harus pergi ke mana-mana walaupun tidak ada penjual keliling yang datang. Penjual



juga dapat berjualan dari rumah tanpa perlu memiliki bangunan kios atau toko. Pembeli tidak harus mengambil uang di bank untuk membayar barang yang dibelinya karena dapat membayar dengan menggunakan uang digital. Contoh lain juga dapat kamu temukan atau alami sendiri. Pembelajaran daring selama masa pandemi ialah salah satunya. Di dunia industri, lebih banyak lagi hal yang kini dilakukan dengan bantuan teknologi komputer. Segala sesuatu diselesaikan secara otomatis karena sudah ada robot-robot yang mampu mengerjakan pekerjaan rutin yang berulang-ulang. Cara orang melakukan kegiatan menjadi berubah karena teknologi komputer dan internet yang dimanfaatkan di berbagai bidang.

Nah, situasi seperti contoh di atas memudahkan orang untuk melakukan kegiatan berbelanja, belajar, dan sebagainya. Namun, pikirkan, bagaimana nasib para pekerja yang semula bekerja sebagai kasir, pegawai gudang, pegawai yang melakukan pengepakan barang di pabrik, *teller* di bank, dan sebagainya? Ada banyak orang yang kehilangan pekerjaan karena teknologi mengambil alih hal-hal yang biasa mereka kerjakan. Di masa depan, pasti akan lebih banyak lagi peran teknologi dalam kegiatan manusia. Mungkin juga akan makin banyak jenis pekerjaan yang hilang. Sebaliknya, pencipta teknologi itulah yang makin dibutuhkan.

Menghadapi situasi tersebut, kamu yang sedang belajar di jenjang SMA perlu berpikir kritis dan mulai memikirkan masa depan. Tentu, tidak semua orang akan berkarier di bidang komputer. Zaman sekarang, kesempatan untuk berkarya di berbagai bidang sangat terbuka. Namun, hampir semua ilmu lain membutuhkan komputasi atau penyimpanan informasi serta pembentukan pengetahuan yang membutuhkan komputer. Bidang-bidang yang semula sulit terjangkau untuk diriset, makin tertolong dengan kemampuan komputasi, misalnya untuk menyimulasikan alam raya dan benda langit, atau menengok struktur mikro sel hidup atau material seperti molekul, atom, atau gen manusia. Semua profesi mulai ahli sejarah, ahli ekonomi, guru, apoteker, dokter, perawat, ilmuwan, insinyur, pedagang, semua membutuhkan alat bantu kerja yang “mengandung” komputer. Bahkan, di rumah pun, untuk sekadar berkomunikasi, ponsel pintar sudah dilengkapi dengan aplikasi. Produk hiburan seperti permainan, atau video yang dinikmati jutaan pengguna lewat kanal siaran, bahkan film drama ialah karya digital yang diciptakan para produsen dengan bantuan ahli digital media dan juga ahli seni.



2. Aspek Ekonomi Informatika

Industri perangkat keras dan perangkat lunak berkembang sejak ditemukannya komputer. Mula-mula, perusahaan perangkat keras *mainframe* dan mini-komputer yang muncul, misalnya IBM, Hitachi, UNIVAC, dan berbagai merek lainnya. Kemudian, dengan lahirnya PC (*personal computer*) dan diikuti dengan laptop, tablet serta telepon pintar, industri perangkat keras mulai tumbuh. Bukan hanya komputer, tetapi juga semua periferal seperti printer, alat input/output lainnya. Industri perangkat keras melahirkan industri manufaktur perangkat keras. Di Indonesia, industri ini masih dalam tahap perakitan.

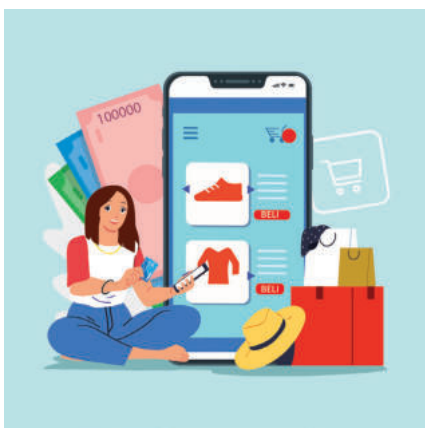
Seiring dengan berkembangnya industri di bidang perangkat keras, industri di bidang perangkat lunak juga berkembang. Mula-mula, industri perangkat lunak menyatu dengan perangkat keras, dan perangkat lunak dibangun untuk perusahaan atau pemerintahan, bukan untuk kehidupan individu sehari-hari. Ini juga diwarnai dengan cara kerja manusia dengan komputer *mainframe* dan mini-komputer, di mana pengguna tidak berhubungan langsung dengan perangkat keras. Generasi selanjutnya ialah di mana manusia mulai dapat berinteraksi dengan komputer lewat kabel atau jaringan lokal dengan menggunakan terminal berbasis tabung. Komputer hanya dipakai para profesional di kantor, atau untuk para periset di laboratorium. Dengan lahirnya PC, komputer makin dipakai di rumah dan untuk kehidupan sehari-hari, perkembangannya didukung dengan lahirnya Internet. Komputer sekarang sudah dalam genggam manusia dalam bentuk telepon pintar. Industri perangkat lunak berkembang pesat untuk memenuhi kebutuhan manusia berinteraksi dengan komputer, atau manusia berinteraksi dengan manusia lainnya baik dalam kelompok maupun individual.

Berbeda dengan industri manufaktur perangkat industri perangkat lunak makin meningkat. Hal itu disebabkan, makin murah komputer dan perangkat lainnya seperti telepon pintar. Makin banyak diciptakan perangkat lunak karena perangkat lunak makin banyak menunjang kehidupan sehari-hari dalam berbagai bidang. Penggunaan perangkat lunak misalnya untuk berkomunikasi dan unjuk diri di dunia digital, edukasi, perdagangan/bisnis,

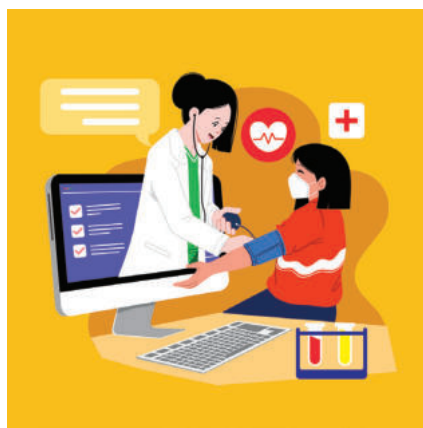
Inovasi di abad digital melahirkan bisnis bidang teknologi informasi yang makin dibutuhkan. Hampir semua sektor pemerintahan dan bisnis mengubah cara berbisnis dengan menyediakan jasa layanan *online*. Anak-anak muda yang kreatif menumbuhkan perusahaan-perusahaan dari garasi, yang melahirkan perusahaan raksasa di bidang teknologi informasi seperti Microsoft, Apple,



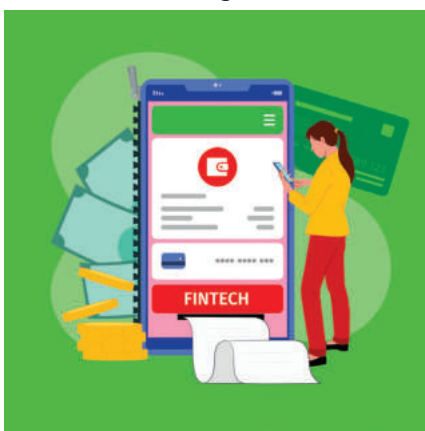
Google, dan Facebook. Industri berbasis teknologi informasi tumbuh pesat, yang terkenal, misalnya Silicon Valley. Berbeda dengan model bisnis biasanya, model bisnis di bidang teknologi informasi ini mampu mendominasi akibat adopsi pasar yang sangat cepat. Suatu bisnis media sosial, misalnya, dapat memperoleh pengguna hingga milyaran hanya dalam waktu kurang dari 5 tahun. Perkembangan ini mendisrupsi bursa perusahaan terkemuka di dunia yang saat ini didominasi oleh perusahaan berbasis teknologi informasi. Jika pada tahun 2013, lima perusahaan dengan pasar terbesar terdiri atas Apple (Teknologi), Exxon (Minyak dan Gas), Berkshire Hathaway (Konglomerasi Investasi), PetroChina (Minyak dan Gas), serta Walmart (Retail), pada tahun 2018, posisi lima besar tersebut seluruhnya dikuasai oleh perusahaan di bidang teknologi informasi seperti Apple, Alphabet, Microsoft, Amazon, dan Tencent.



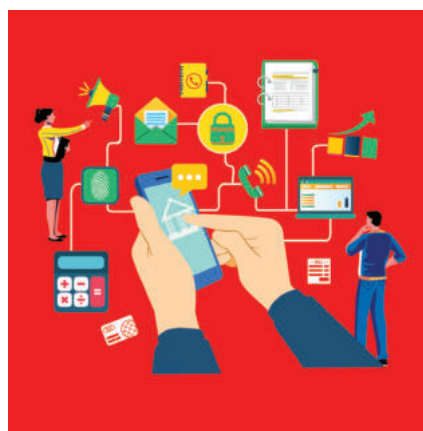
Ilustrasi tentang e-commerce



Ilustrasi tentang e-health



Ilustrasi tentang fintech



Ilustrasi tentang digital services

Gambar 3.19 Ilustrasi Ekonomi Kreatif di Indonesia



Pertumbuhan pesat dalam teknologi informasi telah memberikan nilai tambah signifikan dalam berbagai aspek kehidupan manusia. Dalam lingkup bisnis, pengadopsian teknologi informasi menjadi suatu keharusan untuk menjaga daya saing dan menciptakan keunggulan kompetitif. Perusahaan tidak hanya menginvestasikan dalam teknologi untuk menjaga posisinya, tetapi juga untuk mencapai keunggulan kompetitif melalui inovasi teknologi. Di sisi konsumen, teknologi informasi memberikan kemudahan dalam berbagai aspek kehidupan, seperti komunikasi melalui media sosial, perdagangan *online*, dan perjalanan yang lebih efisien. Dalam beberapa kasus, munculnya aplikasi “*super app*” memadukan berbagai layanan dalam satu platform.

Dalam konteks Indonesia, perkembangan teknologi telah mendorong pertumbuhan berbagai perusahaan rintisan (*startup*), terutama di sektor teknologi. Data menunjukkan bahwa sektor aplikasi dan *video game* telah memberikan kontribusi signifikan terhadap pendapatan domestik bruto. Indonesia juga menjadi rumah bagi beberapa perusahaan dengan nilai valuasi tertinggi di wilayah Asia Tenggara, seperti Go-Jek, Traveloka, Tokopedia, dan Bukalapak. Pemerintah Indonesia aktif mengembangkan program untuk mendukung perkembangan perusahaan rintisan dan meningkatkan jumlah tenaga kerja digital.

Nilai valuasi perusahaan rintisan tidak hanya dilihat dari keuntungan finansial, tetapi juga dari jumlah pengguna dan investasi yang diterima dari investor. Ini menjadi ukuran dalam proses akuisisi atau penjualan kepemilikan perusahaan rintisan. Para pendiri perusahaan rintisan menciptakan inovasi mereka dengan berdasarkan pemikiran terhadap permasalahan masyarakat yang dapat diatasi dengan teknologi informasi. Oleh karena itu, perlindungan hukum atas kekayaan intelektual mereka sangat penting, yang mencakup hak kekayaan intelektual termasuk hak atas perangkat lunak.

Hasil pemikiran intelektual dalam bidang teknologi informasi menciptakan inovasi, yang pada gilirannya dapat menjadi produk yang bermanfaat bagi masyarakat dan berkontribusi pada perekonomian serta martabat bangsa. Bidang informatika memiliki potensi besar untuk menggerakkan pengindustrian inteligensi dengan menghasilkan produk-produk yang meningkatkan fungsi ilmu pengetahuan dan teknologi dalam kehidupan sosial, serta mendorong kemampuan Indonesia dalam berbangsa dan bernegara.



Aktivitas 10.3-12-U

Aspek Ekonomi Produk Informatika

Pada bagian ini, kamu akan berdiskusi secara berkelompok pada satu atau lebih topik berikut. Ikutilah instruksi dari guru di kelas.

Langkah-Langkah

- Carilah di Internet, berbagai “gerai” digital yang disebutkan di atas. Golongkanlah sesuai area bisnisnya misal: transportasi, travel, finansial, kesehatan, makanan dan minuman, jasa, dll. Rangkum dalam sebuah tabel berikut. Perhatikanlah, bahwa sesuai dengan namanya, biasanya, perusahaan memulai dengan satu area bisnis tertentu. Dalam perjalanannya, mereka membuka layanan “satu pintu” untuk berbagai transaksi lainnya. Menurutmu, mengapa mereka melakukan hal itu?

Nama Perusahaan	Daftar Layanan	Nama Deskripsi Layanan Utamanya

- Beberapa anak muda yang sering bepergian ke luar negeri menerima jasa titipan belanja. Mereka menawarkan jasanya secara *online* lewat media sosial. Berkat itu, mereka dapat menikmati lebih sering bepergian “gratis”, dan yang menitip pun senang karena mendapatkan barang yang sulit diperoleh di Indonesia dengan mudah tanpa harus bepergian ke luar negeri yang ongkosnya mahal, dan untuk barang tertentu dapat bebas pajak. Media sosial yang semula dimaksudkan untuk berkomunikasi dan bergaul di dunia maya menjadi sarana bisnis. Menurutmu, apakah hal ini sah-sah saja?
- Generasi muda zaman sekarang makin sedikit membaca koran. Di zaman dulu, setiap pagi, orang menanti kedatangan pengantar yang



membawa koran ke rumah, atau membeli koran yang ditawarkan di perempatan jalan. Beberapa koran sudah beralih ke koran digital, dan selain tetap menjual koran dalam media cetak, mereka membuka situs Internet menyediakan layanan berita “berbayar” dan juga ada yang secara “gratis”. Menurutmu, bagaimana masa depan koran dan profesi pengantar koran tersebut?

4. Pernahkah kamu membuka aplikasi untuk menonton video atau musik? Aplikasi untuk menonton video tersebut memungkinkan kamu menonton video yang “dititipkan” di situs mereka, dan jika dibuka publik, siapa saja dapat membuka videomu. Dengan aplikasi pemutar video, kamu dapat menikmati berita, belajar, nonton film hiburan dan sebagainya. Namun, sering kali di awal, ditayangkan iklan yang harus dibaca dan hanya dapat dilewati (*skip*) setelah beberapa detik. Berikan pendapatmu:
 - a. Apakah kamu “terganggu” dengan iklan yang muncul tanpa kamu minta?
 - b. Mengapa ada iklan tersebut? Andaikata kamu pemilik situs video atau berita, mengapa kamu mengizinkan perusahaan lain menayangkan iklan di lapakmu? Tanyakan orang tua atau keluargamu yang berasal dari generasi sebelumnya. Mereka tidak mengenal aplikasi video dan jika ingin menonton film, mereka pergi ke gedung bioskop. Adakah iklan yang diputar pada saat menonton film?
 - c. Seandainya kamu ingin menonton tanpa iklan sama sekali, mungkinkah mematikannya? Bagaimana caranya?
5. Selain koran, buku juga makin banyak dalam bentuk elektronik. Sama dengan perangkat lunak, ini menimbulkan masalah karena dengan mudah dapat dibagikan *file*-nya. Jika dalam perpustakaan dunia nyata buku dapat dikembalikan, tidak demikian halnya dengan beberapa perpustakaan digital yang dioperasikan. Menurutmu, apa yang paling perlu diperhatikan, di mana kamu hidup dalam sebuah dunia digital, yang sekali barang diberikan tak dapat ditarik kembali? Dalam kaitannya dengan aspek ekonomi, apa pendapatmu yang perlu dilakukan?



Ayo, Renungkan!

Setelah mengerjakan aktivitas di atas, renungkanlah pertanyaan-pertanyaan berikut dan catat jawabanmu di buku kerjamu.

1. Hal baru apa yang kamu pelajari dari aktivitas hari ini?
2. Apakah sekarang kamu memiliki keinginan untuk menciptakan inovasi berbasis informatika untuk meningkatkan kegiatan ekonomi?
3. Inovasi informatika apa yang terpikirkan olehmu setelah melaksanakan aktivitas ini?

3. Karier dan Aspek Hukum Informatika

Perangkat lunak merupakan hasil buah pikir (karya intelektual) manusia, seperti halnya buku atau karya lainnya. Oleh karena itu, perangkat lunak harus dihargai sebagai salah satu hak kekayaan intelektual. Sayangnya, karena bentuknya yang berupa benda digital, perangkat lunak lebih mudah untuk diperoleh melalui cara yang tidak legal/benar secara hukum, atau dengan istilah umumnya ialah pembajakan. Perlu ditekankan bahwa melakukan pembajakan perangkat lunak itu merupakan tindakan melawan hukum.

Untuk dapat memahami hal-hal apa saja yang boleh/tidak boleh dilakukan terhadap suatu perangkat lunak, kamu perlu melihat jenis lisensi apa yang disediakan oleh pembuat perangkat lunak tersebut. Lisensi perangkat lunak mencakup izin, hak, dan pembatasan yang diberlakukan atas perangkat lunak, baik berupa suatu komponen atau program berdiri sendiri. Penggunaan suatu perangkat lunak tanpa lisensi dapat dianggap pelanggaran atas hak eksklusif pemilik menurut hukum hak cipta atau, kadang, paten dan dapat membuat pemilik menuntut pelanggarnya.

Dalam suatu lisensi, penerima lisensi diizinkan untuk menggunakan perangkat lunak berlisensi sesuai dengan persyaratan khusus dalam lisensi. Pelanggaran persyaratan lisensi, bergantung pada lisensinya, dapat menyebabkan pengakhiran lisensi, dan hak pemilik untuk menuntut pelanggarnya. Lisensi dapat dianggap sebagai sebuah perjanjian hukum yang resmi antara pembuat perangkat lunak yang menyediakan lisensi tersebut (*licensor*) dan pihak pengguna perangkat lunak yang menerima lisensinya (*licensee*).



Ada beberapa jenis lisensi perangkat lunak. Berikut ini ialah rangkuman beberapa jenis utama dari lisensi perangkat lunak.

a. Lisensi Komersial

Lisensi komersial ialah lisensi yang bersifat paling restriktif (mengikat). Sesuai namanya, lisensi ini biasanya diterapkan untuk perangkat lunak yang berbayar, di mana pengguna hanya diperbolehkan menggunakan perangkat lunak setelah membayar suatu harga tertentu kepada pembuat perangkat lunak. Selain itu, dalam penggunaannya, pun biasanya terdapat berbagai peraturan yang mengatur apa saja yang boleh/tidak boleh dilakukan oleh pengguna selama masa pemakaian. Peraturan itu misalnya: hanya boleh meng-*instalasi* dalam jumlah tertentu atau pada perangkat tertentu saja, tidak boleh memperbanyak/menggandakan perangkat lunak di luar yang telah disepakati, tidak boleh mengubah kode program atau perilaku perangkat lunak di luar metode yang diperbolehkan, dan lain-lain. Perangkat lunak dengan lisensi komersial ini tentunya rentan terhadap tindakan pembajakan, terutama perangkat lunak yang diperlukan oleh banyak orang, seperti: sistem operasi, pengolah kata, pengolah gambar, dan lain-lain.

b. Lisensi Gratis (*Freeware*)

Sebagian perangkat lunak mungkin disediakan secara gratis, dan dapat diunduh serta di-*install* dari sebuah sumber tertentu yang disediakan oleh penggunanya. Meskipun perangkat lunak tersebut disediakan secara gratis, bukan berarti tidak ada batasan yang diatur dalam lisensinya. Misalnya, meskipun sebuah perangkat lunak berlisensi gratis, tetapi mungkin tidak memperbolehkan pengguna untuk mendistribusikan ulang perangkat lunak tersebut, atau mengubah kode programnya. Sebuah variasi lisensi yang mirip (tetapi tidak sama persis) dengan lisensi gratis ialah lisensi *shareware*. Lisensi ini memperbolehkan pengguna untuk secara gratis mencoba perangkat lunak tersebut selama masa percobaan tertentu yang ditetapkan (misal selama 1 bulan), dan setelah itu, pengguna diminta untuk membayar untuk menggunakan perangkat tersebut lebih lanjut.

c. Lisensi *Free/Open Source*

Lisensi ini ialah lisensi yang memperbolehkan pengguna untuk tidak hanya menggunakan perangkat lunak tersebut, tetapi juga untuk melihat, mengubah, dan mendistribusikan kode sumber program yang digunakan untuk membuat perangkat lunak tersebut. Dengan menggunakan lisensi ini,



penulis perangkat lunak dapat membagikan kode program yang dibuatnya, agar dapat ditingkatkan oleh orang lain, dapat menimbulkan kolaborasi/ kerja sama untuk membangun sebuah perangkat lunak berkualitas yang dapat digunakan oleh khalayak umum secara gratis (tanpa membayar). Saat ini, *open source* sudah menjadi sebuah gerakan yang didukung oleh banyak orang, organisasi, perusahaan serta pemerintahan di seluruh dunia. *Open source* telah mampu menghasilkan berbagai perangkat lunak berkualitas, mulai dari sistem operasi (misal Linux), perangkat pengolah dokumen (*office*), penjelajah Internet (*browser*), klien *email*, pengolah gambar, pemutar dokumen multimedia (suara dan video) dan lain sebagainya. Salah satu pelopor gerakan *open source* ialah proyek GNU (GNUis Not Unix) yang mulai memopulerkan sistem operasi yang gratis. Pada pertengahan 1980-an, proyek GNU mengeluarkan lisensi-lisensi perangkat lunak bebas yang terpisah untuk setiap paket perangkat lunaknya. Kesemuanya digantikan pada 1989 dengan versi satu dari Lisensi Publik Umum (GNU), *General Public License* disingkat (GPL). Versi 2 dari GPL yang dirilis pada 1991, dan diikuti dengan dengan GPL versi 3 pada tahun 2007.

d. Lisensi Domain Publik

Lisensi ini ialah jenis lisensi di mana semua hak cipta terhadap perangkat lunak telah dilepaskan (oleh si pembuat perangkat lunak) sehingga kepemilikan perangkat lunak tersebut diberikan kepada masyarakat umum. Setiap orang diperbolehkan untuk menggunakan, menggandakan, serta mendistribusikan ulang, ataupun mengubah kode programnya tanpa ada batasan.



Gambar 3.20 Komik tentang Seseorang yang Sedang Membeli Komputer



Lisensi perangkat lunak bebas memberikan banyak kemajuan di bidang pengembangan perangkat lunak dan juga perangkat keras. Hal itu karena adanya keterbukaan informasi tentang perangkat lunak. Perdebatan mengenai hak kekayaan intelektual perangkat lunak dikaitkan dengan kebebasan memakai dan memodifikasi ini masih akan menjadi perdebatan yang tidak selesai, tetapi akan membawa banyak kemajuan dan inovasi.

Yayasan Perangkat Lunak Bebas (FSF, fsf.org), sebuah organisasi yang mengeluarkan Definisi Perangkat Lunak Bebas, merilis daftar lisensi perangkat lunak bebas. Daftar tersebut membedakan antara lisensi perangkat lunak bebas yang kompatibel dan yang tidak kompatibel dengan lisensi pilihan FSF, yaitu Lisensi Publik Umum GNU, yang merupakan sebuah lisensi *copyleft*. Dalam daftar tersebut, juga tercantum lisensi-lisensi yang dianggap FSF tidak bebas untuk beberapa alasan tertentu.

Di Indonesia, kesadaran masyarakat tentang lisensi perangkat lunak masih perlu ditingkatkan. Masih sering ditemui adanya pelanggaran terhadap lisensi perangkat lunak. Hal itu terjadi baik melalui pembajakan, penggandaan perangkat lunak yang tidak sesuai dengan lisensi, serta penjualan perangkat lunak tanpa lisensi resmi. Padahal, sudah ada undang-undang yang mengatur mengenai hal ini. Di Indonesia, lisensi perangkat lunak tercakup dalam konsep Hak Kekayaan Intelektual (HaKI), yang diatur dalam Undang-Undang Nomor 7 Tahun 1994 tentang Pengesahan WTO (*Agreement Establishing The World Trade Organization*), dan lebih spesifik lagi pada Pasal 40 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta (UU Hak Cipta). Istilah HaKI atau Hak atas Kekayaan Intelektual merupakan terjemahan dari *Intellectual Property Right* (IPR), memiliki pengertian pemahaman mengenai hak atas kekayaan yang timbul dari kemampuan intelektual manusia, yang mempunyai hubungan dengan hak seseorang secara pribadi, yaitu hak asasi manusia (*human right*). HaKI merupakan hak eksklusif yang diberikan suatu hukum atau peraturan kepada seseorang atau sekelompok orang atas karya ciptanya.

Untuk memperbaiki kondisi penghargaan terhadap hak cipta terkait penggunaan perangkat lunak ini di Indonesia, perlu kamu pahami dan tumbuhkan kesadaran mengenai pentingnya mematuhi aspek legal lisensi perangkat lunak dalam kehidupan sehari-hari. Banyak orang yang dalam kehidupan sehari-harinya tidak/jarang melanggar hukum. Namun, dalam penggunaan perangkat lunak, mereka tidak begitu memperdulikan



apakah perangkat lunak yang digunakan memiliki lisensi yang benar/tidak. Padahal, dari kacamata hukum, tindakan menggunakan perangkat lunak dengan lisensi yang tidak legal sama saja dengan tindakan melanggar hukum lain, misalnya mencuri. Oleh karena itu, perlu ditekankan pemahaman bahwa pelanggaran lisensi perangkat lunak juga merupakan kegiatan melawan hukum yang harus dihindari.

Alasan lain yang juga menjadi penyebab mengapa masih banyak yang menggunakan perangkat lunak tanpa lisensi yang benar ialah karena permasalahan biaya: banyak perangkat lunak populer yang diperlukan memiliki lisensi komersial dan dijual dengan harga yang tidak murah dan masih memberatkan bagi banyak orang di Indonesia. Sebagai salah satu solusi permasalahan ini ialah dengan meningkatkan kesadaran dan pengetahuan rakyat Indonesia. Mereka perlu mengetahui adanya berbagai alternatif perangkat lunak dengan lisensi gratis ataupun *open source* yang memiliki kemampuan dan fungsionalitas yang tidak kalah dari perangkat lunak komersial. Dalam banyak hal, perangkat lunak ini sudah mencukupi untuk kebutuhan sebagian besar pengguna.



Aktivitas 10.3-13-U

Aspek Hukum Produk Informatika

1. Di dunia nyata, banyak barang bermerek yang mahal harganya ditiru, seperti tas, koper, baju, dan sebagainya. Pembuat barang palsu mencantumkan merek yang sama pada barang yang diproduksi. Hal ini juga terjadi pada perangkat lunak atau produk dalam bentuk digital lainnya, yang secara populer biasa disebut sebagai perangkat lunak/video/buku “bajakan”. Menurutmu, apa bedanya membajak barang nyata dan barang digital? Apakah tindakan membajak itu boleh dilakukan?
2. Banyak perangkat lunak disediakan untuk dapat dipakai secara legal dan “gratis”. Tentu, kamu sudah menikmati beberapa perangkat lunak semacam itu. Dari aspek ekonomi perangkat lunak, kamu tahu bahwa mengembangkan perangkat lunak membutuhkan biaya. Menurutmu, bagaimana perusahaan pengembang perangkat lunak bisa hidup?
3. Carilah di Internet, perangkat lunak berlisensi dan “padanan”-nya (artinya yang fungsi utamanya sama, walau mungkin tidak sama persis).



No.	Jenis Perangkat Lunak	Merek Berlisensi	Gratis
1.	Sistem Operasi <i>Smartphone</i>		
2.	Sistem Operasi PC		
3.	Paket <i>Office</i>		
4.	Pengolah Kata		
5.	Pengolah Lembar Kerja		
6.	Pengolah Presentasi		
7.	Editor Gambar		
8.	Kompiler Bahasa Pemrograman C		
9.	Pemutar Media (Suara, Video)		

- Carilah di Internet halaman web atau buku yang dapat diunduh dengan berlisensi gratis, misalnya buku di <https://csunplugged.org>. Lisensinya disebut sebagai CCL (Common Creative Licence). Apa saja yang boleh dilakukan dengan buku yang kamu *download* dengan lisensi tersebut? Ada berapa jenis lisensi CCL?
- Carilah referensi di Internet (Wikipedia, Blog, Berita, pencarian di Google) untuk dapat menjawab pertanyaan berikut: “Apa yang menjadi motivasi bagi orang atau perusahaan untuk menulis sebuah program untuk perangkat lunak, dengan lisensi *open source*?”
- Ada beberapa jenis lisensi *open source* yang berbeda, dengan konsekuensi/ akibat yang berbeda-beda juga bagi pengguna perangkat lunak dengan lisensi tersebut (baik pengguna yang sekadar menggunakan, ataupun yang akan memodifikasi kode program perangkat lunak tersebut). Contoh dua buah lisensi *open source* yang cukup berbeda ialah GPL (v1 – v3) dan BSD. Carilah referensi di Internet untuk dapat memahami, apa perbedaan utama antara kedua jenis lisensi *open source* tersebut!



E. Budaya Karier dan Beragam Sertifikasi di Bidang Informatika

Dalam bagian ini, kamu akan mengenal apa saja profesi di bidang Informatika. Kamu juga akan belajar mengapa penting profesi itu harus disertifikasi.

1. Karier di Bidang Informatika

Mereka yang bekerja di bidang informatika jarang sekali mengatakan bahwa mereka bekerja di bidang informatika. Biasanya, mereka lebih cenderung menyebut mereka memiliki karier di bidang teknologi informasi atau yang dalam bahasa Inggris disebut *information technology* (IT). Belakangan, ada banyak profesional yang menggunakan berbagai macam posisi pekerjaan yang lebih spesifik seperti *system analyst*, *back-end engineer*, *front-end engineer*, *devops engineer*, *user experience designer*, *data scientist*, *product manager*, hingga *chief technology officer* (Gambar 3.40). “Posisi” tersebut terus berkembang dan berevolusi mengikuti perkembangan teknologi dan kebutuhan industri yang sedemikian cepatnya.



(a) Dua orang pemrogram melakukan teknik pemrograman berpasangan (*pair programming*) untuk menghasilkan program yang berkualitas tinggi.





(b) Tim dari keilmuan informatika, teknik, dan elektro bekerja sama mengembangkan sebuah robot yang digunakan dalam proses manufaktur.



(c) Seorang analis sistem sedang melakukan diskusi grup dengan target pengguna dari perangkat lunak yang akan mereka buat.

Gambar 3.21 Ilustrasi Beberapa Profesi di Bidang Informatika

Sumber: (a) Vajrapani666/Wikimedia Commons, (b) Steve Juvetson/Wikimedia Commons, (c) NSaad (WMF)/Wikimedia Commons

Pada saat ini, kebanyakan orang awam memahami Informatika sebagai disiplin ilmu yang mempelajari fenomena di sekitar komputer. Fenomena ini



meliputi desain komputer dan proses komputasi, representasi objek informasi dan transformasinya, perangkat keras, perangkat lunak, efisiensi, dan kecerdasan mesin. Di Eropa, disiplin ilmu komputer disebut sebagai “informatika”, sedangkan di Amerika Serikat disebut sebagai “*computing*”. Pada unit ini, kita akan melihat lebih dalam mengenai makna dari suatu profesi di bidang informatika, yang lebih luas daripada sekadar “mencari nafkah”, serta pengelompokannya di dalam dunia keprofesian.

Menurut Prof. Peter J. Denning, seorang peneliti terkemuka yang banyak melahirkan tulisan ilmiah terkait keprofesian di bidang informatika, suatu profesi harus memiliki empat ciri yang khas. Keempat ciri tersebut seperti berikut,

- a. Mendalami suatu bidang yang menjadi perhatian manusia dalam waktu yang lama. Profesi informatika berperan untuk memenuhi kebutuhan manusia atas kemampuan komputasi dan komunikasi yang efektif. Hal ini juga didorong oleh perkembangan teknologi yang makin cepat dan makin melebur dengan seluruh sendi-sendi kehidupan manusia. Pada bidang ini, seorang profesional di bidang informatika akan berusaha menyelesaikan berbagai permasalahan dengan memanfaatkan kemampuan komputer.
- b. Mengodifikasikan kumpulan prinsip atau pengetahuan konseptual. Kumpulan prinsip-prinsip ini diwujudkan dalam berbagai bentuk. Kurikulum yang dibuat oleh ACM dan IEEE, program studi di perguruan tinggi, dan lembaga-lembaga pelatihan merupakan salah satu wujud kodifikasi prinsip-prinsip di bidang informatika yang dapat diakses oleh seorang profesional di bidang informatika.
- c. Mengodifikasikan kumpulan praktik, yang juga meliputi kompetensi. Hal ini salah satunya diwujudkan dan diakui melalui sertifikasi kompetensi profesional yang telah dijelaskan pada unit sebelumnya.
- d. Menetapkan standar kompetensi, etika, dan praktik. Hal ini menjadi standar bagi seorang profesional di bidang informatika untuk berperilaku dan bersikap dalam menjalankan tugas-tugas profesinya. Hal ini bermanfaat agar profesional di bidang informatika memiliki integritas dan dapat dipercaya.

Berdasarkan uraian di atas, apabila kamu ingin menjadi seorang profesional di bidang informatika, keempat hal tersebut perlu kamu pahami. Pekerjaan bukan sekadar sebagai sarana untuk mendapatkan penghasilan, tetapi untuk memberikan manfaat bagi masyarakat. Selain menguasai kumpulan prinsip dan praktik, kamu pun dapat berpartisipasi aktif untuk merumuskan suatu prinsip atau



praktik baru dan ikut menyebarkannya kepada orang lain di profesi yang sama. Hal ini dapat kamu lakukan dengan berbagai cara, seperti menulis, memberikan ceramah, atau membuat pelatihan. Terakhir, kamu harus mengetahui etika serta standar kompetensi dan praktik di bidang itu untuk memastikan kemampuan dan sikap yang kamu tunjukkan menunjukkan integritas dari profesimu.

Berdasarkan peran informatika, Prof. Denning membagi profesi di bidang informatika dalam tiga kategori, yaitu profesi yang sangat spesifik di bidang informatika, profesi yang sarat dengan informatika, serta profesi yang memberikan dukungan teknologi informasi. Contoh-contoh pekerjaan tersebut dapat dilihat pada Tabel 3.5 di bawah ini.

Tabel 3.5 Profesi di Bidang Informatika

Profesi yang Sangat Spesifik di Bidang Informatika	Profesi yang Sarat dengan Informatika	Profesi yang Memberikan Dukungan Teknologi Informasi
• Kecerdasan buatan • Ilmu komputer • Rekayasa komputer • Ilmu komputasional • Rekayasa basis data • Grafika komputer • Interaksi manusia-komputer • Rekayasa jaringan • Sistem operasi • Rekayasa kinerja • Robotika • Komputasi saintifik • Arsitektur perangkat lunak • Rekayasa perangkat lunak • Keamanan sistem	• Dirgantara • Bioinformatika • Sains kognitif • Sains perpustakaan digital • E-commerce • Layanan finansial • Rekayasa genetika • Ilmu informasi • Kebijakan publik • Privasi	• Teknisi komputer • Teknisi <i>helpdesk</i> • Teknisi jaringan • Pelatih IT profesional • Spesialis keamanan siber • Administrator sistem • Desainer layanan web • Desainer identitas web • Administrator basis data



Setiap bidang profesi akan dikaitkan dengan profesional (pekerjanya). Secara umum, pekerjaan di bidang informatika akan dikelompokkan ke bidang ilmu yang sesuai dengan bidang program studi yang ditempuh sebelum bekerja. Karena bidang informatika mencakup teori sampai dengan praktik, bidang ilmu dasar sampai dengan teknologi dan aplikasinya, lulusan bidang informatika dapat memilih jalur sains atau teknologi, menjadi ilmuwan (*scientist*), atau menjadi insinyur (*engineer*). Ilmuwan “mengamati” dan mengeksplorasi dunia untuk dapat menjelaskan secara ilmiah fenomena alam yang terjadi, sementara insinyur lebih fokus pada penciptaan produk-produk yang diciptakan untuk menunjang kehidupan dengan memanfaatkan hasil ilmuwan, sekaligus menemukan dan menciptakan ilmu rekayasa yang baru.

Saat ilmuwan perlu membuktikan suatu konsep ilmiah, dia perlu melakukan observasi, analisis serta merancang dan melaksanakan eksperimen. Insinyur menciptakan produk, dengan melakukan tahapan berikut: analisis, perancangan, implementasi, pengujian, pemasangan dan pengoperasian. Pada saat pengoperasian, dibutuhkan pemeliharaan produk. Insinyur dibantu oleh teknisi, dan pemakaian produk membutuhkan operator.

Secara lebih spesifik, profesi di bidang informatika dapat dibagi-bagi menjadi beberapa posisi pekerjaan (*job title*) yang memiliki deskripsi serta tanggung jawab yang berbeda. Posisi-posisi tersebut senantiasa berkembang dan berevolusi. Akibatnya, bisa jadi, ada posisi baru yang muncul, berubah, bergabung, atau berkurang. Ketika seseorang melamar pekerjaan, atau berusaha merekrut orang lain untuk bekerja di usaha miliknya, deskripsi dan tanggung jawab ini perlu dicermati.

Secara lebih rinci, beberapa pekerjaan di bidang informatika diberikan pada Tabel 3.6. Selain posisi pekerjaan yang disebutkan ini, masih banyak posisi pekerjaan lain, yang dapat kamu temui di situs-situs penawaran pekerjaan, atau media jaringan sosial tempat para profesional berkumpul. Posisi pekerjaannya dituliskan dalam istilah bahasa Inggris.

Tabel 3.6 Jenis-Jenis Profesi di Bidang Informatika

Posisi Pekerjaan	Deskripsi dan Tanggung Jawab
<i>Chief Information Officer (CIO)/Chief Technology Officer (CTO)</i>	Bertanggung jawab pada semua aspek terkait teknologi informasi di suatu perusahaan.

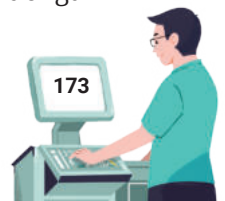


Posisi Pekerjaan	Deskripsi dan Tanggung Jawab
<i>Computer Programmer</i>	Menulis program komputer yang telah didesain.
<i>Computer Scientist</i>	Menciptakan suatu prinsip, metode, atau praktik baru di bidang ilmu komputer.
<i>Hardware Engineer</i>	Menciptakan, mengimplementasikan, dan menguji komponen fisik dari komputer.
<i>Software Engineer</i>	Mengembangkan, merawat, menguji, mengevaluasi, dan meningkatkan kualitas suatu perangkat lunak berdasarkan kebutuhan yang telah diberikan oleh analis sistem atau arsitek.
<i>Software Developer</i>	Menganalisis kebutuhan pengguna dan <i>software requirements</i> untuk menentukan kelayakan desain dalam batasan waktu, risiko, kualitas, dan biaya. Mengembangkan desain sistem, perangkat lunak, prosedur, dan dokumentasi pengujian.
<i>Web Developer</i>	Bertanggung jawab untuk membuat web.
<i>Database Administrator</i>	Mengembangkan dan menerapkan pemantauan, kinerja, kapasitas, dan strategi perluasan database. Menginstal, mengonfirmasi, meng- <i>upgrade</i> , memantau, dan melakukan pemeliharaan pada <i>database</i> .
<i>IT Architect</i>	Mendesain arsitektur teknologi informasi untuk mendukung proses suatu perusahaan.
<i>Network Administrator</i>	Memelihara, mengoperasikan, mengelola, dan mendukung infrastruktur jaringan.



Posisi Pekerjaan	Deskripsi dan Tanggung Jawab
<i>Systems Analyst</i>	Menganalisis dan menerjemahkan tujuan bisnis menjadi produk-produk. Dalam konteks Informatika, produk ini disebut <i>computational artefact</i> , yaitu berupa proses bisnis dan spesifikasi kebutuhan sistem informasinya.
<i>Security Analyst</i>	Menganalisis dan menerjemahkan aspek keamanan dari sistem yang akan dikembangkan untuk mendukung tujuan bisnis.
<i>Information Researcher</i>	Menciptakan suatu prinsip, metode, atau praktik baru untuk mengolah data dan informasi.
<i>Video Game Developer</i>	Mengembangkan perangkat lunak <i>game</i> dan bekerja sama dengan tim profesional kreatif untuk meningkatkan kualitas <i>game</i> .
<i>Health Information Technician</i>	Mengolah data terkait kesehatan menggunakan prinsip dan praktik informatika.
<i>Data Scientist</i>	Merancang, mengembangkan, dan menerapkan model statistik prediktif (biasanya pada <i>big data sets</i>).
<i>Data Engineer</i>	Mengolah data yang dimiliki perusahaan untuk mendapatkan wawasan baru yang berguna bagi perusahaan.
<i>Web Designer</i>	Merancang antarmuka pengguna web untuk memberikan grafik, konten, <i>markup</i> , dan skrip.
<i>Mobile Apps developer</i>	Bertanggung jawab mengembangkan aplikasi dalam bentuk <i>mobile</i> .

Adopsi sebuah teknologi baru di dunia industri akan membutuhkan posisi-posisi baru. Pada saat perusahaan mulai mengadopsi sistem informasi untuk membantu pekerjaan mereka, mereka membuka posisi yang terkait dengan



rekayasa perangkat lunak, misalnya posisi analis sistem, pemrogram, dan perekayasa basis data. Kemudian, isu keamanan siber meningkat sehingga banyak posisi di bidang analis keamanan siber (*cyber security*) menjadi banyak ditawarkan. Saat muncul tuntutan untuk menghasilkan sebuah perangkat lunak yang mudah digunakan dan memberikan kesan elegan bagi pengguna. Posisi yang terkait dengan desain pengalaman pengguna (*user experience*) dan kebergunaan (*usability*) sistem pun ditawarkan. Saat ini, ketika sebuah perangkat lunak diharapkan dapat bekerja dengan cepat walaupun digunakan oleh jumlah pengguna yang sangat besar, banyak usaha pun mencari orang-orang untuk mengisi posisi pengembangan dan operasional IT (*DevOps*) dan arsitek perangkat lunak. Akhirnya, pada saat ini, mulai banyak perusahaan yang merekrut orang-orang dengan kemahiran di analisis data, kecerdasan buatan, atau pembelajaran mesin untuk melakukan otomatisasi pada proses bisnis di perusahaan tersebut. Contoh yang paling lazim ialah untuk mendeteksi penipuan (*fraud*) pada transaksi secara elektronik.

Ilustrasi di atas terjadi karena banyak badan usaha melakukan transformasi digital dengan menerapkan teknologi informasi dalam berbagai proses yang mereka lakukan. Sebagian perusahaan tidak mengembangkan dan memelihara sistem, tetapi menyewa jasa SAS (*Software as a service*) yang beroperasi di *Cloud*. Saat ini, pengembangan maupun penerapan teknologi informasi tersebut bukan lagi suatu hal yang membuat suatu usaha menjadi unggul dalam persaingan (*competitive advantages*), tetapi merupakan suatu syarat agar usaha tersebut dapat bersaing di dunia bisnis (*competitive necessity*). Perubahan ini terjadi dengan sangat cepat, terutama dalam satu dekade terakhir di Indonesia. Oleh karena itu, dalam berkarier di bidang informatika, sangatlah penting untuk selalu meningkatkan kompetensi baik dalam bentuk prinsip maupun praktik untuk menjaga daya saing di kebutuhan dunia yang berubah dengan cepat.

Ketika perusahaan, yang menyadari pentingnya informatika, membuka posisi baru di bidang ini, kebutuhan tenaga kerja bidang informatika pun meningkat. Akan tetapi, menurut Kementerian Komunikasi dan Informatika, jumlah talenta di bidang informatika masih sangat kurang. Bahkan, di tahun 2017, Kemenkominfo menyatakan bahwa Indonesia Darurat Tenaga Programmer. Tidak hanya di Indonesia, ketimpangan tersebut juga terjadi di negara besar seperti Amerika Serikat. Menurut Code.org pada tahun 2020, dari 398.857 lowongan, hanya ada 71.226 lulusan perguruan tinggi yang dapat mengisi posisi tersebut.



Sebagai seorang pelajar di tingkat SMA, persiapan untuk meniti karier di bidang informatika telah dapat dilakukan sejak saat ini. Kamu dapat memanfaatkan media sosial yang sering digunakan untuk bergabung dengan grup-grup komunitas di bidang informatika atau mengikuti berbagai kegiatan atau *meet up* yang dilaksanakan oleh komunitas atau perusahaan di bidang teknologi secara rutin. Bertemu langsung dengan orang-orang yang sudah berkarier di bidang informatika dapat membuat kamu memahami lebih jauh tentang tanggung jawab profesional yang mereka miliki. Selain itu, kamu pun juga dapat melihat tentang tren teknologi yang sedang berkembang, terutama di Indonesia. Membaca laporan-laporan terkini dari World Economic Forum atau lembaga lain juga akan membantu untuk melihat bidang kerja di masa depan yang kamu minati.

Dengan banyaknya posisi pekerjaan baru yang bermunculan, ingatlah bahwa bisa jadi bidang pekerjaan kamu baru akan muncul setelah kamu lulus dari perguruan tinggi kelak. Di sinilah rasa keingintahuan, inisiatif, dan kegigihanmu untuk belajar dan beradaptasi akan membantu kamu menjadi seorang pribadi yang tidak hanya unggul, tetapi juga dapat membawa Indonesia menjadi negara maju dengan kekuatan ekonomi dan inovasi yang besar. Kamu dapat menjadi bagian penting untuk menggapai cita-cita mulia Indonesia Emas pada tahun 2045.

2. Sertifikasi di Bidang Informatika

Kompetensi di bidang informatika tidak hanya dapat diperoleh melalui jalur pendidikan tinggi yang disahkan dalam bentuk ijazah. Belajar informatika dapat dari sumber apa pun. Sangat umum ditemukan seorang pengembang perangkat lunak yang belajar pemrograman secara otodidak. Pada saat ini, di Indonesia juga telah banyak kegiatan seminar, pelatihan, atau *workshop* yang dapat diikuti oleh masyarakat umum termasuk siswa sekolah menengah.

Belajar dan berlatih pun dapat dilakukan dengan basis komunitas. Di berbagai kota dan kabupaten di Indonesia, telah lahir berbagai komunitas di bidang informatika yang terbuka bagi siapa pun yang ingin belajar dan berbagi ilmu. Di internet pun, telah banyak tersedia sumber-sumber belajar yang dapat diakses, baik yang sifatnya gratis maupun berbayar dan memberikan sertifikat. Dengan kata lain, akses terhadap pendidikan informatika sangatlah luas dan memberi kesempatan bagi siapa saja untuk menguasai bidang ini. Tidak sedikit siswa SMP atau SMA yang aktif di komunitas-komunitas dan telah membuat karya inovatif dan cemerlang walaupun dia belum secara formal mendapatkan materi



informatika di jenjang pendidikannya. Hanya dua modal yang kamu perlukan untuk belajar informatika, yaitu rasa ingin tahu dan semangat yang tinggi. Salah seorang ilmuwan terkemuka di bidang informatika, Prof. Jun Rekimoto, bahkan belajar memprogram tanpa komputer dengan menulis kode program di kertas dan membayangkan kode tersebut berjalan.

Walaupun demikian, kompetensi yang diperoleh seseorang perlu diakui. Salah satu bentuk pengakuan tersebut berupa sertifikasi. Sertifikasi ini merupakan suatu dokumen pengakuan yang diperoleh seseorang setelah dianggap memenuhi standar kompetensi yang telah ditetapkan oleh badan, organisasi, atau perusahaan yang memiliki keahlian di bidang tersebut. Biasanya, sertifikasi ini diberikan setelah seseorang lulus dari ujian yang diberikan. Seseorang yang telah memiliki sertifikat akan dapat menggunakan sertifikat tersebut saat bekerja. Hal ini dapat memberi kepercayaan dari tempat bekerja terhadap keahlian yang dimiliki oleh pekerjanya. Berdasarkan kepercayaan tersebut, upah yang diberikan kepada mereka yang memiliki sertifikat sering kali lebih tinggi dibandingkan dengan mereka yang belum bersertifikasi.

Sertifikasi dapat diperoleh dari produsen perangkat keras maupun perangkat lunak, lembaga profesional, asosiasi profesi, atau pemerintah. Sertifikasi dapat terikat pada suatu produk, misal sertifikasi penggunaan perangkat lunak perkantoran atau pada suatu kemampuan, misal sertifikasi sebagai seorang *data scientist*. Lembaga sertifikasi fokus kepada definisi kompetensi dan pelaksanaan ujian, tidak mempedulikan proses belajar sebelumnya. Artinya, seseorang tanpa belajar formal juga dapat mengikuti sertifikasi. Berikut beberapa perusahaan bereputasi yang menyediakan sertifikasi yang diakui secara internasional.

- Microsoft

Menawarkan sertifikasi mulai dari sertifikasi pemakaian produk seperti Microsoft Office, .Net, SQL Server, dan Azure serta sertifikasi terkait keprofesian seperti *data scientist* dan *data engineer*. Microsoft juga memberikan gelar Microsoft Most Valuable Professional bagi para profesional yang unggul dan berpengaruh di bidangnya.

- Google

Menawarkan sertifikasi penggunaan produk seperti Google Suite dan juga sertifikasi terkait keprofesian seperti Android dan Google Cloud Platform. Google memberikan gelar Google Developer Expert bagi para profesional



yang unggul dan berpengaruh di bidangnya. Google memiliki beberapa mitra di Indonesia dan menawarkan berbagai pelatihan gratis sebelum mengambil sertifikasi yang disediakan, misalnya lewat Google Developers Kejar dan beasiswa.

- Oracle

Menawarkan sertifikasi terkait produknya seperti Java, OCJP, dan SCJP.

- CISCO

Menawarkan pelatihan terkait jaringan seperti CCNA.

- Kementerian Komunikasi dan Informatika Republik Indonesia

Menyediakan program pelatihan dalam rangka sertifikasi, misalnya program Pelatihan dan Sertifikasi Talenta Digital yang bekerja sama dengan perusahaan-perusahaan yang memberikan sertifikasi, dan juga bekerja sama dengan perguruan tinggi.

Suatu sertifikasi yang kompleks dapat terdiri atas beberapa sertifikasi yang lebih dasar sehingga membentuk semacam jalur sertifikasi (*certification track*). Misalnya, untuk mendapatkan sertifikasi keahlian menggunakan aplikasi paket perkantoran, kamu harus terlebih dahulu mendapatkan sertifikasi keahlian di setiap aplikasi di paket tersebut, misalnya aplikasi pengolah kata, pengolah lembar kerja, dan presentasi. Kamu dapat mengecek jalur sertifikasi yang ditawarkan di web setiap penyedia sertifikasi.

Bentuk ujian sertifikasi dapat berupa soal-soal teori ataupun praktik langsung. Misalnya, pada sertifikasi teknologi tingkat dasar, ujian sertifikasi dapat berupa soal-soal pilihan ganda untuk menguji pemahaman peserta ujian. Akan tetapi, sertifikasi juga dapat berupa soal-soal praktik untuk menguji keterampilan peserta dalam melaksanakan suatu tugas yang terkait dengan kompetensi yang diujikan. Sertifikasi ada yang dapat dilakukan secara *online* dengan pengawasan tertentu, ada juga yang mengharuskan peserta untuk hadir di suatu lokasi untuk diawasi secara langsung pada saat ujian.

Dua hal yang biasanya menjadi kekhawatiran dalam mengambil sertifikasi ialah tekanan untuk lulus sertifikasi dan masalah biaya. Beberapa sertifikasi yang berbayar tidak akan mengembalikan biaya pendaftaran terlepas kamu lulus atau tidak dari ujian sertifikasi tersebut. Oleh karena itu, yang paling penting ialah menguasai bidang yang akan disertifikasi dengan banyak belajar dan latihan



sebelumnya. Selain itu, ada baiknya kamu mencoba mempelajari bagaimana bentuk ujian sertifikasi yang diberikan serta mencoba latihan sertifikasi yang biasanya diberikan oleh lembaga tersebut. Biasanya, contoh-contoh untuk mencoba disediakan dan dapat diakses bebas biaya

Dari sisi biaya, di Indonesia, cukup banyak akses pelatihan untuk sertifikasi yang dapat diikuti dengan gratis atau berbiaya murah. Selain itu, beberapa perusahaan juga memberi kesempatan berupa beasiswa pelatihan atau sertifikasi yang dapat diambil oleh masyarakat umum, termasuk siswa SMA. Akan tetapi, untuk mendapatkan informasi tersebut, perlu ada tindakan aktif dan inisiatif dari kamu untuk mencari informasi. Cara yang paling mudah ialah dengan rajin mengikuti akun-akun berita, perusahaan, komunitas, atau lembaga terkait di media sosial. Dengan cara demikian, selain menjalin komunikasi dengan temannya di media sosial, kamu juga dapat terpapar dengan informasi dan akses yang bermanfaat bagi masa depan karier dan pengembangan potensi dirimu dengan cara yang menyenangkan.



Aktivitas 10.3-14-U

Menyusun Rencana Masa Depanmu

Pada materi ini, kamu telah memahami bagaimana informatika dapat berperan di berbagai bidang, bagaimana cara memperoleh pengakuan kompetensi di bidang informatika, serta aspek ekonomi dan hukum yang terkait dengan kehidupan sehari-hari.

Langkah-Langkah

1. Dari materi tersebut, apakah kamu telah memahami alasan perlunya memiliki pengetahuan di bidang informatika, walaupun cita-cita kamu mungkin berada di bidang lain?
2. Setelah mengetahui tentang program studi di bidang informatika, apakah kamu tertarik untuk menempuh studi lanjut di informatika? Jika ya, di bidang apa?
3. Setelah mengetahui tentang sertifikasi dan karier di bidang informatika, kompetensi informatika apa yang ingin kamu kuasai? Sertifikasi apa yang akan kamu ambil dalam waktu dekat?
4. Setelah mengetahui aspek dan dampak ekonomi dari informatika, apakah kamu tertarik untuk mengembangkan bisnis yang memanfaatkan informatika?



5. Mengetahui aspek hukum dan hak kekayaan intelektual di bidang informatika, apakah kamu akan menjadi makin menghargai karya-karya di bidang informatika? Apakah kamu akan mulai menggunakan perangkat lunak, buku, musik, atau *file* digital lain dengan lisensi yang benar, dengan menghargai hak para penciptanya?

Tugas Individu

Pilih Program Studimu di Perguruan Tinggi

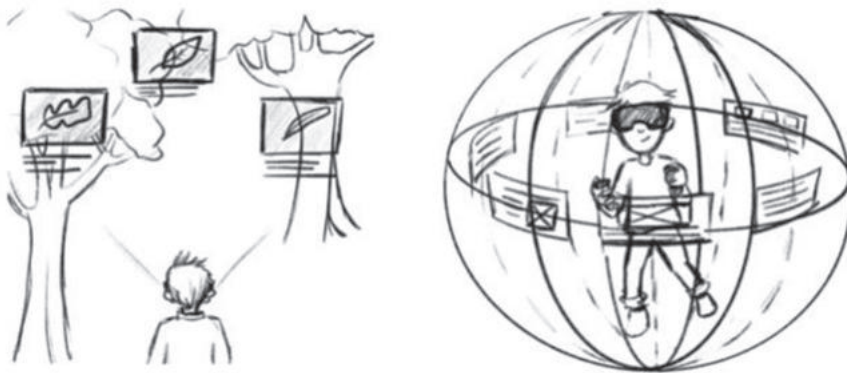
Ayo, pikirkan bidang studi apa yang kamu cita-citakan. Untuk setiap poin di bawah ini, pikirkanlah dengan baik berdasarkan paparan di atas. Kamu juga dapat melakukan pencarian di internet atau membaca buku di perpustakaan untuk menjadi bahan berpikir. Kemudian, tuliskan hasil pemikiranmu pada buku catatan.

1. Apakah kamu berminat untuk studi lanjut di bidang informatika? Mengapa?
2. Jika berminat, program studi apa yang akan kamu pilih? Jelaskan alasannya.
3. Apabila tidak berminat, menurutmu adakah peran dari informatika dalam bidang yang kamu minat? Jelaskan seperti apa peran informatika menurut pemahamanmu.

3. Informatika untuk Masa Depan Indonesia

Kreativitas dan inovasi berawal dari berpikir dan berimajinasi (Gambar 3.22). Banyak penulis dan sastrawan yang berani mengungkapkan bayangan mereka akan masa depan. Banyak yang membayangkan seperti apa dunia di abad ke-21, bahkan lebih jauh lagi ketika manusia sudah melintasi galaksi. Imajinasi itu tidak lagi menjadi sekadar hiburan, tetapi akhirnya menjadi sumber inspirasi bagi para inovator dan penemu yang perlahan-lahan mengubah dunia dengan segala teknologi mutakhir yang mereka ciptakan. Nah, kini, tibalah saatnya kamu juga mulai berimajinasi dan membayangkan seperti apa negeri kita ini, NKRI – Negara Kesatuan Republik Indonesia, di masa depan.





Gambar 3.22 Inovasi-inovasi unggul di bidang informatika banyak bermula dari coretan-coretan kreatif di buku catatan atau jurnal seseorang.

Sumber: Dokumen Kemdikbudristek, 2021

Republik Indonesia, sebagai sebuah negara berdaulat, harus terus tumbuh dan memantapkan posisinya di dunia internasional untuk mewujudkan dan mempertahankan suatu negara yang adil dan makmur. Perserikatan Bangsa-Bangsa (PBB) telah mendeklarasikan suatu konsep bernama tujuan pembangunan berkelanjutan (*sustainable development goals*) atau SDG (Gambar 3.23) yang berisi tujuan yang harus dicapai pada tahun 2030 agar suatu bangsa dapat menjadi bangsa yang maju secara berkelanjutan.



Gambar 3.23 Tujuan Pembangunan Berkelanjutan

Sumber: Perserikatan Bangsa-Bangsa/sdgs.un.org



SDG merupakan kelanjutan dari tujuan pembangunan milenium (*millennium development goals*) atau MDG. Akan tetapi, dalam pelaksanaannya, MDG memiliki banyak kendala. Menurut Jeffrey Sachs, salah satu penasihat MDG, salah satu kendala dalam terwujudnya MDG ialah data yang sulit dikumpulkan dan diolah dengan cepat menjadi informasi yang dapat digunakan oleh para pengambil kebijakan. MDG telah berakhir pada tahun 2015, dan kini telah banyak kemajuan pada bidang komputer yang dapat kita gunakan untuk memastikan SDG dapat dicapai dengan baik dan sesuai dengan target (Gambar 3.24).



(a)



(b)

Gambar 3.24 Ilustrasi gambaran dunia di masa depan, Masyarakat 5.0 (*Society 5.0*) (a) abstraksi, (b) contoh aktivitas

Sumber: Japan Cabinet Office



Selain gagasan SDG, masyarakat saat ini telah mengalami kemajuan dengan adanya gagasan *Society 5.0*, yang didefinisikan sebagai sebuah masyarakat yang berpusat pada manusia, yang menyeimbangkan kemajuan ekonomi dengan resolusi masalah sosial. Masyarakat 5.0 dengan kuat mengintegrasikan ruang siber (alam digital, interkoneksi melalui internet) dengan ruang fisik (alam raya dan tempat-tempat di bumi yang dibagi menjadi 5 benua dengan lokasi geografis masing-masing). *Society 5.0* diusulkan pada “5th Science and Technology Basic Plan” sebagai masyarakat masa depan yang menjadi aspirasi Jepang. Masyarakat 5.0 ini merupakan kelanjutan dari masyarakat pemburu (*Society 1.0*), masyarakat pertanian (*Society 2.0*), masyarakat industri (*Society 3.0*), dan masyarakat informasi (*Society 4.0*).



Aktivitas 10.3-15-U

Tuang Gagasan untuk Indonesia di Masa Depan

Saatnya melahirkan gagasan kreatifmu! Kamu dapat memilih salah satu tujuan pembangunan berkelanjutan, kemudian gambarkanlah gagasanmu tentang bagaimana komputer dapat digunakan di bidang tersebut di Indonesia pada tahun 2045, untuk mewujudkan Masyarakat 5.0. Kamu dapat membuat rancangan dari gagasanmu dalam bentuk apa pun, bahkan dapat berupa coretan sketsa yang ditambah dengan penjelasan atau deskripsi. Kemudian, kamu akan mempresentasikan gagasanmu di depan kelas. Gurumu mungkin akan mengadakan pameran kecil untuk menunjukkan gagasanmu ini ke seluruh sekolah. Jangan lupa untuk memikirkan seperti apa dampak dari gagasanmu, baik positif maupun negatif, kepada masyarakat.



Pilih satu jawaban yang tepat!

1. Manakah yang bukan merupakan salah satu jenis komputer berdasarkan ukurannya?
 - A. *Mini PC*
 - B. *Mini Computer*
 - C. *Micro Computer*
 - D. *Medium Computer*
 - E. *Mainframe*
2. Manakah yang bukan merupakan fungsi utama dari suatu sistem operasi?
 - A. Mengelola *file* yang ada di komputer
 - B. Mengelola penggunaan *memory*
 - C. Menyediakan antarmuka bagi pengguna
 - D. Menyediakan perangkat pengolahan dokumen perkantoran
 - E. Mengelola berbagai perangkat keras yang terhubung
3. Jelaskan apa fungsi dari register pada sebuah prosesor!
4. Manakah yang bukan merupakan fitur yang biasa ada pada aplikasi pengolah kata?
 - A. Fitur *copy+paste* untuk menyalin dan menempelkan teks/objek
 - B. Fitur *cut* untuk memotong sebuah teks/objek
 - C. Fitur untuk menyisipkan tabel
 - D. Fitur untuk menghitung nilai dari nilai-nilai numerik dengan berbagai fungsi matematika/statistika
 - E. Fitur untuk melakukan *mail merge*



5. Manakah yang merupakan pernyataan yang benar mengenai berbagai perangkat lunak perkantoran?
- A. Dokumen yang dibuat pada perangkat lunak perkantoran tidak dapat dikirimkan melalui *email*.
 - B. OLE merupakan fitur yang hanya ada di perangkat lunak pengolahan kata saja.
 - C. Daftar Isi dapat dibuat secara otomatis, tetapi apabila ada perubahan isi, harus kita perbarui secara manual.
 - D. Fitur *copy*, *cut*, dan *paste* dapat diterapkan tidak hanya pada teks, tetapi juga gambar, tabel, dan objek-objek lainnya.
 - E. OLE tidak dapat digunakan untuk menghubungkan data pada satu aplikasi perkantoran ke aplikasi perkantoran lainnya.
6. Di antara jenis-jenis lisensi berikut, manakah yang paling sesuai dengan penggunaan perangkat lunak yang dapat didistribusikan dengan bebas dan dapat diubah kode sumber (*source code*)-nya, tetapi dengan tetap mempertahankan hak cipta pembuatnya?
- A. Lisensi *Freeware*
 - B. Lisensi *Free/open source*
 - C. Lisensi Publik
 - D. Lisensi Komersial
 - E. Lisensi *Shareware*
7. Andi bekerja di sebuah bank dengan tugas utamanya ialah menganalisis kebutuhan pengolahan data dan informasi yang diperlukan oleh bank tersebut, lalu menerjemahkan kebutuhan-kebutuhan tersebut menjadi spesifikasi perangkat lunak yang harus dikembangkan untuk digunakan pada bank tempatnya bekerja. Berdasarkan deskripsi tersebut, jenis profesi apakah yang paling sesuai untuk Andi?
- A. *Computer Scientist*
 - B. *Software Developer*
 - C. *Network Administrator*
 - D. *Data Engineer*
 - E. *System Analyst*



8. Manakah yang bukan merupakan bagian dari tugas seorang *network administrator* pada sebuah perusahaan/instansi?
- A. Mendesain topologi jaringan komputer di perusahaan tersebut.
 - B. Melakukan konfigurasi sistem pada jaringan komputer di perusahaan tersebut.
 - C. Melakukan pengawasan/monitoring terhadap keberlanjutan layanan jaringan komputer.
 - D. Melakukan deteksi dan pencegahan terhadap serangan-serangan keamanan informasi pada perusahaan tersebut.
 - E. Mendesain perangkat keras yang diperlukan untuk menunjang kelancaran jaringan komputer di perusahaan tersebut.



Pengayaan

Jika tertarik dengan materi ini dan ingin mempelajari lebih lanjut, kamu dapat mengakses ke link berikut ini.

1. Khan Academy menyediakan pelajaran mengenai Computers and Internet secara gratis di: <https://buku.kemdikbud.go.id/s/ezyh2g>
2. Berbagai saluran di Youtube membahas mengenai aspek-aspek dalam sistem komputer, misalnya: Computerphile (<https://buku.kemdikbud.go.id/s/3dfuas>), CrashCourse Computer (<https://buku.kemdikbud.go.id/s/w7zk26>) dan lain-lain
3. Terdapat beberapa web yang menyediakan simulator/permainan yang dirancang untuk mempelajari cara kerja prosesor dan sistem digital, misalnya: SimpleCPU (<https://buku.kemdikbud.go.id/s/1ufu6v>), Logisim (<https://buku.kemdikbud.go.id/s/yyrwc7>) dan lain-lain.

Untuk mempelajari lebih lanjut mengenai berbagai perangkat lunak perkantoran, terutama yang dibuat oleh Microsoft, terdapat berbagai sumber yang dapat digunakan, misalnya:



1. Laman layanan Office dari Microsoft (<https://buku.kemdikbud.go.id/s/q0ijox>)
2. Tutorial dari GCF Global (<https://buku.kemdikbud.go.id/s/mv1qzg>)
3. Tutorial dari berbagai saluran di Youtube, seperti TeachersTech (<https://buku.kemdikbud.go.id/s/60xwzf>)

Untuk mempelajari lebih lanjut mengenai berbagai lisensi perangkat lunak, kamu dapat membaca referensi dari beberapa sumber, misalnya:

1. Github (<https://buku.kemdikbud.go.id/s/iedbra>)
2. Free Software Foundation (<https://buku.kemdikbud.go.id/s/zz8vgf>)
3. Open Source Initiatives (<https://opensource.org/licenses>)



Refleksi

Pada bab ini, kamu telah mempelajari berbagai aspek dari teknologi digital komputer, baik dari sisi perangkat keras maupun perangkat lunak, serta berbagai dampak dan keterkaitannya dengan berbagai aspek sosial, ekonomi, dan hukum dari keberadaan teknologi digital ini. Setelah menyelesaikan materi pada bab ini, apa saja hal-hal yang dapat kamu simpulkan? Apakah ada hal-hal yang menarik perhatian kamu mengenai cara kerja komputer yang belum kamu pahami? Apakah pengetahuan yang kamu dapatkan pada bab ini mengenai cara kerja perangkat keras dan perangkat lunak komputer dapat kamu terapkan dalam kehidupan kamu sehari-hari? Apakah dengan pemahaman kamu yang lebih dalam mengenai sistem komputer membuat kamu menjadi (makin) tertarik untuk menekuni bidang komputer dan suatu saat untuk meniti karier di bidang ini?



Keamanan dan Manajemen Informasi

“
Bagaimana mengamankan informasi penting yang kamu miliki?
”





Tujuan Pembelajaran

Setelah belajar bab ini, kamu akan memahami konsep dan penerapan keamanan dan manajemen informasi yang kamu miliki di dunia digital dan internet.

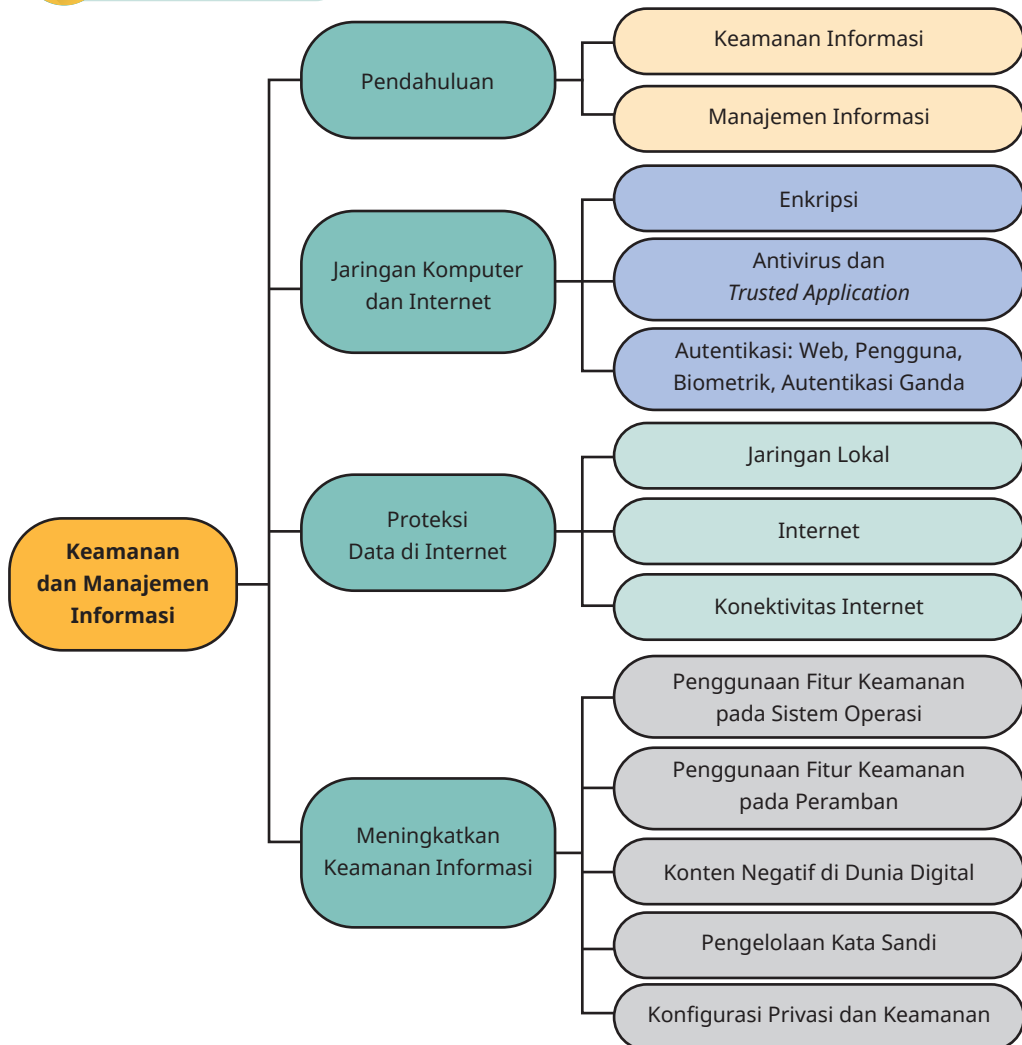


Kata Kunci

Konektivitas, keamanan informasi, manajemen informasi, konten negatif, kata sandi, manajer kata sandi, autentikasi, konfigurasi keamanan.



Peta Konsep



Saat ini, teknologi informasi telah menjadi bagian integral dari kehidupan sehari-hari kita. Semua aktivitas, mulai dari komunikasi hingga transaksi keuangan di dunia nyata dapat dilakukan secara maya melalui perangkat digital. Namun, dengan meningkatnya kebergantungan pada dunia digital, risiko keamanan data juga makin meningkat. Data pribadi kita, seperti informasi identitas, riwayat finansial, dan data sensitif lainnya, menjadi target utama bagi para pelaku kejahatan dunia maya. Oleh karena itu, penting bagi kita semua untuk memahami dan mengimplementasikan praktik keamanan informasi yang efektif.

Di jenjang SMP, kamu telah mengeksplorasi dasar-dasar dunia keamanan data dan informasi, ancaman terhadap keamanan data dan menjaga keamanan diri dari ancaman kejahatan digital. Setelah belajar eksplorasi dasar-dasar dunia keamanan data dan informasi, jawablah beberapa pertanyaan berikut.

1. Apa jenis-jenis ancaman terhadap keamanan data diri kamu di dunia digital?
2. Bagaimana menjaga keamanan data dan informasi diri dari ancaman kejahatan digital?
3. Bagaimana menjaga perangkat kamu dari ancaman kejahatan digital?
4. Bagaimana membuat kata sandi yang kuat?

A. Pendahuluan

Penggunaan internet dan komputer/ponsel pintar yang makin meluas telah banyak menggantikan beberapa kegiatan fisik manusia menjadi kegiatan daring, seperti berkomunikasi, berdagang, bertransaksi perbankan, bersekolah, dll. Ketika banyak pekerjaan dilakukan secara daring, data dan informasi pribadi dan sensitif mudah tersebar di jaringan internet. Data tersebut di antaranya ialah data pribadi yang disimpan di ponsel seperti biodata, nomor kontak teman. Data itu termasuk juga data yang disimpan di penyimpanan awan oleh aplikasi yang kamu gunakan seperti media sosial, e-commerce, dll. Termasuk di dalamnya di instansi pemerintah maupun di institusi bisnis (bank, asuransi, dll).

Hal ini menimbulkan kerawanan pencurian informasi di dunia maya yang berlanjut ke dunia nyata. Saat itulah, kamu harus memahami berbagai aspek keamanan data dan informasi untuk menjaga informasi pribadi kamu agar tetap aman dan terhindar dari kejahatan di internet.



1. Keamanan Informasi

Apa itu keamanan informasi? Keamanan informasi terdiri atas dua kata, yaitu keamanan dan informasi. Keamanan secara fisik dapat dianalogikan dengan perlindungan sebuah gedung, seseorang, organisasi atau negara dari ancaman kejahatan. Keamanan dapat dilakukan melalui objek fisik seperti pagar, dinding dan kunci. Keamanan juga dapat dilakukan oleh orang, proses, pengawasan, otorisasi seperti yang kamu temukan di area tertentu seperti: objek penting negara, bandara, dll.

Informasi ialah hasil pengolahan data. Data sendiri berasal dari kata dalam bahasa Latin, yaitu *datum* yang artinya fakta, keterangan yang benar dan nyata yang dapat diobservasi dan dikumpulkan dari sumber data. Data dapat dijadikan dasar kajian (analisis atau kesimpulan).

Dalam bidang Informatika, data disimpan dalam bentuk yang dapat diproses oleh komputer, seperti representasi digital dari teks, angka, gambar, grafis, suara (audio) atau video. Data dapat bersifat kualitatif yang berarti menggambarkan sesuatu atau kuantitatif yang berupa numerik (angka). Data dapat direkam atau dimasukkan, disimpan, dan ditampilkan. Informasi ialah makna yang disampaikan oleh serangkaian representasi yang merupakan hasil pemrosesan data. Informasi telah memiliki makna. Misalnya, data angka: 160, 181. Ketika tidak memiliki satuan ukuran, data tersebut belum memiliki makna. Jika memiliki satuan, misalnya cm, data tersebut menjadi informasi tinggi badan, jarak antara dua barang di rak, dll. Data dan informasi sering dipertukarkan artinya sebenarnya memiliki arti yang berbeda. Informasi memiliki sifat yang akurat, kontekstual, relevan, bertujuan, spesifik, dan dapat dikelola.

Istilah keamanan sering muncul dalam konteks Informatika yang merujuk pada hal yang berkaitan dengan keamanan penggunaan peranti digital, seperti ponsel pintar, PC, atau gawai lainnya yang juga merupakan sumber data. Peranti-peranti tersebut biasanya terhubung dengan internet.

Keamanan informasi ini berkaitan dengan artefak komputasional yang kamu gunakan. Saat ini, jika kamu menggunakan aplikasi *chatting* di ponsel pintar, kamu menggunakan aplikasi yang mungkin memiliki celah keamanan. Jika kamu menggunakan sistem operasi pada ponsel pintar, informasi kamu juga rawan untuk dicuri. Keamanan informasi terkait dengan pengembangan artefak komputasional yang aman, dari proses pengembangannya yang dimulai dari analisis kebutuhan, perancangan, pengkodean, pengujian,



pengoperasian, dan perbaikannya jika ada kesalahan (*bug*). *Bug* ini dapat menjadi celah keamanan informasi.

Dalam beberapa tahun terakhir, muncul istilah *cybersecurity* (keamanan informasi di internet). *Cybersecurity* mencakup beberapa bidang ilmu yang berhubungan dengan aspek manusia, hukum, kebijakan, etika, dan bahkan hubungan antarnegara. Keamanan data dan informasi lebih luas daripada keamanan dunia maya. Namun, pada materi ini, keamanan data dan informasi akan banyak dijelaskan dalam bingkai keamanan dunia maya.

Keamanan informasi di dunia digital merujuk pada prinsip-prinsip yang menjadi dasar bagi langkah-langkah keamanan informasi. Prinsip itu sering disebut dengan singkatan CIA, yang merujuk pada konteks keamanan informasi, yaitu seperti berikut.

a. Confidentiality (Kerahasiaan)

Prinsip ini menekankan perlunya melindungi informasi agar hanya dapat diakses oleh pihak yang berhak. Dengan menjaga kerahasiaan, informasi tidak dapat diakses atau diungkapkan oleh individu atau entitas yang tidak memiliki otorisasi.

b. Integrity (Integritas)

Prinsip integritas berkaitan dengan memastikan bahwa informasi tetap utuh dan tidak mengalami perubahan yang tidak diinginkan. Informasi harus dapat diandalkan dan akurat.

c. Availability (Ketersediaan)

Ketersediaan menekankan pentingnya memastikan bahwa informasi selalu tersedia untuk mereka yang berhak mengaksesnya ketika diperlukan. Ini mencakup langkah-langkah untuk mencegah atau merespons terhadap gangguan layanan atau kehilangan ketersediaan informasi.

Prinsip CIA ini membentuk dasar bagi pengembangan kebijakan keamanan informasi dan panduan tindakan untuk melindungi informasi pada sistem. Penerapan prinsip-prinsip ini membantu organisasi dan individu untuk merancang dan mengimplementasikan langkah-langkah keamanan yang efektif untuk melindungi informasi mereka dari ancaman dan risiko keamanan. Keberhasilan dalam menjaga kerahasiaan, integritas, dan ketersediaan informasi akan membantu membangun kepercayaan dalam pengelolaan informasi di lingkungan digital.



2. Manajemen Informasi

Manajemen informasi di dunia maya, atau yang sering disebut dengan “*cyber information management*,” mencakup proses pengumpulan, penyimpanan, pengelolaan, analisis, dan distribusi informasi dalam lingkungan *online*. Ini merupakan aspek penting dari berbagai aspek kehidupan modern, termasuk bisnis, pendidikan, pemerintahan, penelitian, dan hiburan.

Komponen kunci dalam manajemen informasi di dunia maya ialah pengumpulan informasi, penyimpanan, pemrosesan, dan analisis data.

a. Pengumpulan Informasi

Ini langkah pertama dalam proses manajemen informasi. Informasi dapat dikumpulkan dari berbagai sumber *online*, seperti situs web, basis data, media sosial, dan sumber-sumber lainnya.

b. Penyimpanan

Setelah dikumpulkan, informasi itu perlu disimpan dan diproses. Ini dapat melibatkan penggunaan sistem penyimpanan data *online*, basis data, dan teknologi *cloud* untuk memastikan keamanan dan ketersediaan informasi.

c. Pengelolaan Informasi

Manajemen informasi mencakup penyusunan dan pengelolaan data sehingga dapat diakses dengan mudah, diorganisir dengan baik, dan terlindungi dari ancaman keamanan.

d. Analisis Data

Data yang dikumpulkan di dunia maya sering kali sangat besar, dan analisis data memungkinkan organisasi untuk mendapatkan wawasan (*insight*) berharga dari informasi ini. Teknik seperti analisis data besar (*big data*) dan kecerdasan buatan (AI) digunakan untuk menggali wawasan dari data tersebut.

Manajemen informasi di dunia maya memainkan peran sentral dalam kesuksesan organisasi modern dan berkontribusi pada efisiensi, produktivitas, dan keamanan informasi. Penerapan praktik terbaik dalam manajemen informasi merupakan kunci untuk mencapai tujuan tersebut.

Platform awal dari keamanan dan manajemen informasi di dunia digital ialah adanya jaringan komputer dan internet. Jaringan komputer dan internet secara revolusioner telah mengubah cara berkomunikasi kita sekarang ini dan membuat data dan informasi dapat dengan mudah diakses.

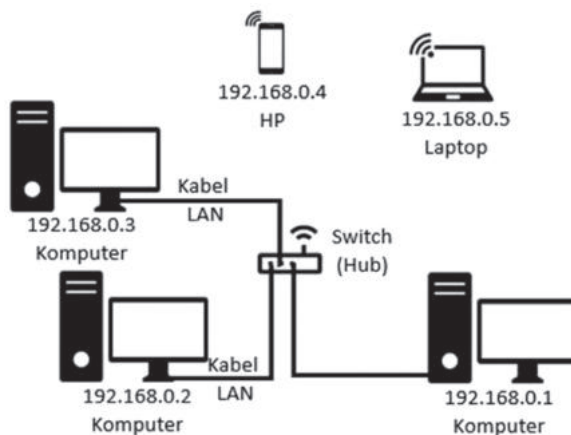


B. Jaringan Komputer dan Internet

Jaringan komputer ialah dua atau lebih perangkat komputer yang tersambung atau terjaring dengan menggunakan sebuah sistem komunikasi standar, yaitu *Transmission Control Protocol/Internet Protocol* (TCP/IP). Jaringan komputer menghubungkan komputer dalam jaringan lokal atau jaringan yang lebih luas yang disebut internet.

1. Jaringan Lokal

Jaringan lokal ialah jaringan komputer berkabel maupun nirkabel yang menghubungkan komputer dengan perangkat lainnya dalam area terbatas seperti tempat tinggal, sekolah, laboratorium, kampus universitas, atau gedung kantor. Perangkat atau komputer yang ada di jaringan lokal hanya dapat diakses oleh perangkat lain yang berada pada jaringan yang sama. Setiap perangkat atau komputer yang terhubung dalam jaringan lokal akan memiliki ID unik yang berbeda satu sama lainnya dan disebut Alamat IP (alamat IP), misalnya 192.168.0.1. Dalam implementasinya, biasanya, satu perangkat akan disebut *server*, sedangkan perangkat lainnya disebut *client*.



Gambar 4.1 Contoh Jaringan Lokal

Sumber: Dokumen Kemdikbudristek (2021)

Gambar 4.1 menunjukkan contoh jaringan lokal yang menghubungkan lima buah perangkat, yaitu tiga buah komputer, satu buah HP, dan satu buah laptop. Jika kamu perhatikan, setiap perangkat memiliki alamat IP yang berbeda. Karena alamat IP ini hanya berlaku pada jaringan lokal, dapat juga disebut IP *private*. Apa yang akan terjadi jika ada dua buah perangkat memiliki alamat IP yang sama?



Apa sebenarnya arti dan fungsi dari alamat IP? Alamat IP merupakan identitas sebuah komputer dalam jaringan komputer. Alamat IP berfungsi sebagai alamat pengiriman data dari satu perangkat ke perangkat lainnya. Mungkin kamu dapat menganalogikan alamat IP seperti alamat rumah. Saat akan mengirim paket, tentunya akan lebih mudah kalau sudah mengetahui alamat pastinya. Sama halnya dengan ketika kamu akan mengirim data dari satu perangkat ke perangkat lain. Kamu harus tahu data itu akan dikirim ke komputer dengan alamat IP yang mana. Alamat IP sendiri biasanya terdiri atas 32 binary digit (bit) yang dipisah menjadi 4 bagian sehingga setiap bagian akan terdiri atas 8 bit. Nah, 8 bit ini merupakan bilangan biner yang diterjemahkan ke dalam bilangan desimal. Tentu kamu sudah belajar tentang bilangan biner. Berikut ialah contoh dari alamat IP.

<i>IP addres dalam bilangan biner</i>	<i>IP address</i>
10101100.11011001.00001010.00001110	127.217.10.14

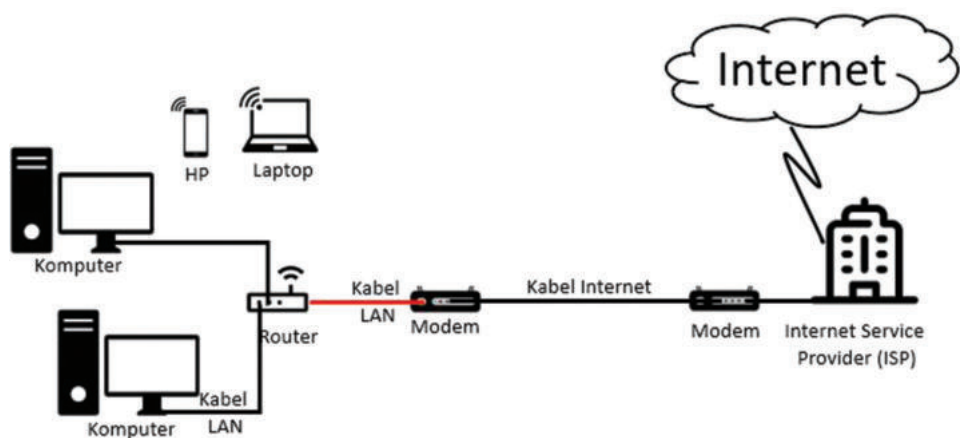
2. Internet

Berbeda dengan jaringan lokal yang hanya menghubungkan perangkat dengan jangkauan yang terbatas pada area tertentu saja, jaringan internet memiliki koneksi dengan cakupan yang lebih luas yang dapat menghubungkan perangkat di seluruh dunia. Jadi, misalnya kamu berada di Jakarta, dengan jaringan internet, kamu dapat mengakses perangkat atau informasi yang ada di Kalimantan bahkan di negara lain. Luar biasa, kan?

Internet sendiri merupakan kepanjangan dari *interconnected-network*, yaitu sebuah jaringan komputer yang menghubungkan banyak perangkat di seluruh dunia. Jaringan internet ini memungkinkan adanya pertukaran data dalam bentuk paket (*packet switching communication protocol*) untuk melayani miliaran pengguna di seluruh dunia. Secara konsep, jaringan internet juga disebut jaringan area luas (*Wide Area Network*).

Pada awalnya, di tahun 1960-an, internet hanyalah sebagai proyek penelitian yang didanai oleh Departemen Pertahanan Amerika Serikat pada tahun 1969, melalui proyek lembaga ARPA yang mengembangkan jaringan yang dinamakan ARPANET (Advanced Research Project Agency Network), kemudian berkembang menjadi infrastruktur publik pada 1980-an dengan dukungan dari banyak universitas negeri dan perusahaan swasta.





Gambar 4.2 Jaringan lokal dapat diubah menjadi jaringan internet.

Sumber: Dokumen Kemdikbudristek (2021)

Untuk dapat mengakses jaringan internet, biasanya, perangkat yang kamu miliki juga harus terhubung dengan salah satu *Internet Service Provider* (ISP) dengan menggunakan mekanisme koneksi internet tertentu. Jika jaringan lokal yang kamu miliki terhubung dengan ISP baik dengan kabel maupun nirkabel, kamu dapat mengakses internet dari jaringan di rumah atau sekolah kamu. Jika jaringan lokal terhubung ke ISP, kamu sudah terhubung dengan jaringan internet. Hal ini ditunjukkan dengan garis merah (yang diasumsikan sebagai kabel) pada Gambar 4.2. Sama seperti jaringan lokal, setiap perangkat pada jaringan internet juga harus memiliki alamat IP yang berbeda-beda. Karena alamat IP ini berlaku secara global, dapat juga disebut sebagai *IP public*. Akan tetapi, tidak semua perangkat lokal yang terhubung dengan internet memiliki *IP public*. Biasanya, *IP public* hanya digunakan pada perangkat utama yang terhubung dengan jaringan internet secara langsung, dalam hal ini ada perangkat modem yang terhubung dengan ISP.

3. Konektivitas Internet

Koneksi dengan internet pada umumnya dilakukan menggunakan kabel internet yang dapat berupa kabel *coaxial*, kabel *fiber optik* maupun kabel *twisted pair*. Meski demikian, kamu juga dapat terhubung ke jaringan internet tanpa menggunakan kabel (*wireless connection*). Baik menggunakan kabel maupun tanpa kabel, keduanya memiliki metode tersendiri bagaimana terkoneksi ke jaringan internet. Berikut ialah beberapa jenis koneksi internet yang memungkinkan perangkat kamu terhubung ke dalamnya.

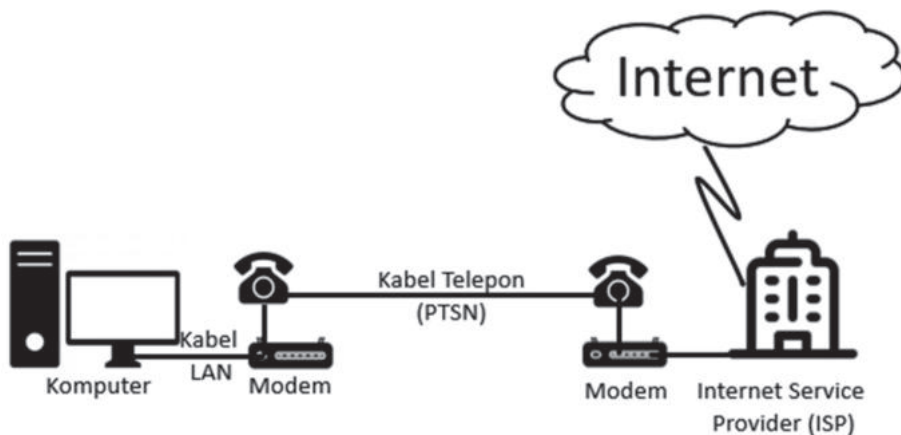


a. Konektivitas Internet pada Jaringan Menggunakan Kabel

Konektivitas internet pada jaringan menggunakan kabel pada umumnya memanfaatkan jaringan kabel telepon maupun televisi yang terhubung dengan ISP. Berikut ialah beberapa konektivitas pada jaringan menggunakan kabel.

1) Dial-Up PSTN (*Public Switched Telephone Network*)

Di awal internet masuk ke Indonesia, teknologi Dial-Up untuk terkoneksi ke internet merupakan metode yang umum digunakan. Untuk terkoneksi dengan internet, Dial-Up memanfaatkan jaringan telepon rumah yang berbentuk kabel. Tiga perangkat yang harus dimiliki untuk koneksi ini ialah komputer, modem, dan sambungan telepon rumah. *Dial-up* ini akan bekerja melalui jalur PSTN (*Public Switched Telephone Network*) hingga dapat terhubung dengan ISP. Kecepatan akses jenis koneksi internet *Dial-Up* hanya 12 hingga 20 Kbps, maksimum hanya 56 Kbps (*kilobyte per second*). Gambar 4.3 menunjukkan ilustrasi bagaimana konektivitas internet menggunakan cara *Dial-Up*.



Gambar 4.3 Diagram konektivitas internet jaringan menggunakan kabel melalui Dial-Up dan ADSL.

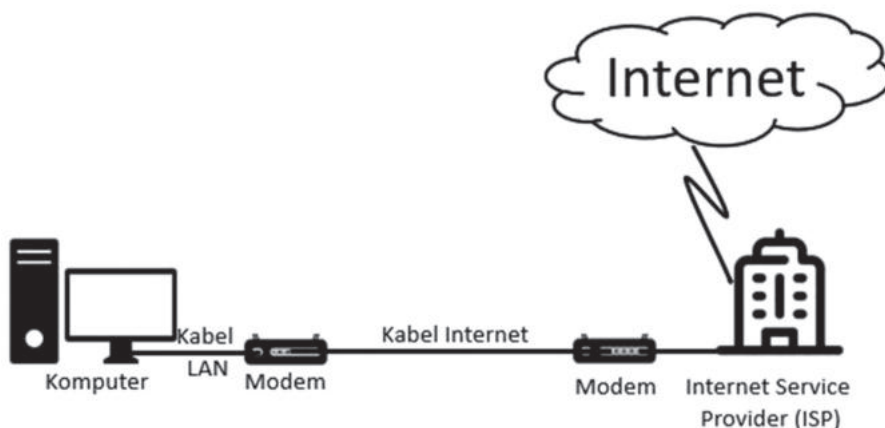
Sumber: Dokumen Kemdikbudristek (2021)

2) LAN (*Local Area Network*)

LAN termasuk koneksi internet yang banyak dikenal saat ini. Sistem kerjanya ialah menggunakan satu komputer sebagai server yang terhubung dengan internet menggunakan kabel telepon atau antenna melalui ISP. Kemudian, komputer lainnya hanya perlu terkoneksi dengan server untuk dapat mengakses internet dengan memakai kartu LAN



(LAN Card) dan kabel koaksial (UTP). Jaringan koneksi internet ini hanya mencakup wilayah yang sangat kecil. Ilustrasi jaringan internet dengan kabel LAN ditunjukkan pada Gambar 4.4.



Gambar 4.4 Konektivitas Jaringan LAN

Sumber: Dokumen Kemdikbudristek (2021)

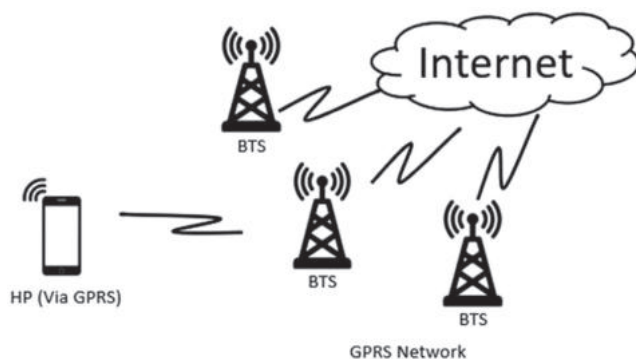
b. Konektivitas Internet pada Jaringan Nirkabel

Konektivitas internet pada jaringan nirkabel merupakan jaringan internet yang tidak menggunakan kabel untuk menghubungkan satu perangkat dengan perangkat lain. Jaringan nirkabel ini sering dipakai untuk jaringan komputer baik pada jarak yang dekat (beberapa meter, memakai alat/pemancar *bluetooth*) maupun pada jarak jauh (lewat satelit). Jaringan nirkabel biasanya menghubungkan satu sistem komputer dengan sistem yang lain dengan menggunakan beberapa macam media transmisi tanpa kabel, seperti: gelombang radio, gelombang mikro, maupun cahaya infra merah. Berikut ialah beberapa konektivitas pada jaringan nirkabel.

1) GPRS (*General Packet Radio Service*)

Jenis koneksi internet GPRS menggunakan gelombang radio untuk komunikasi data dan suara. GPRS ini mempunyai kemampuan untuk mengirimkan data dan suara pada alat komunikasi bergerak. Sistem GPRS ini dapat dipakai untuk transfer data, berbentuk paket data yang terkait dengan pengiriman surel (surat elektronik) hingga berselancar di dunia maya. Layanan GPRS ini dipasang pada jenis ponsel dengan tipe GSM (*Global Systems for Mobile Communications*). Ilustrasi jaringan internet nirkabel GPRS ditunjukkan pada Gambar 4.5.



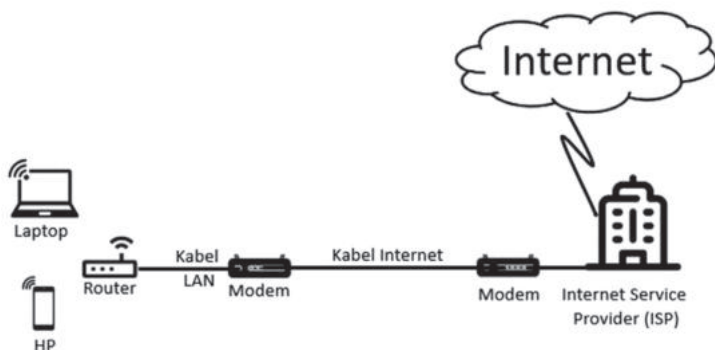


Gambar 4.5 Ilustrasi konektivitas jaringan nirkabel via GPRS

Sumber: Dokumen Kemdikbudristek (2021)

2) Wifi

Wifi ialah singkatan dari *wireless fidelity* yang merupakan sistem standar yang digunakan untuk terkoneksi dengan internet tanpa menggunakan kabel. Teknologi ini memakai frekuensi tinggi, berada pada spektrum 2,4 GHz. Salah satu keunggulan jenis koneksi internet wifi ialah praktis karena tidak perlu memasang kabel jaringan. Namun, koneksi wifi memiliki jangkauan terbatas. Ilustrasi jaringan internet nirkabel wifi ditunjukkan pada Gambar 4.6.

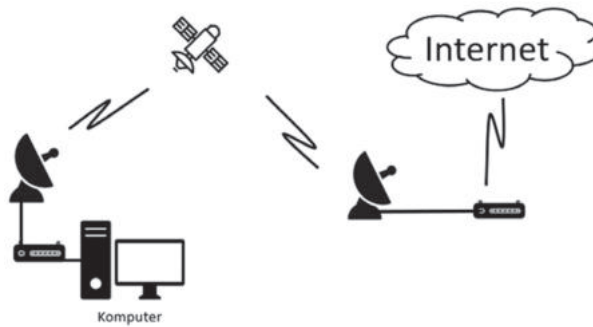


Gambar 4.6 Ilustrasi konektivitas jaringan nirkabel Wifi.

Sumber: Dokumen Kemdikbudristek (2021)

3) Akses Satelit

Jenis koneksi internet ini merupakan layanan internet yang memakai antenna parabola sehingga kecepatan akses cukup tinggi. Namun, biasanya, jenis koneksi internet ini memang mahal terutama di Indonesia, meski sepadan dengan kecepatan aksesnya. Ilustrasi jaringan internet nirkabel satelit ditunjukkan pada Gambar 4.7.



Gambar 4.7 Ilustrasi konektivitas jaringan nirkabel melalui satelit.

Sumber: Dokumen Kemdikbudristek (2021)

c. Komunikasi Data dengan Ponsel

Ponsel ialah salah satu perangkat yang dapat terkoneksi dengan internet dan paling banyak digunakan saat ini. Saat ini, penggunaan ponsel dengan internet menyentuh 98% pengguna.

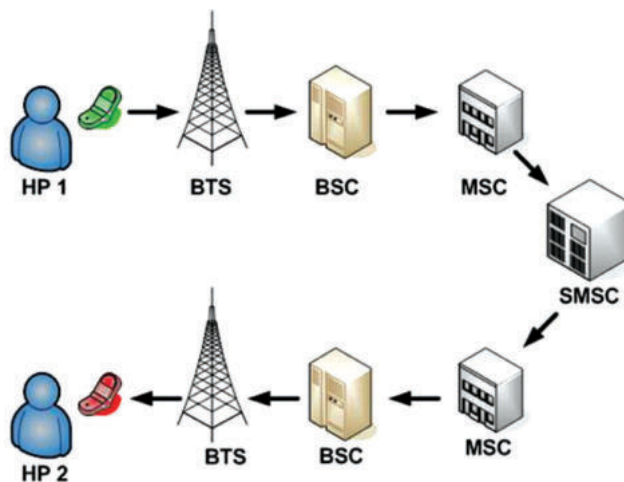
1) Jaringan Komunikasi pada Ponsel

Komunikasi data merupakan sebuah mekanisme pengiriman dan penerimaan data dari satu perangkat ke perangkat lain. Ketika kamu menggunakan ponsel, baik untuk keperluan mengirim pesan atau SMS (*short message system*), panggilan suara, maupun transfer dokumen, di dalamnya ada mekanisme komunikasi data yang terjadi antara ponsel pengirim dan ponsel penerima. Komunikasi data antarponsel dilakukan melalui beberapa perangkat seperti BTS (*Base Transceiver Station*), dan perangkat lain. Namun, karena makin berkembangnya teknologi internet, saat ini, komunikasi data menggunakan ponsel dapat melalui jaringan internet antara lain seperti berikut.

- a. GPRS (*General Packet Radio Service*) merupakan jaringan internet pada ponsel yang memiliki kecepatan antara 35-171 Kbps. Dengan kecepatan ini, biasanya, hanya cukup untuk mengirim pesan yang tidak terlalu panjang.
- b. EDGE (*Enhanced Data rate for GSM Evolution*) merupakan jaringan internet yang memiliki kecepatan lebih baik dibandingkan dengan GPRS karena memiliki kecepatan antara 120-384 Kbps. 3G merupakan jaringan internet GPRS versi 3 yang menggunakan protokol transfer data United Mobile Telecommunication Technology sehingga menghasilkan kecepatan antara 384 Kbps – 2 Mbps.



- c. HSPA (*High Speed Packet Access*) yang dikenal dengan nama 3.5G dan memiliki kecepatan 600 Kbps – 10 Mbps. HSDPA (*High Speed Downlink Packet Access*) atau dikenal dengan nama H+ pada layar HP memiliki kecepatan yang stabil pada kisaran 7.2 Mbps.
- d. 4G/LTE (*Long Term Evolution*) merupakan generasi keempat dari GPRS yang memiliki kecepatan hingga 100 Mbps. Saat ini, hampir sebagian besar ponsel di pasaran sudah mendukung jaringan 4G/LTE.
- e. Pada tahun 2023, beberapa perusahaan telekomunikasi di Indonesia telah menerapkan jaringan 5G dengan kecepatan maksimum mencapai 10 Gbps. Gambar 4.8 menunjukkan ilustrasi bagaimana cara kerja komunikasi data menggunakan ponsel saat mengirimkan SMS.



Gambar 4.8 Salah satu alur kerja proses komunikasi data SMS melalui ponsel.

Sumber: Dokumen Kemdikbudristek (2021)

2) Komunikasi Data dengan Ponsel

Ponsel/*handphone*/HP merupakan sebuah perangkat yang bekerja layaknya sebuah komputer karena dapat terhubung ke dalam sebuah jaringan lokal maupun jaringan internet. Oleh karena itu, sangat memungkinkan bagi kamu untuk saling melakukan komunikasi data antara satu ponsel dan ponsel lainnya. Ada beberapa jenis koneksi jaringan pada perangkat ponsel yang digunakan untuk komunikasi data. Sebagian besar koneksi tersebut menggunakan media tanpa kabel antara lain koneksi dengan kabel data, koneksi dengan *bluetooth*, dan koneksi dengan wifi.



a) Koneksi dengan Kabel

Koneksi dengan kabel pada ponsel biasanya dilakukan dengan menggunakan *data cable* (kabel data). Penggunaannya lebih banyak untuk menghubungkan ponsel dengan perangkat komputer atau perangkat lainnya yang memiliki *port* USB. Karena menggunakan kabel, jangkauan koneksi ini sangat terbatas sesuai dengan panjang kabel data yang kamu miliki.



Gambar 4.9 Contoh kabel data USB-A ke USB-C

Sumber: Fructibus/Wikimedia Commons

b) Koneksi dengan *Bluetooth*

Bluetooth ialah komunikasi data tanpa kabel yang menggunakan gelombang elektromagnetik yang beroperasi pada frekuensi 2.45. *Bluetooth* memungkinkan pengguna perangkat seperti ponsel, printer, *mouse*, *keyboard*, dan perangkat lainnya dapat terkoneksi tanpa kabel dengan jangkauan sekitar 1 meter.



Gambar 4.10 Contoh ilustrasi komunikasi data menggunakan *bluetooth*.

Sumber: 200degrees/Pixabay



c) Koneksi dengan Wifi

Koneksi data dengan jaringan tanpa kabel menggunakan teknologi wifi merupakan jenis koneksi yang banyak digemari. Alasan utamanya ialah karena *bandwidth*-nya yang besar dan hemat energi baterai. Kapasitas *bandwidth* maksimum sangat bergantung pada jarak dan data *rate* dari perangkat *access point*. Kekurangannya ialah jangkauannya yang tidak jauh, maksimum 100 meter, tetapi dengan *bandwidth* yang minimum (data rate 1Mbps).



Gambar 4.11 Contoh ilustrasi komunikasi data menggunakan wifi.

Sumber: Dokumen Kemdikbudristek (2021)

Aktivitas 10.4-01-U
Perancangan Jaringan Komputer

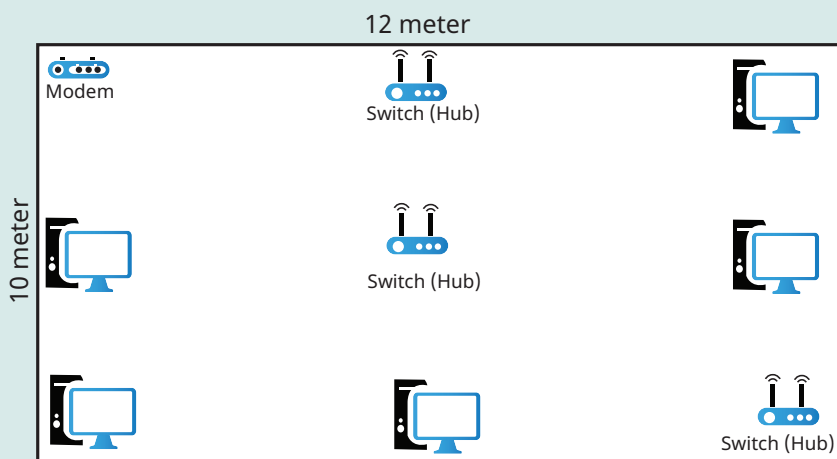
Pada aktivitas ini, kamu akan belajar bernalar kritis dengan merancang konfigurasi sebuah jaringan komputer yang menghubungkan perangkat komputer sehingga dapat berkomunikasi dengan perangkat komputer lainnya.

Deskripsi Kasus

Ani berencana memasang jaringan internet baru pada suatu *Internet Service Provider*. ISP tersebut menyediakan satu buah modem secara gratis yang memiliki IP *public* yang hanya dapat terkoneksi dengan maksimal dua perangkat lainnya. Adapun Ani sendiri memiliki tiga buah *switch/router* yang masing-masing dapat terpasang dengan maksimal tiga perangkat lainnya. Ani menginginkan agar semua perangkat komputer yang ada di rumahnya



terkoneksi dengan internet. Namun, dia juga menginginkan agar biaya pasang yang dikeluarkan semurah mungkin. Biaya pasang ini bergantung pada total panjang kabel yang harus digunakan. Jika diberikan posisi semua perangkat yang dimiliki oleh Ani seperti pada Gambar 4.12, bantulah dia untuk memodelkan konfigurasi jaringan di rumahnya sehingga biaya yang harus dikeluarkan semurah mungkin. Posisi perangkat tidak boleh dipindah-pindah dan semua *switch/router* tidak harus digunakan.



Gambar 4.12 Kasus Perancangan Jaringan

Sumber: Dokumen Kemdikbudristek (2021)

Langkah-Langkah

1. Rancanglah model jaringan komputer pada gambar di atas.
2. Setelah merancang model konfigurasi jaringan, bandingkan dengan rancangan teman kamu. Apakah sama? Mana yang lebih baik?
3. Selanjutnya, buatlah sketsa denah rumahmu. Bayangkan kamu memiliki beberapa komputer yang harus terkoneksi dengan internet. Buatlah model konfigurasi jaringannya.
4. Setelah selesai merancang model konfigurasi jaringan komputer, jangan lupa, kamu harus meminta kepada teknisi jaringan jika ingin memasangnya. Pasti teknisi akan sangat senang karena kamu telah membuatkan rancangan sehingga teknisi hanya perlu memasangnya.
5. Setelah selesai, bandingkanlah hasil pekerjaanmu dengan pekerjaan temanmu.



C. Proteksi Data di Internet

Ketika menggunakan internet, hampir setiap hari kamu selalu mendengar adanya pelanggaran keamanan dan serangan dunia maya. Hal itu mungkin terjadi karena banyak kerentanan yang ada pada sistem kita. Dalam subbab ini, akan dibahas mengenai perkakas yang dapat melindungi data dan informasi kamu di internet sehingga lebih aman.

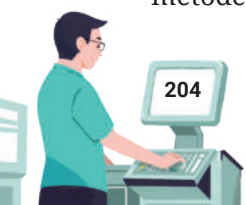
Konsep keamanan dasar pada jaringan komputer dan internet mencakup sejumlah praktik dan prinsip yang dirancang untuk melindungi integritas, kerahasiaan, dan ketersediaan data dalam suatu jaringan. Berikut ialah beberapa konsep dasar keamanan jaringan.

1. *Firewall*: Menggunakan *firewall* untuk memantau dan mengontrol lalu lintas jaringan antara jaringan internal dan eksternal. *Firewall* dapat mengizinkan atau memblokir lalu lintas berdasarkan aturan keamanan tertentu. Misalnya, untuk memblokir situs tertentu dari jaringan internet suatu sekolah.
2. Enkripsi: Menggunakan teknik enkripsi untuk melindungi data yang dikirim melalui jaringan. Enkripsi membantu mencegah akses tidak sah terhadap informasi yang sensitif.
3. Proteksi terhadap Virus dan *Malware*: Menggunakan perangkat lunak antivirus dan *antimalware/spyware* untuk melindungi sistem dari ancaman *malware* seperti virus, worm, trojan, dan spyware.
4. Pembaruan Perangkat Lunak: Menjaga perangkat lunak, termasuk sistem operasi, aplikasi, dan perangkat lunak keamanan, tetap diperbarui dengan versi terbaru untuk mengatasi kerentanan keamanan yang ditemukan.
5. Kesadaran Pengguna: Memberikan pelatihan dan informasi kepada pengguna agar mereka dapat mengidentifikasi praktik keamanan yang baik, seperti membuat kata sandi yang kuat, tidak mengklik tautan yang mencurigakan, dan melaporkan aktivitas yang mencurigakan.

Berikut ialah penjelasan lebih rinci dari konsep keamanan informasi pada jaringan komputer dan internet.

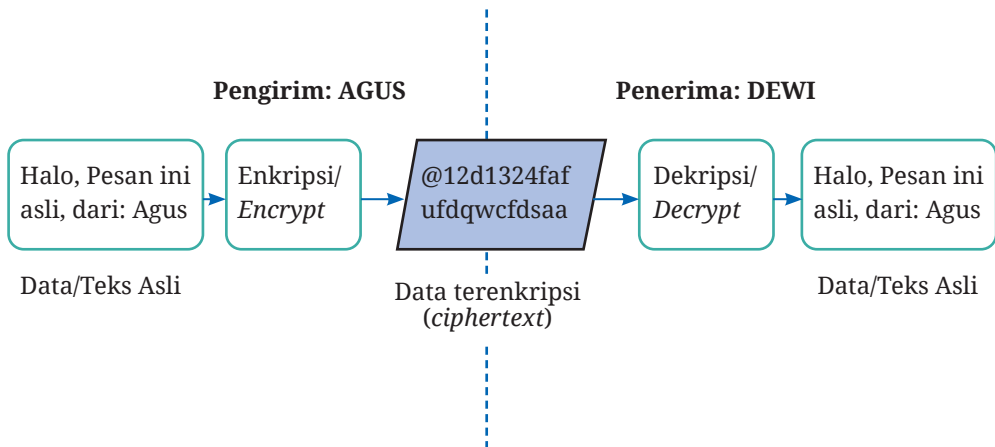
1. Enkripsi

Enkripsi ialah alat keamanan yang sangat berharga untuk pengamanan data pada komunikasi data di jaringan komputer/internet. Enkripsi ialah suatu metode yang mengodekan data sebelum dikirim melalui jaringan komputer.



Data tersebut disandikan sedemikian rupa sehingga tidak dapat dibaca sebelum dikembalikan ke bentuk aslinya (di-decrypt). Dengan enkripsi, data yang menyebar dalam jaringan komputer, atau dalam bentuk lainnya, tidak dapat dibaca tanpa didekripsi. Metode ini membuat data menjadi lebih aman.

Cara enkripsi data digambarkan dengan diagram pada gambar 4.13 berikut.



Gambar 4.13 Enkripsi dan Dekripsi Data

Beberapa perancang internet awal sangat menganjurkan penggunaan enkripsi pada komunikasi data. Pada saat dikembangkan, memang protokol utama komunikasi Internet, TCP/IP, tidak melakukan enkripsi data karena enkripsi membutuhkan sumber daya komputasi yang besar dan mahal. Masalah juga diperparah dengan problem pendistribusian kunci (*key*) dekripsi yang sulit.

Saat ini, teknologi enkripsi sudah sangat mapan, tetapi masih dianggap mahal dan merepotkan. Teknologi ini membutuhkan biaya pengembangan dan sumber daya komputasi yang tidak sedikit sehingga pemerintah dan bisnis pun sering tidak menggunakannya bahkan untuk aplikasi yang sangat penting.

Sebagai contoh, institusi militer di Amerika, pada awalnya, tidak mengenkripsi sistem pemberi masukan (*feeder*) video pada sistem *drone*. Celah ini kemudian mampu dimanfaatkan seseorang (musuh) untuk meretasnya dengan menggunakan perangkat lunak seharga 500 ribu rupiah yang tersedia di internet. *Drone* kemudian akan mendapatkan informasi yang salah sehingga tidak dapat melakukan operasi pengintaian yang tepat sasaran. Insiden tersebut menunjukkan contoh enkripsi data yang tidak sempurna. Penggunaan penting berikutnya dari enkripsi ialah untuk kepentingan penyimpanan data dan dokumen. Dalam beberapa kasus



pencurian besar data pribadi konsumen dari perusahaan pengecer, didapatkan bahwa *database* mereka termasuk kata sandi, nomor kartu kredit, dan nomor penting lainnya tersimpan dalam keadaan tanpa enkripsi.

Teknik lain yang banyak digunakan peretas untuk mendapatkan informasi kredensial pengguna ialah menggunakan wifi tidak terenkripsi. Peretas dengan santai duduk di *café* kopi sambil melakukan *scanning* transmisi wifi untuk mencari orang yang terhubung ke jaringan. Jika ada pengguna yang melakukan *login* dengan menggunakan informasi pribadi dan kredensial, data tersebut dapat dicuri. Jadi, berhati-hatilah jika kamu berada di area publik dengan wifi gratis, koneksi dan data kamu dapat berada dalam kondisi rentan. Gunakan koneksi data milik sendiri jika melakukan transaksi *online*.

Saat ini, aplikasi bertukar pesan pada ponsel pintar telah banyak menggunakan teknik enkripsi dalam berkomunikasi. Aplikasi tersebut di antaranya ialah Whatsapp, Telegram, Facebook Messenger, dll. Whatsapp sebagai salah satu aplikasi dapat diatur untuk mendapatkan enkripsi dari ujung ke ujung (*end-to-end encryption*).



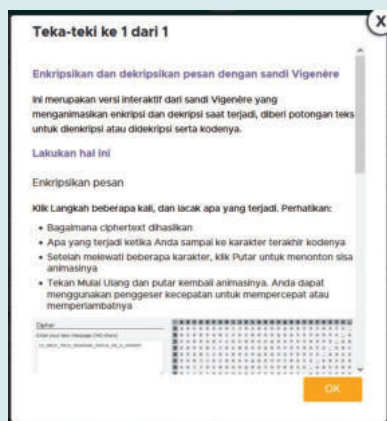
Aktivitas 10.4-02-P

Enkripsi dan Dekripsi dengan Vigenere Cipher

Pada aktivitas ini, kamu belajar bernalar kritis dengan memahami konsep enkripsi dan dekripsi dengan Vigenère cipher yang selanjutnya menerapkan enkripsi dan dekripsi algoritma sendiri.

Langkah-Langkah

1. Kunjungilah sebuah web interaktif berikut: <https://buku.kemdikbud.go.id/s/yml3pl>



Halaman web ini ialah versi interaktif dari Vigenere cipher yang menganimasikan proses terjadinya enkripsi dan dekripsi, dengan diberikan teks yang akan dienkripsi atau didekripsi dan kuncinya.

2. Lakukan enkripsi pesan berikut sesuai dengan petunjuk pada *website*.

PESAN_INI_ASLI

Gunakan Secret Key: **INFORMATIKA**

3. Apa hasil enkripsi dari pesan di atas? Lanjutkan untuk mengenkripsi teks berikut.

No.	Teks Asli	Teks Terenkripsi
1.	KETEMU_DI_LAB	
2.		IKTNSQRTVQKIFEYVLTCX

4. Selanjutnya, lakukan enkripsi data yang sama dengan menggunakan *secret key* yang berbeda, misalnya **KUNCI_X**, dan catat hasilnya.
5. Selanjutnya, pikirkanlah algoritma apa yang lebih kompleks dari algoritma Vigenere cipher sehingga pesan yang kamu kirimkan menjadi lebih aman.

2. Antivirus dan Trusted Application

Pada era digital yang makin kompleks ini, ancaman perangkat lunak berbahaya telah menjadi perhatian utama bagi pengguna komputer di seluruh dunia. Dari serangan virus yang merusak hingga perangkat lunak berbahaya yang meretas data pribadi, perlindungan terhadap sistem dan informasi pribadi menjadi suatu keharusan. Di tengah maraknya ancaman tersebut, antivirus muncul sebagai garda terdepan dalam melindungi perangkat dari serangan berbahaya. Dengan teknologi yang terus berkembang, antivirus tidak hanya sekadar mendeteksi dan menghapus virus, tetapi juga menyediakan lapisan pertahanan yang lebih dalam untuk mencegah berbagai jenis ancaman yang terus berkembang.

Selain itu, di tengah banyaknya aplikasi yang tersedia di *application online store*, dan dunia digital yang penuh dengan ancaman keamanan, kepercayaan dalam aplikasi yang digunakan menjadi sangat penting. Konsep “*trusted application*” atau aplikasi yang terpercaya menjadi landasan dalam memastikan



bahwa perangkat lunak yang dijalankan oleh pengguna dapat dipercaya untuk menjaga keamanan, privasi, dan integritas data. *Trusted application* bukan sekadar aplikasi yang bekerja dengan benar, tetapi juga telah melalui proses validasi dan verifikasi untuk memastikan bahwa tidak ada potensi kerentanan atau bahaya yang tersembunyi.

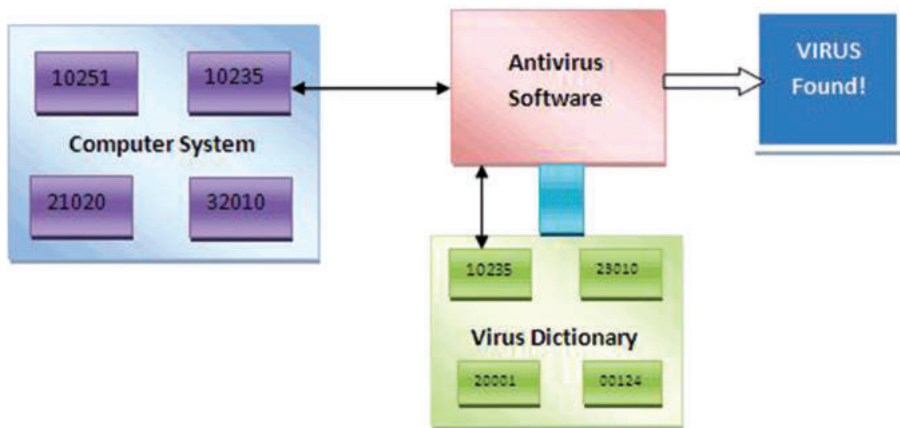
a. Antivirus

Merebaknya *malware* atau virus menyebabkan pengguna merasa terganggu kenyamanannya ketika bekerja dengan komputer dan takut ketika bertransaksi di internet. Beberapa alat dan perangkat lunak telah tersedia untuk membantu pengguna melindungi peranti dan *file* dari gangguan virus. Perangkat lunak itu juga untuk menghindari peranti kita menjadi celah menjadi mata rantai yang lemah dalam rantai sistem keamanan.

Perangkat lunak tersebut di antaranya ialah antivirus atau *anti-malware*. Perangkat lunak antivirus akan mencari virus/*malware* di komputer dengan menggunakan dua cara berikut.

Pertama, jika kamu telah memasang antivirus di komputer, saat kamu memasang peranti baru, seperti kamera, printer atau *drive* USB ke komputer, antivirus akan melakukan pemindaian semua *file* yang berhubungan dengan peranti tersebut. Antivirus akan mencari *signature* (tanda tangan) virus, yaitu urutan karakter unik yang berhubungan dengan virus tersebut, dengan mencocokkannya dengan *signature* virus yang disimpan dalam *dictionary* (kamus) antivirus. Jika cocok, ia akan memberi tahu kamu bahwa ada *file* yang mengandung virus dan akan dilakukan “karantina”. Karantina biasanya dilakukan dengan menempatkannya pada folder khusus sampai kamu memutuskan apakah *file* tersebut akan dibersihkan atau dihapus. Ilustrasi bagaimana kerja antivirus tampak pada Gambar 4.14. Selain itu, kamu dapat mengatur perangkat lunak antivirus untuk memindai *file* di komputer secara berkala untuk mengetahui apakah ada *malware* pada komputer.





Signature based Antivirus

Gambar 4.14 Cara Kerja Antivirus

Kedua, memantau sistem komputer untuk aktivitas yang ditetapkan sebagai “kegiatan virus”. Beberapa kegiatan virus itu antara lain mengubah *file* sistem yang biasanya tidak akan diubah, memodifikasi bagian *file* memori komputer di luar area program yang diizinkan, atau memodifikasi beberapa program secara bersamaan. Ketika perangkat lunak *anti-malware* mendeteksi aktivitas semacam itu, perangkat lunak itu akan dinon-aktifkan dan pengguna akan diperingatkan.

Namun, seiring waktu, peretas menemukan cara untuk menghindari perangkat lunak *anti-malware*, peretas mulai mengubah tanda tangan virus. *Anti-malware* meningkatkan kemampuan perangkat lunak mereka; peretas menemukan cara baru lagi untuk menghindari; dan seterusnya. Hal ini dapat dianalogikan dengan modus pencuri dan polisi.

b. Aplikasi Terpercaya (Trusted Application)

Saat ini, dengan merebaknya *malware*, sebagian besar produsen sistem operasi telah menambahkan fitur ke sistem operasi mereka. Sistem ini memberi peringatan kepada pengguna bahwa seharusnya perangkat lunak di komputer atau perangkat seluler berasal dari pengembang asli yang bersertifikat.

Pengembang perangkat lunak dapat mengajukan permohonan ke produsen sistem operasi, misalnya Apple, Microsoft, atau Google untuk sertifikat digital. Jika telah memiliki sertifikat digital, aplikasi apa pun yang dibuat oleh pengembang akan memiliki sertifikat digital yang menempel dan



merupakan aplikasi terpercaya (*trusted application*). Dengan mengaktifkan fitur ini pada sistem operasi, perangkat lunak yang berjalan di peranti harus berasal dari pengembang bersertifikat. Jika tidak, sistem operasi tidak mengizinkannya untuk berjalan.

Namun, sama seperti pembuatan virus, peretas mungkin juga menemukan kerentanan pada fitur ini, seperti proses yang digunakan sistem operasi Android untuk memvalidasi sertifikat. Peretas berhasil memalsukan sertifikat digital dari pengembang sehingga pembuat sistem operasi harus menambal sistem untuk mengatasi kesalahan tersebut.

Beberapa sistem operasi memungkinkan fitur *trusted application* dapat dimatikan (*disable*). Dengan demikian, aplikasi yang dikembangkan oleh perusahaan kecil, individu, atau dipakai untuk diri sendiri yang tidak memiliki sertifikat dapat dijalankan di ponsel/komputer. Namun, menggunakan aplikasi yang belum memiliki sertifikat ialah tindakan yang berisiko.

Sistem operasi seluler Apple, iOS ialah sistem operasi ponsel yang memiliki ketentuan yang lebih ketat. iOS mengharuskan semua aplikasi yang terpasang pada perangkat iPhone/iPad berasal dari pengembang yang bersertifikat. Apple menyediakan App Store, dimana aplikasi yang dapat dipasang di ponsel Apple hanya ada di App Store. Banyak orang menilai kebijakan ini membatasi kreativitas dan mengurangi persaingan. Pengguna juga menganggap fitur ini menjadikan iPhone dan iPad mereka tidak leluasa dikendalikan dan tidak fleksibel sehingga pengguna iPhone/iPad berusaha untuk meretas dengan menonaktifkan persyaratan sertifikasi. Hal ini biasanya disebut *jailbreaking*. Namun, dengan melakukan *jailbreaking*, kamu memperbesar kemungkinan perangkat akan terkena virus. Ada beberapa virus yang secara khusus menargetkan iPhone yang di-*jailbreak*. Jadi, sebenarnya, sebagai pengguna perangkat teknologi informasi dan internet, kamu harus menyeimbangkan antara: keamanan di satu sisi, atau fleksibilitas, kenyamanan, dan pengendalian oleh pengguna di sisi lainnya.

3. Autentikasi: Web, Pengguna, Biometrik, Autentikasi Ganda

Mekanisme keamanan berikutnya ialah autentikasi. Autentikasi dilakukan untuk memastikan dan mengonfirmasi bahwa suatu objek ialah otentik atau asli. Autentikasi dapat diterapkan pada beberapa objek, seperti situs web, *user*, *email*, dokumen, dll.



a. Autentikasi Web

Situs web merupakan salah satu sumber daya yang banyak digunakan di internet. Situs web dapat diautentikasi bahwa situs tersebut ialah autentik. *Browser web*, *search engine*, dan perangkat lunak tambahan (*add-on*) telah dapat membantu memfilter situs web yang dianggap autentik dan aman. Perangkat lunak tambahan juga memiliki fasilitas yang mampu untuk menampilkan peringatan bagi situs yang diketahui mengumpulkan dan menyalahgunakan informasi pribadi. Meskipun bermanfaat bagi pengguna, tetapi bentuk pemfilteran harus dilakukan dengan hati-hati. Filter yang terlalu ketat akan membuat situs web yang sebenarnya aman akan masuk kategori yang terfilter, seperti juga mekanisme filter *spam* pada *email*.

Kesalahan dalam menandai (memberi nilai/*rating*) untuk keperluan pemfilteran dapat merusak hubungan bisnis dan dapat mengakibatkan gugatan untuk perusahaan pemberi *rating*. Hal ini penting, baik dari perspektif etika maupun bisnis dalam bentuk kehati-hatian dalam merancang dan menerapkan sistem penilaian.

Selain itu, dengan maraknya kejahatan internet seperti *phising* dan *pharming* yang memalsukan internet, banyak institusi bisnis dan keuangan seperti perbankan memberi keamanan lebih pada pengguna daringnya. Institusi bank dan bisnis keuangan juga mengembangkan sistem keamanan untuk web mereka. Teknik ini digunakan untuk meyakinkan pelanggan bahwa web yang diakses oleh pelanggan ialah web autentik sebelum pelanggan memasukkan kata sandi atau informasi sensitif lainnya. Misalnya, saat pelanggan membuat akun pertama kali, web site meminta pelanggan untuk mengunggah gambar digital (misalnya, kucing/anjing peliharaan) atau memilih dari gambar yang tersedia. Pada saat berikutnya, ketika pelanggan mengakses web site, sebelum proses *login* dengan mengisi nama (atau pengenalan lain yang tidak penting), sistem akan menampilkan gambar yang diunggah/dipilih sebelumnya. Dengan demikian, situs mengautentikasi dirinya sendiri kepada pelanggan sebelum pelanggan melakukan autentikasi dirinya dengan memasukkan kata sandi.

b. Autentikasi Pengguna

Autentikasi pengguna ialah bagian penting dari keamanan informasi. Banyak kejahatan peretasan terjadi karena autentikasi pengguna ini. Autentikasi



pengguna umumnya menggunakan *username* dan *password*. *Username* dan *password* ialah data sensitif yang harus dijaga agar tidak diketahui orang lain.

Kejahatan dengan memanfaatkan autentikasi pengguna telah banyak terjadi. Di Amerika, seorang peretas membobol banyak akun milik pialang *online*, dan kemudian membeli saham menggunakan akun curian tersebut. Jumlah pembelian yang besar mendorong harga naik, dan peretas itu kemudian mendapatkan keuntungan dengan menjual sahamnya. Ketika seorang peretas dapat membobol akun kamu, banyak hal yang dapat dilakukan oleh peretas tersebut. Pembobolan ini juga sering terjadi pada bank *online* dan situs keuangan lainnya. Di Indonesia, penyalahgunaan pin ATM, kata sandi oleh orang yang tidak berhak marak terjadi yang disebabkan oleh penipuan atau banyak hal lainnya. Aksi *phising* juga pernah terjadi pada tahun 1990-an dimana situs palsu sebuah bank ternama di Indonesia mampu mendapatkan ribuan *username* dan *password* dari pengguna *online banking* dari bank tersebut.

Hal inilah yang mendorong institusi keuangan untuk mengembangkan prosedur yang lebih baik untuk mengotentikasi pengguna. Walaupun berbekal identitas dan nomor curian serta informasi identitas lain yang lain, perusahaan diharapkan dapat membedakan pemilik akun asli atau palsu. Autentikasi pengguna atau pelanggan secara jarak jauh pada dasarnya sulit. Dengan demikian, memang diperlukan informasi lebih dari yang biasa diperlukan untuk mengidentifikasi seseorang dan kemudian memberikan otorisasi.

Metode autentikasi yang saat ini banyak digunakan salah satunya ialah penggunaan biometrik dari pengguna seperti sidik jari, suara, wajah, retina mata, dll. Beberapa situs meminta memasukkan nama guru favorit, mobil yang dibeli pertama kali, dll. saat mendaftarkan diri sebagai pengguna, yang untuk selanjutnya diminta untuk dimasukkan kembali sebelum proses *login*. Ada juga situs yang menyimpan informasi mengenai peranti yang biasanya digunakan pelanggan untuk *login* dan meminta informasi tambahan ketika seseorang masuk dari peranti yang berbeda. Sebuah situs meminta pelanggan untuk memilih beberapa gambar saat membuka akun, dan kemudian meminta pelanggan untuk identifikasi gambar saat *login*. Hal-hal tersebut di atas ialah banyak metode yang terus berkembang untuk menambah keamanan berinternet.



Kemajuan perangkat lunak autentikasi muncul dengan teknik kecerdasan buatan. Perangkat lunak ini mampu menghitung risiko kepalsuan berdasarkan data perbedaan waktu *login*, jenis *browser* yang digunakan, perilaku, transaksi khas pelanggan, dll. Termasuk juga lokasi geografis dan tempat pengguna, biasanya *login* ke situs web juga dapat digunakan untuk konfirmasi.

c. **Biometrik**

Biometrik ialah karakteristik biologis yang unik bagi seorang individu. Termasuk di dalamnya ialah sidik jari, pola suara, struktur wajah, geometri tangan, pola mata (iris atau retina), dan DNA. DNA sudah lama digunakan dalam penegakan hukum dan sistem peradilan di Indonesia. Teknologi biometrik untuk identifikasi saat ini ialah industri penting bernilai miliaran dolar dengan banyak aplikasi yang bermanfaat dan memberikan kenyamanan dan keamanan.

Saat ini, telah banyak penggunaan biometrik untuk autentikasi. Kamu dapat membuka ponsel cerdas, tablet, dan pintu dengan menyentuh pemindai sidik jari. Dengan biometrik, kamu tidak lagi menggunakan kata sandi yang mungkin dapat terlupa, atau kunci fisik yang mungkin tertinggal atau hilang. Dengan sidik jari, akses peretas menjadi banyak berkurang.

Beberapa aplikasi ponsel cerdas saat ini telah menggunakan pengenalan sidik jari, wajah, atau suara. Hal itu bermanfaat untuk mengautentikasi pemilik dan melindungi pencurian informasi atau dana di dompet elektronik.

Untuk mengurangi risiko terorisme, beberapa bandara menggunakan sistem identifikasi sidik jari untuk memastikan bahwa hanya karyawan yang boleh memasuki area terlarang. Di pabrik, pekerja tidak lagi memasukkan kartu presensi fisik. Sebagai gantinya, mereka gunakan *scan* sidik jari tangan. Sekolah dan kampus saat ini di Indonesia, bahkan telah banyak yang menggunakan sidik jari atau wajah untuk presensi kehadiran.

Di sisi lain, peretas berusaha menemukan cara untuk mencari celah keamanan identifikasi biometrik. Peneliti di AS dan Jepang pernah melakukan riset dengan mengganti sidik jari manusia dengan sidik jari mayat atau dengan jari palsu yang dibuat dari gelatin atau plastisin (*Play-Doh*). Pemindai mata juga dapat ditipu dengan menggunakan lensa kontak.



Namun, teknologi biometrik juga berkembang jauh lebih baik. Saat ini misalnya, pemindai jari dapat mengambil gambar sidik jari dengan resolusi sangat tinggi dari sidik jari subdermal sehingga dapat membedakan sidik jari orang mati. Pemindaian iris mata menggunakan analisis pola lingkaran berwarna yang mengelilingi pupil mata, telah ada pada ponsel cerdas. Pemindaian retina diklaim lebih akurat, dimana mengambil gambar dan menganalisis pola unik pembuluh darah di retina mata. Pola retina mata juga akan memudar dengan cepat setelah kematian. Pemindaian dengan retina dianggap hampir tidak mungkin dapat dipalsukan.

d. Autentikasi Dua Langkah

Autentikasi dua langkah (*Two-Factor Authentication/TFA*) ialah metode keamanan yang memerlukan dua langkah atau faktor berbeda untuk mengonfirmasi identitas pengguna. Tujuannya ialah meningkatkan keamanan secara signifikan dibandingkan dengan autentikasi satu langkah yang biasa menggunakan kata sandi saja. Dengan menggunakan dua langkah yang berbeda, meskipun satu faktor keamanan diketahui atau dikompromi, yang lainnya masih dapat memberikan lapisan perlindungan.

Berikut ialah dua faktor umum yang digunakan dalam autentikasi dua langkah.

1) Faktor yang kamu ketahui (*Something You Know*)

Ini ialah faktor tradisional seperti kata sandi atau PIN yang diketahui oleh pengguna. Faktor ini mencakup informasi yang hanya diketahui oleh pengguna. Keamanannya dapat terancam jika kata sandi bocor atau diakses oleh pihak yang tidak berwenang.

2) Faktor yang kamu miliki (*Something You Have*)

Ini melibatkan kepemilikan suatu objek fisik atau informasi, seperti token autentikasi, kartu pintar, atau ponsel cerdas. Pengguna perlu memiliki objek ini untuk menyelesaikan proses autentikasi.

Contoh autentikasi dua langkah:

- Pertama: Pengguna memasukkan kata sandi mereka (Faktor yang Kamu Ketahui).



- Kedua: Setelah kata sandi diakui, sistem mengirimkan kode verifikasi unik ke ponsel pengguna melalui pesan teks atau aplikasi autentikasi (Faktor yang Kamu Miliki). Pengguna kemudian memasukkan kode tersebut untuk menyelesaikan autentikasi.

Keuntungan autentikasi dua langkah melibatkan peningkatan keamanan karena memerlukan lebih dari sekadar pengetahuan kata sandi, yang dapat lebih mudah terkompromi. Bahkan, jika kata sandi bocor, seseorang masih memerlukan Faktor yang Kamu Miliki (seperti ponsel atau token) untuk mengakses akun atau sistem. Sistem autentikasi dua langkah umumnya digunakan di berbagai platform *online*, layanan *email*, perbankan *online*, dan aplikasi keamanan lainnya untuk melindungi data dan informasi pengguna.

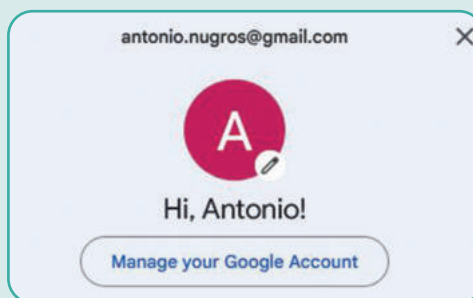
Aktivitas 10.4-03-P **Autentikasi Dua Langkah**

Pada aktivitas ini, kamu belajar untuk mengotentikasi dua langkah akun google kamu. Oleh sebab itu, kamu memerlukan komputer yang telah terpasang peramban dan koneksi internet. Jika tidak ada, aktivitas ini dapat dilakukan secara *unplugged*.

Dengan autentikasi atau verifikasi 2 langkah, kamu dapat menambahkan lapisan keamanan ekstra ke akun jika kata sandimu dicuri. Setelah kamu menyiapkan Verifikasi 2 Langkah, kamu dapat *login* ke akun kamu dengan menggunakan kata sandimu dan ponselmu. Ikuti langkah berikut untuk mengizinkan verifikasi 2 langkah pada akun googlemu.

Langkah-Langkah

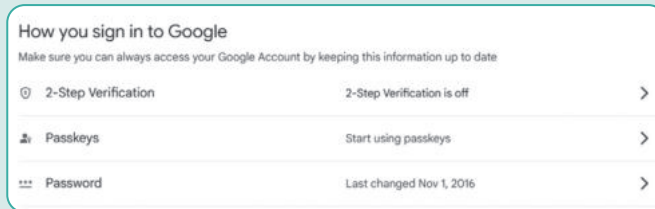
1. Buka akun Google kamu.



Sumber: Tangkapan layar gmail.com

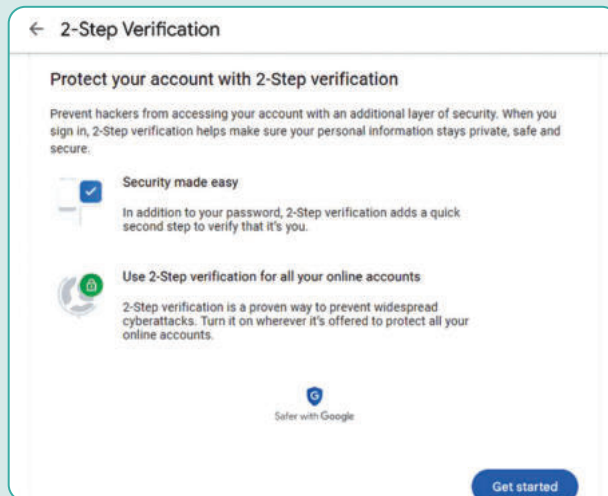


2. Klik Manage your Google Account.
3. Di panel navigasi, pilih Security/Keamanan.
4. Di bagian “Login ke Google”, pilih Verifikasi 2 Langkah lalu Mulai.



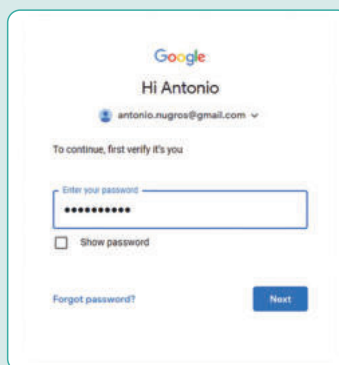
Sumber: Tangkapan layar gmail.com

5. Baca perintah dan ikuti langkah-langkah di layar.



Sumber: Tangkapan layar gmail.com

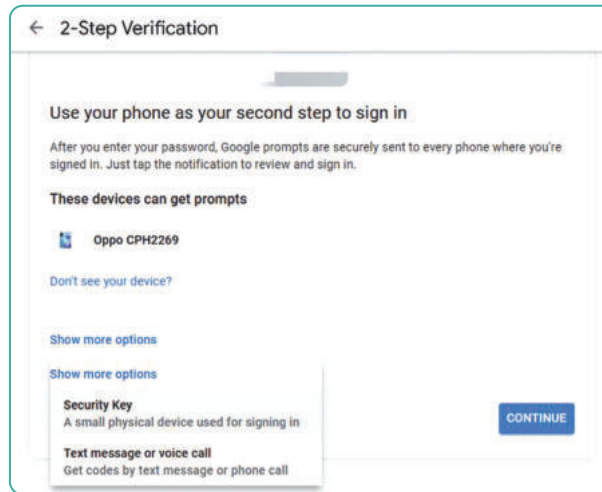
6. Pilih *Get started*.



Sumber: Tangkapan layar gmail.com

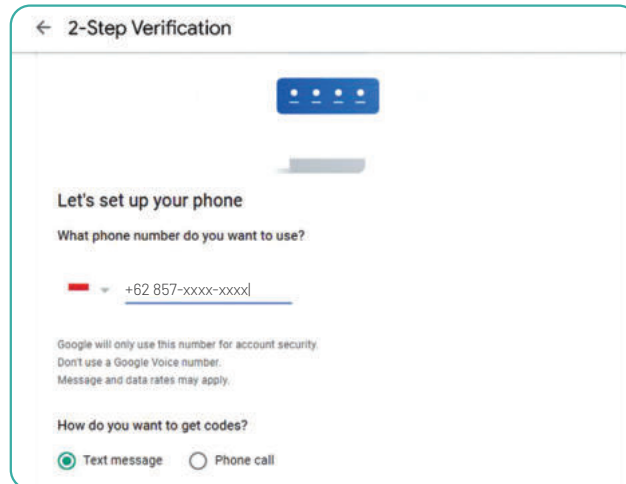


7. Setelah itu, verifikasi akunmu yang saat ini sedang digunakan, kemudian klik *Next*.



Sumber: Tangkapan layar gmail.com

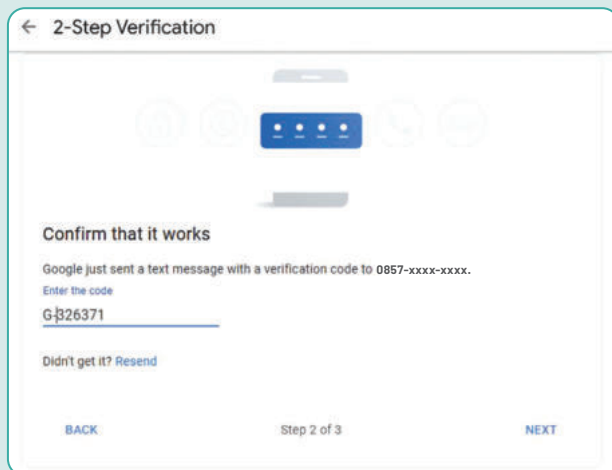
8. Verifikasi menggunakan telepon sebagai langkah kedua saat *login*. Selanjutnya, pilih nomor telepon yang akan dijadikan alat verifikasi kedua.



Sumber: Tangkapan layar gmail.com

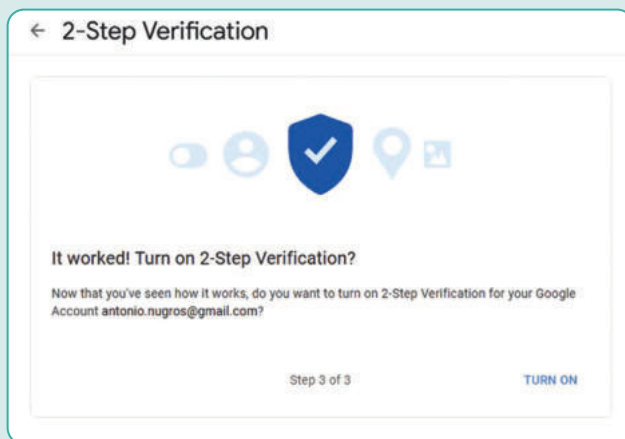
9. Verifikasi kedua setelah *username* dan *password*, akan menggunakan telepon yang kamu miliki. Pilih nomor telepon yang akan dijadikan alat verifikasi kedua.





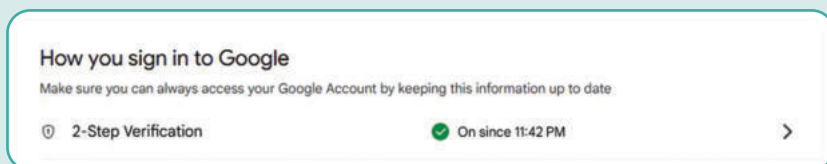
Sumber: Tangkapan layar gmail.com

10. Setelah itu, ada konfirmasi untuk memastikan verifikasi dua langkah ini berhasil.



Sumber: Tangkapan layar gmail.com

11. Jika sukses, verifikasi dua langkah telah berjalan dengan baik. Sekarang, untuk masuk ke akunmu, kamu harus melakukannya dengan dua langkah. Akunmu sekarang lebih aman.



Sumber: Tangkapan layar gmail.com



Tips: Jika kamu menggunakan akun yang dibuat oleh sekolah, langkah-langkah ini mungkin tidak berfungsi. Jika kamu tidak dapat mengatur verifikasi 2 langkah pada akunmu, hubungi administrator sekolah untuk mendapatkan bantuan.

Ayo, Renungkan!

Apa kelebihan dan kekurangan menggunakan autentikasi dua langkah tersebut, ditinjau dari keamanan data dan kenyamanan pengguna?

e. Autentikasi Multifaktor

Autentikasi pengguna dengan menggunakan kata sandi dan pemindaian biometrik, saat ini cukup memadai. Namun, dengan banyaknya usaha untuk meretas, muncul autentikasi dengan metode lain. Metode lain tersebut disebut dengan autentikasi multifaktor. Pada teknologi autentikasi, terdapat tiga kategori teknologi, yaitu seperti berikut.

1. Sesuatu yang diketahui oleh pengguna, misalnya kata sandi, PIN, atau frasa kunci rahasia.
2. Sesuatu tentang diri pengguna, seperti: suara, sidik jari, atau pemindaian retina.
3. Sesuatu yang dimiliki pengguna, misalnya kartu (kartu masuk, debit, kredit), ponsel cerdas, pin generator (seperti key BCA, token Mandiri), atau fob.

Metode autentikasi multifaktor menggunakan setidaknya dua item dari kategori berbeda. Beberapa contohnya ialah seperti berikut.

- a. Autentikasi menggeser kartu debit (3) dan memasukkan PIN (1).
- b. Memasukkan kata sandi (1) dan kemudian mengetik kode khusus situs web yang dikirim ke ponsel kamu (3).
- c. Berbicara (2) dan frasa kunci rahasia (1).
- d. Menggunakan sidik jari (2) dan kode khusus dari fob (3).

Memasukkan kata sandi dan PIN tidak akan menjadi autentikasi multifaktor karena kedua item berada dalam satu kesatuan kategori.



Aktivitas 10.4-04-U Merancang Autentikasi

Aktivitas ini mengajak kamu untuk bernalar kritis. Sebagai pengembang perangkat lunak, kamu mendapat proyek untuk merancang autentikasi sebuah ruang yang menyimpan teknologi rahasia dan *blueprint* alat. Untuk amannya, ruang ini harus dilengkapi dengan autentikasi yang multi faktor. Ruang ini hanya boleh diakses oleh pemilik dan keluarganya yang berjumlah 4 orang dewasa.

Tantangan

Rancanglah model autentikasi pada proyek ini. Jelaskan alasan menggunakan model tersebut.

D. Meningkatkan Keamanan Informasi

Saat ini, dengan makin banyaknya kejahatan di internet karena aktivitas *online*, pengembang perangkat lunak ikut bertanggung jawab terhadap keamanan pengguna dengan mengembangkan fitur untuk keamanan informasi. Fitur tersebut terdapat pada sistem operasi, browser, dan aplikasi.

1. Penggunaan Fitur Keamanan pada Sistem Operasi

Salah satu fitur keamanan pada sistem operasi ialah fitur keamanan yang ada pada Windows 10, yaitu Windows Security. Fitur ini dapat digunakan untuk memproteksi peranti dan data yang ada pada komputer. Fitur Windows Security di antaranya ialah seperti berikut.

a. Virus and Threat Protection

Perlindungan terhadap virus dan ancaman lain. Fitur perlindungan terhadap virus dinamakan Microsoft Defender Antivirus. Fitur ini dapat digunakan untuk memonitor/memantau ancaman keamanan di peranti yang kamu miliki. Dengan fitur ini, kamu mendapatkan perlindungan terhadap virus dan ancaman lain dengan antivirus terbaru. Peranti kamu akan dilindungi secara aktif ketika Windows 10 dihidupkan. Fitur ini akan terus-menerus memindai *malware* (perangkat lunak berbahaya), virus, dan ancaman keamanan lainnya. Selain perlindungan secara *real-time* (waktu nyata),



pembaruan dapat diunduh secara otomatis untuk membantu menjaga perangkat tetap aman dan terlindungi dari ancaman keamanan.

b. Account Protection (Perlindungan Akun)

Akses *login* dengan bermacam pilihan dan *settings*, termasuk menggunakan kunci dinamis tersedia pada Windows Security.

c. Firewall and Network Protection (Firewall dan Perlindungan Jaringan)

Fitur ini memungkinkan pengguna mengelola setelan *firewall* dan memonitor kegiatan dan aktivitas yang terjadi pada jaringan dan koneksi internet.

d. Application Control and Browser (Kontrol Aplikasi dan Browser)

Fitur ini digunakan untuk mengatur Microsoft Defender SmartScreen yang digunakan untuk membantu melindungi peranti dari aplikasi, *file*, situs, dan unduhan yang berpotensi berbahaya. Pengguna dapat mengatur perlindungan sesuai dengan kebutuhan.

e. Device Security (Keamanan Peranti)

Fitur ini digunakan untuk membantu melindungi perangkat dari serangan perangkat lunak berbahaya. Pada fitur ini, pengaturan dapat dilakukan untuk mengisolasi proses pada sistem operasi dan memori dari jangkauan *malware*. *Security processor* digunakan untuk menambah kemampuan enkripsi pada perangkat. Fitur *secure boot* ialah fitur untuk mencegah *malware* dengan tipe RootKit untuk hidup di peranti, yang dicegah pada saat *booting*.

f. Device Performance and Health (Kinerja dan Kesehatan Perangkat)

Fitur ini digunakan untuk menyajikan informasi tentang kesehatan kinerja perangkat dan menjaga perangkat tetap bersih dan mutakhir dengan versi terbaru Windows 10.

2. Penggunaan Fitur Keamanan pada Peramban

Maraknya *malware* dan virus di internet menyebabkan komputer kamu rentan terhadap ancaman *online*. Ketika kamu melakukan penjelajahan di internet, selalu ada kemungkinan untuk tersusupi oleh ancaman tersebut. Namun, dengan berkembangnya teknologi keamanan informasi perangkat lunak, peramban yang



kamu lakukan menjadi makin aman. Ada beberapa fitur penting dari peramban yang harus kamu ketahui, dan tips penting penjelajahan di internet yang aman. Fitur penting tersebut yaitu:

a. Periksa Alamat Situs

Dewasa ini, banyak situs palsu yang bertebaran di internet. *Phising* dan *pharming* ialah kejahatan yang mencoba mengecoh kamu dengan situs palsu. Memeriksa ulang alamat situs ialah teknik yang baik yang dapat digunakan untuk memastikan alamat situs yang kamu tuju. Sebagai contoh, situs klikbca.com dan situs klickbca.com hanya memiliki perbedaan satu huruf. Akibatnya, salah tulis akan membuat kamu akan dibawa ke situs yang berbeda.



Gambar 4.15 Contoh Situs dengan Perbedaan Satu Huruf

Sumber: Tangkapan layar situs klikbca.com dan klickbca.com

b. Yakinkan Simbol Keamanan

Situs web yang aman ialah situs web yang menggunakan protokol HTTPS (*Hypertext Transfer Protocol Secure*), yang tampak dengan simbol gembok pada *address bar* dari situs web yang kamu akses. Situs perbankan dan e-commerce biasanya menggunakan protokol ini. Protokol ini ialah protokol keamanan yang membuat situs web aman untuk digunakan bertransaksi di internet. Informasi pribadi dan keuangan kamu aman untuk dimasukkan ke dalam website dengan simbol ini. HTTPS menggunakan *Transport Layer Security* (TLS) atau *Secure Socket Layer* (SSL) yang membuat komunikasi data dilakukan dengan enkripsi.

Lakukanlah pembaruan berkala pada peramban. Dengan berjalannya waktu, ancaman keamanan di internet selalu bertambah dan diperbarui, termasuk munculnya virus dan *malware* baru. Peramban harus menyesuaikan



diri dengan hal tersebut sehingga pembaruan peramban merupakan cara yang penting untuk menjaga keamanan berinternet. Peramban biasanya menginformasikan pesan jika ada pembaruan pada peramban, yang dapat dilakukan secara otomatis maupun manual.

c. Pelacakan pada Peramban

Saat kamu menggunakan peramban untuk melakukan penjelajahan di internet, semua riwayat penjelajahan kamu akan disimpan oleh peramban. Semua situs web yang kamu kunjungi dan jelajahi akan disimpan dalam *cookie* di peramban.

Pelacakan aktivitas penjelajahan pengguna internet dan menyimpannya pada *cookie* digunakan untuk memaksimalkan kinerja dari peramban dalam hal kecepatan yang membuat penggunanya merasa nyaman. Namun, data pada *cookie* juga dapat digunakan untuk memetakan kebiasaan dan preferensi pengguna ketika menjelajahi internet. *Cookie* juga dapat digunakan oleh pengiklan untuk menampilkan iklan barang-barang yang kamu sukai.

Berikut ialah contoh-contoh pelacakan pada peramban.

1. Youtube, Tokopedia, Netflix ialah situs web yang mempelajari video atau produk apa saja yang sering kamu tonton atau cari. Dengan data kamu di *cookie*, situs web tersebut akan merekomendasikan video yang berpeluang untuk kamu tonton dan produk yang kamu sukai dan berpotensi untuk kamu beli.
2. *Search engine* seperti Google dan Bing juga menyimpan riwayat pencarian kamu di *cookie*. Riwayat pada *cookie* ini akan digunakan untuk menghasilkan hasil pencarian yang lebih relevan dan juga berhubungan dengan iklan di internet.

Bagaimana *cookie* bekerja?

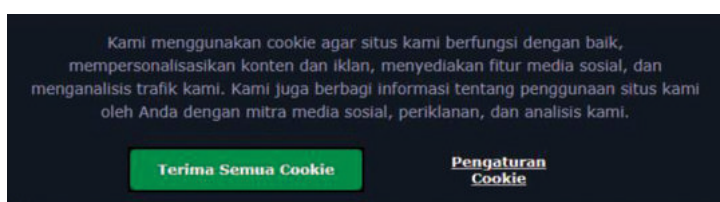
Cookie pada peramban ialah kumpulan data kecil yang digunakan untuk menyimpan informasi aktivitas kamu dalam penjelajahan situs web. Ketika kamu menulis alamat web, mengklik hyperlink atau icon, semua aktivitas tersebut akan dicatat di *cookie*. *Cookie* juga dapat menyimpan informasi spesifik dari situs web tertentu yang kamu kunjungi. Jika kamu tidak memiliki akun pada situs tersebut, riwayat penjelajahan akan tersimpan pada *cookie* di peramban kamu. Sebagai contoh, misalnya ada situs berita



yang menggunakan *cookie*, berita apa saja yang kamu akses sebelumnya akan disimpan dan digunakan untuk memberikan rekomendasi berita yang tepat ketika kamu mengunjungi situs berita itu kembali.

Pada umumnya, data pada *cookie* tidak terlalu berbahaya untuk aktivitas kamu di internet. Kecil kemungkinan data pada *cookie* akan disusupi dengan *malware* dan mencuri data pribadi kamu. Namun walaupun begitu, kamu tetap dapat mengatur *cookie* pada peramban untuk tidak menyimpan riwayat penjelajahan kamu.

Banyak situs web memberikan pilihan apakah kamu akan mengaktifkan *cookie* atau tidak. Jika tidak ingin mengaktifkan *cookie*, pilih *decline*/menolak.



Gambar 4.16 Contoh tampilan pengguna untuk memilih mengaktifkan cookie di situs antivirus.

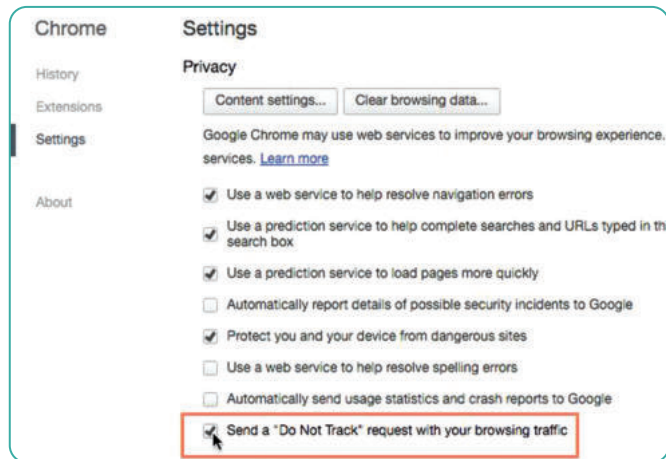
Berikut contoh tampilan pemberitahuan penggunaan *cookie* pada situs salah satu bank di Indonesia.



Gambar 4.17 Contoh tampilan pemberitahuan penggunaan *cookie* pada situs salah satu bank di Indonesia.

Peramban modern saat ini telah memberikan pengaturan yang dapat menonaktifkan fitur *cookie* sehingga tidak mencatat riwayat penjelajahan kamu. Cara untuk menonaktifkan *cookie* ialah dengan menghidupkan *setting* “Do Not Track” di peramban kamu. Sebagian besar situs web sebenarnya secara *default* menonaktifkan fitur ini. Gambar 4.15 ialah contoh pengaturan *cookie* “Do Not Track” pada peramban Chrome.





Gambar 4.18 Contoh pengaturan fitur *cookie* pada peramban Chrome.

Sumber: Tangkapan layar Google Chrome

Pada beberapa situs web, terkadang tidak memberikan fitur “*Do Not Track*” kepada penggunanya. Jika kamu ingin menghindari pencatatan aktivitas internetmu, kamu dapat menggunakan penjelajahan pribadi, yang berbeda menu pada tiap peramban. Pada Google Chrome, dikenal dengan istilah Incognito Window, dimana dapat dipilih pada menu New incognito Window. Pada Microsoft Edge, dapat digunakan dengan menu New InPrivate window. Pada Firefox, dapat digunakan dengan fitur New Private Window. Peramban Safari menggunakan menu New Private Window untuk *private browsing*.

d. Pelacakan Akun

Meskipun fitur *cookie* pada peramban telah dinonaktifkan, tetapi riwayat aktivitas pada akun tertentu masih dapat dilacak. Ketika kamu membuat akun pada situs media sosial seperti facebook misalnya, sebenarnya, kamu telah memberikan izin kepada facebook untuk mencatat aktivitas kamu dengan akun facebook tersebut. Catatan riwayat aktivitas kamu dengan akun tidak disimpan secara lokal pada peramban di komputer kamu, tetapi dicatat di server. Riwayat ini biasanya diberikan kepada pengiklan di aplikasi facebook dan secara *default*, fitur ini aktif. Pengaturan riwayat akun *online* biasanya dapat diatur. Gambar berikut menunjukkan contoh pengaturan riwayat di facebook.



Facebook Ads

Third Party Sites

Facebook does not give third party applications or ad networks the right to use your name or picture in ads. If we allow this in the future, the setting you choose will determine how your information is used.

You may see social context on third party sites, including in ads, through Facebook social plugins. Although social plugins enable you to have a social experience on a third party site, Facebook does not share your information with the third party sites hosting the social plugins. [Learn more about social plugins.](#)

If we allow this in the future, show my information to

No one
Only my friends
No one

Save Changes Cancel

Gambar 4.19 Contoh pengaturan riwayat di facebook.

3. Konten Negatif di Dunia Digital

Internet saat ini telah memiliki konten yang sangat banyak. Konten tersebut banyak yang membantu manusia dalam berbagai hal. Namun, dengan ketiadaan kurasi dan pengendalian, konten di internet dapat berbentuk konten yang bersifat positif dan negatif.

a. Konten Positif

Konten positif ialah jenis konten yang memberikan manfaat, edukasi, inspirasi, hiburan yang sehat, atau dampak positif pada pemirsa. Konten positif dapat memotivasi, mengedukasi, dan membantu orang untuk tumbuh dan berkembang secara pribadi. Berikut beberapa jenis konten positif di internet.

1. Edukasi: Video tutorial, artikel, dan sumber daya lainnya yang memberikan informasi dan pengetahuan yang berharga tentang berbagai topik, seperti ilmu pengetahuan, seni, keterampilan, atau sejarah.
2. Hiburan yang Sehat: Video, musik, film, dan *game* yang menghibur tanpa merusak nilai-nilai atau moral.
3. Motivasi dan Inspirasi: Video, kutipan, atau cerita yang menginspirasi dan memotivasi orang untuk mencapai tujuan dan potensi terbaik mereka.
4. Seni dan Kreativitas: Karya seni, musik, fotografi, dan seni rupa yang memamerkan bakat dan kreativitas individu.



5. Berita Positif: Berita yang mengangkat cerita-cerita positif, kesuksesan, dan kebaikan yang terjadi di dunia.
6. *Podcast Pendidikan*: *Podcast* yang membahas topik-topik yang mendidik dan memberikan wawasan baru kepada pendengarnya.
7. Lingkungan dan Keberlanjutan: Konten yang mendukung kesadaran lingkungan, pelestarian alam, dan keberlanjutan.
8. Konten Anak-Anak yang Mendidik: Video dan *game* yang dirancang khusus untuk anak-anak dengan tujuan mendidik dan menghibur secara sehat.
9. Cerita Inspiratif: Cerita nyata tentang orang-orang yang mengatasi rintangan dan mencapai kesuksesan, yang dapat menginspirasi orang lain.

Memilih untuk mengonsumsi dan berkontribusi pada konten positif di internet dapat membantu menciptakan lingkungan online yang lebih sehat. Hal itu akan memberikan manfaat bagi individu dan masyarakat secara keseluruhan.

b. Konten Negatif

Konten negatif di internet ialah jenis konten yang dapat merusak, merugikan, atau memiliki dampak negatif pada pemirsa. Konten ini sering kali tidak sejalan dengan norma-norma sosial atau etika, dan dapat mencakup berbagai bentuk yang merugikan. Berikut ialah beberapa contoh jenis konten negatif di internet.

1. Kekerasan Fisik dan Nonfisik: Video atau gambar yang menampilkan tindakan kekerasan, pelecehan fisik, atau perlakuan kasar terhadap individu atau makhluk lain.
2. Pelecehan, Pencemaran Nama Baik, dan Pelecehan Seksual: Konten yang mengejek, mencemarkan nama baik, atau melecehkan individu, sering kali dalam bentuk pelecehan seksual.
3. Kebencian dan Diskriminasi: Materi yang mengandung pesan kebencian, diskriminasi rasial, etnis, agama, gender, orientasi seksual, atau kelompok sosial tertentu.
4. Konten Seksual yang Eksploitatif: Materi yang memuat konten seksual yang merugikan, eksploitatif, atau tidak sesuai dengan norma sosial.



5. **Konspirasi dan Disinformasi:** Informasi palsu atau teori konspirasi yang dapat meresahkan atau membingungkan pemirsa.
6. **Bullying dan Tindakan Intimidasi:** Tindakan penganiayaan verbal atau *cyberbullying* yang dapat merugikan mental dan emosional korban.
7. **Konten yang Memprovokasi Kekerasan:** Materi yang memotivasi atau memprovokasi tindakan kekerasan atau kejahatan.
8. **Penghinaan dan Kritik yang Tidak Sehat:** Kritik yang kasar, penghinaan, atau retorika yang merugikan individu atau kelompok tertentu.
9. **Gangguan Privasi:** Pelanggaran privasi pribadi, seperti penyebaran informasi pribadi tanpa izin.

Penting untuk menjaga kewaspadaan saat mengakses internet dan membantu mengidentifikasi dan melaporkan konten negatif ketika ditemui. Banyak platform *online* memiliki pedoman komunitas dan mekanisme pelaporan untuk melindungi pengguna dari konten negatif dan melindungi integritas lingkungan *online*.

Negara saat ini telah berusaha keras untuk mencegah beredarnya konten negatif di internet. Pelaksanaan pencegahan penyebaran konten negatif telah diamanatkan dalam Undang-Undang No. 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (ITE). Pemerintah juga berkewajiban untuk memfasilitasi pemanfaatan teknologi informasi dengan baik, serta dalam kondisi tertentu apabila diperlukan akan melakukan pemutusan akses terkait dengan Pasal 40 ayat (2a) pada UU No. 19 Tahun 2016 tentang Revisi UU ITE.

c. Menyaring Konten Negatif di Dunia Digital

Konten negatif yang beredar di internet sangatlah merugikan bagi kita semua. Untuk menyaring konten negatif di internet, kamu dapat menggunakan berbagai cara, di antaranya ialah seperti berikut.

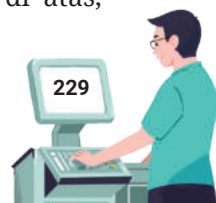
1. **Gunakan Perangkat Lunak Penyaring:** Sejumlah perangkat lunak penyaring konten dapat membantu kamu untuk memblokir atau menyaring konten yang tidak diinginkan. Selain itu, kamu juga dapat menggunakan perangkat keras (misalnya, *router* dengan fitur penyaringan) atau perangkat lunak penyaring seperti perangkat pelindung keluarga seperti Norton Family, Kaspersky Safe Kids, atau Qustodio, menawarkan fitur pelindung



keluarga yang dapat membantu membatasi akses ke konten yang tidak diinginkan.

2. Menggunakan *Domain Name System* (DNS) untuk memblokir situs konten negatif. Salah satunya ialah DNS Internet Sehat. DNS Internet Sehat ialah layanan DNS yang dibuat oleh pemerintah Indonesia untuk memblokir akses ke situs web yang dianggap memiliki konten negatif, ilegal, atau merugikan. DNS bertindak sebagai penerjemah alamat situs web yang mudah diingat (seperti www.contoh.com) ke alamat IP numerik yang sesungguhnya yang digunakan oleh server web. Layanan DNS Internet Sehat diperkenalkan untuk membantu mencegah akses ke situs web yang dianggap melanggar hukum atau berpotensi membahayakan. Ini sering kali digunakan sebagai bagian dari upaya untuk membatasi akses ke situs web yang berisi pornografi, perjudian, atau konten lainnya yang dianggap tidak sesuai dengan norma-norma moral atau hukum setempat.
3. Aktifkan Kontrol Orang Tua: Banyak sistem operasi dan perangkat memiliki kontrol orang tua yang memungkinkan kamu mengatur batasan akses internet, memblokir situs web tertentu, dan mengontrol jenis konten yang dapat diakses oleh anak-anak atau anggota keluarga lainnya.
4. Perbarui Perangkat dan Aplikasi: Pastikan perangkat dan perangkat lunak keamanannya selalu diperbarui. Pembaruan perangkat lunak keamanan akan membantu melindungi perangkat dari ancaman terbaru.
5. Pendidikan dan Kesadaran: Edukasi tentang bahaya konten negatif di internet ialah kunci. Kenali dan hindari konten yang merugikan. Diskusikan perilaku *online* yang aman dan etis.
6. Gunakan Mesin Pencari Aman: Beberapa mesin pencari, seperti “Kiddle” untuk anak-anak, telah dirancang khusus untuk menyaring hasil pencarian yang aman.
7. Laporkan Konten Negatif: Jika kamu menemukan konten negatif di internet, lakukan pelaporan kepada platform atau penyedia layanan terkait. Banyak platform memiliki mekanisme pelaporan yang memungkinkan pengguna untuk memberikan masukan terkait konten yang melanggar pedoman komunitas.

Ingatlah bahwa tidak ada solusi sempurna untuk menyaring semua konten negatif di internet dengan kombinasi langkah-langkah di atas,



kamu dapat mengurangi risiko terpapar konten yang tidak diinginkan atau merugikan. Selain itu, komunikasi terbuka dan pendidikan tentang penggunaan internet yang aman sangat penting untuk melindungi diri sendiri dan anggota keluarga dari konten negatif.



Aktivitas 10.4-05-U

Menyaring Konten Negatif di Internet

Aktivitas ini mengajak kamu bernalar kritis untuk menyaring konten negatif di internet dengan menggunakan DNS Internet Sehat Kementerian Komunikasi dan Informatika. Kamu diharapkan mengatur DNS Internet Sehat di perangkatmu, sesuai dengan sistem operasi yang digunakan. Langkah-langkahnya ialah sebagai berikut.

Langkah-Langkah

a. Untuk Sistem Operasi Windows

1. Buka “Control Panel”.
2. Pilih opsi “Network and Sharing Center”.
3. Klik pada “Change adapter settings”. Di panel kiri, pilih “Change adapter settings”.
4. Klik kanan pada koneksi aktif kamu: misalnya, koneksi WiFi atau Ethernet. Pilih “Properties”.
5. Pilih “Internet Protocol Version 4 (TCP/IPv4)”: Dalam daftar, cari dan pilih opsi ini, kemudian klik “Properties”.
6. Gunakan DNS Internet Sehat: Pilih opsi “Use the following DNS server addresses” dan masukkan alamat DNS Internet Sehat. Biasanya, alamat DNS Internet Sehat ialah 180.131.145.145 dan 180.131.145.150.
7. Klik “OK”: Setelah memasukkan alamat DNS, klik “OK” untuk menyimpan perubahan.

b. macOS

1. Buka “System Preferences”.
2. Pilih “System Preferences” dari menu Apple.
3. Pilih “Network”: Klik ikon “Network”.



4. Pilih koneksi aktif kamu: misalnya, Wi-Fi atau Ethernet. Klik “*Advanced*”.
5. Beralih ke tab “DNS”: Di jendela baru, beralih ke tab “DNS”.
6. Tambahkan DNS Internet Sehat: Klik “+”, dan tambahkan alamat DNS Internet Sehat. Biasanya, alamat DNS Internet Sehat ialah 180.131.145.145 dan 180.131.145.150.
7. Klik “OK” dan “Apply”. Setelah menambahkan DNS, klik “OK” dan kemudian “Apply” untuk menyimpan perubahan.

c. Linux

1. Edit *file* resolv.conf:
2. Buka terminal dan edit *file* resolv.conf dengan perintah berikut:

```
bash Copy code  
  
sudo nano /etc/resolv.conf
```

3. Tambahkan DNS Internet Sehat: Tambahkan baris berikut *file* resolv.conf:

```
nameserver 180.131.145.145  
nameserver 180.131.145.145
```

4. Simpan perubahan: Simpan perubahan dan keluar dari editor.
5. *Restart* jaringan: Jalankan perintah berikut untuk *merestart* layanan jaringan:

```
sudo /etc/init.d/networking restart
```

6. Namun, perlu dipastikan untuk menggantikan alamat DNS Internet Sehat dengan alamat DNS yang benar saat ini karena dapat berubah dari waktu ke waktu.
7. Setelah itu, akseslah beberapa situs, dan buktikan bahwa situs tersebut sudah terblokir.



Ayo, Eksplorasi!

1. Jelaskan bagaimana cara kerja DNS Internet Sehat sehingga dapat digunakan untuk menyaring konten negatif di internet.
2. Carilah DNS yang lain yang dapat digunakan untuk menyaring konten negatif di internet.

4. Pengelolaan Kata Sandi

Pengelolaan kata sandi ialah proses untuk mengelola kata sandi dengan aman. Hal ini biasa dilakukan dengan perangkat lunak pengelola yang disebut *password manager*. Pengelola kata sandi ialah alat yang membantu pengguna menyimpan dan mengelola kata sandi dengan aman. Dengan menggunakan manajer kata sandi, pengguna dapat meningkatkan keamanan kata sandi mereka, mengelola akses dengan lebih efisien, dan menjaga keamanan akun *online* mereka.

Perangkat lunak *password manager* di antaranya ialah Google Password Manager, LastPass, 1Password, atau Bitwarden. Perangkat lunak ini memiliki fitur yang membantu pengguna untuk melakukan beberapa hal berikut.

1. Buat akun atau masuk dengan akun yang sudah ada, dan mengatur preferensi keamanan, seperti pengaturan autentikasi dua faktor (2FA) jika didukung.
2. Simpan Kata Sandi, dan sistem akan menawarkan untuk menyimpannya secara otomatis.
3. *Autofill* Kata Sandi: Ketika pengguna mengunjungi situs web atau aplikasi yang membutuhkan *login*, manajer kata sandi akan menawarkan untuk mengisi otomatis kolom nama pengguna dan kata sandi.
4. Kelola Kata Sandi yang ada: (a) Akses papan kontrol manajer kata sandi untuk melihat dan mengelola semua kata sandi yang tersimpan. (b). Mengganti atau perbarui kata sandi yang sudah ada.
5. Hasilkan Kata Sandi Kuat: Gunakan fitur penghasil kata sandi kuat yang biasanya disertakan dengan manajer kata sandi untuk membuat kata sandi yang sulit ditebak.



6. Berbagi Kata Sandi Aman: Jika diperlukan, manajer kata sandi memungkinkan kamu berbagi akses ke akun atau layanan tertentu dengan orang lain tanpa harus memberikan kata sandi secara langsung.
7. Periksa Keamanan: Manajer kata sandi umumnya menyertakan alat pemeriksaan keamanan untuk menilai kekuatan kata sandi yang kamu gunakan dan memberikan saran perbaikan jika diperlukan.
8. Sinkronisasi Antarperangkat: Jika kamu menggunakan manajer kata sandi di beberapa perangkat, pastikan untuk mengaktifkan sinkronisasi sehingga perubahan yang kamu buat di satu perangkat akan tercermin di perangkat lainnya.
9. Gunakan Aplikasi *Mobile*: Pasang aplikasi manajer kata sandi pada perangkat *mobile* kamu untuk mengelola kata sandi saat bepergian.
10. Gunakan Integrasi *Browser*: Aktifkan integrasi *browser* untuk mempermudah penggunaan manajer kata sandi saat *browsing web*.
11. Perbarui secara Reguler: Lakukan pembaruan reguler pada aplikasi manajer kata sandi untuk mendapatkan fitur terbaru dan keamanan tambahan.



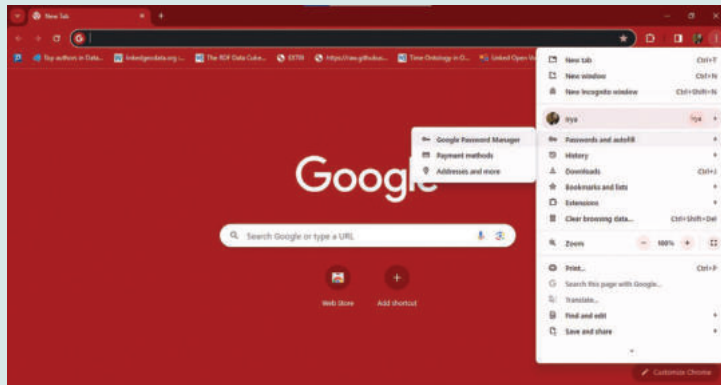
Aktivitas 10.4-06-P Mengelola Kata Sandi

Aktivitas ini mengajak kamu untuk bernalar kritis untuk mengelola kata sandi yang kamu miliki. Kamu mungkin memiliki banyak kata sandi untuk berbagai aplikasi yang kamu gunakan. Pada aktivitas ini, kamu berlatih menggunakan salah satu aplikasi pengelolaan kata sandi yang ada, yaitu Google Password Manager atau Pengelola Sandi Google.

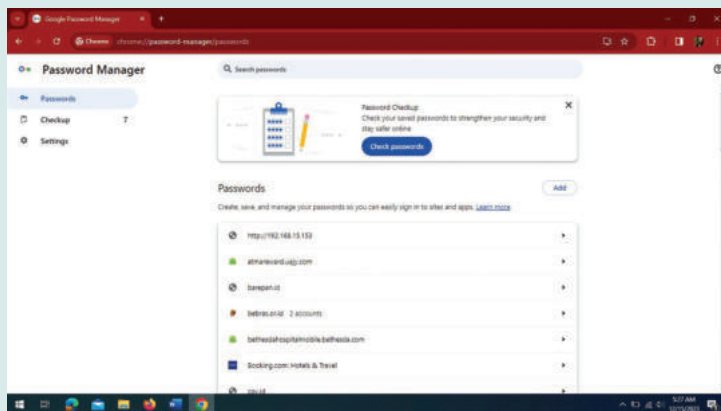
Langkah-Langkah

1. Menambahkan *password* pada situs tertentu dengan Google Password Manager. Lakukan dengan langkah-langkah seperti berikut.
 - a. Gunakan peramban Chrome. Klik menu pada perambanmu, pilih *Passwords* dan *Autofill* dan pilih menu Google Password Manager.





- b. Pilih tombol Add pada Passwords untuk menambahkan *password* yang akan dikelola.



- c. Isikan *password* yang kamu miliki yang digunakan pada situs web tertentu. Kemudian, klik *save*.



- d. Ketika sudah tersimpan, kamu dapat menggunakan *password* tersebut saat mengunjungi situs yang tersimpan datanya di aplikasi pengelola *password* tersebut. Kamu tidak perlu lagi memasukkan *username/email address* dan *password*-nya.



2. Selanjutnya, eksplorasilah fitur Google Password Manager ini dengan mencari fitur untuk menghapus dan memperbaharui *password*-mu. Penting untuk mengamankan informasi yang kamu miliki di internet dengan secara periodik mengganti *password*. Kemudian, carilah kelebihan dan kekurangan menggunakan aplikasi Password Manager ini. Berikan jawabanmu pada tabel berikut.

No.	Kelebihan	Kekurangan
1.		
2.		
3.		

5. Konfigurasi Privasi dan Keamanan

Menerapkan konfigurasi privasi dan keamanan pada akun peranti digital ialah langkah penting untuk melindungi informasi pribadi dan menjaga keamanan akunmu. Peranti digital yang ada saat ini terus berkembang dan beragam dari media sosial, *email*, *game*, sampai layanan *online* lainnya seperti dompet elektronik, *online banking*, dll.

Konfigurasi privasi dan keamanan pada akun peranti digital yang perlu dilakukan ialah seperti berikut.



a. Kata Sandi yang Kuat

Gunakan kata sandi yang kuat yang terdiri atas kombinasi huruf, angka, dan karakter khusus. Jangan gunakan kata sandi yang mudah ditebak atau terlalu umum. Ubah kata sandi secara berkala. Pelajaran mengenai pembuatan kata sandi yang kuat telah ada pada pelajaran Informatika kelas VII.

b. Autentikasi Dua Faktor

Aktifkan autentikasi dua faktor (jika didukung). Ini menambah lapisan keamanan dengan memerlukan verifikasi tambahan selain kata sandi, seperti kode yang dikirim ke ponsel melalui SMS atau dihasilkan oleh aplikasi otentikator. Hal ini telah dibahas di sub-bab sebelumnya.

c. Pengaturan Privasi

Periksa dan atur pengaturan privasi akunmu. Tentukan siapa yang dapat melihat informasi kamu, seperti status, foto profil, atau postingan. Sesuaikan dengan tingkat kenyamananmu.

d. Notifikasi Keamanan

Aktifkan notifikasi keamanan untuk mendapatkan pemberitahuan jika ada aktivitas yang mencurigakan atau masuk dari lokasi yang tidak biasa.

e. Pengaturan Aplikasi dan Permainan

Periksa dan kelola izin aplikasi pihak ketiga yang memiliki akses ke akunmu. Batasi izin hanya untuk aplikasi yang benar-benar diperlukan.

f. Pembaruan Perangkat Lunak

Pastikan perangkat lunak aplikasi dan sistem operasi kamu diperbarui secara berkala untuk mendapatkan pembaruan keamanan terbaru.

g. Keluar dari Sesi

Pastikan untuk keluar dari sesi akun di perangkat umum atau setelah menggunakan komputer bersama, seperti akun surelmu di komputer milik sekolah, atau saat menggunakan komputer di tempat umum.



h. Backup Data

Lakukan *backup* data secara teratur, terutama jika platform menyimpan informasi yang penting bagi kamu.

i. Peninjauan Aplikasi dan Integrasi Pihak Ketiga

Tinjau aplikasi dan integrasi pihak ketiga yang terhubung ke akun kamu dan hapus yang tidak diperlukan atau tidak dipercayai.

j. Pengaturan Lokasi

Jika platform menggunakan fitur pelacakan lokasi, pertimbangkan untuk membatasinya atau menonaktifkannya jika tidak diperlukan.

Terapkan langkah-langkah ini sesuai dengan kebutuhan dan spesifik platform digital yang kamu gunakan. Selalu perhatikan pembaruan dan informasi keamanan terbaru untuk menjaga akun Kamu tetap aman.



Aktivitas 10.4-07-P

Evaluasi Konfigurasi Privasi dan Keamanan pada Peranti Digitalmu

Aktivitas ini mengajak kamu untuk bernalar kritis dengan mengevaluasi hal-hal yang sudah kamu lakukan sehubungan dengan konfigurasi privasi dan keamanan data pada peranti digitalmu.

Langkah-Langkah

Isilah tabel berikut, yang berisi daftar akun peranti digital yang kamu pakai, serta konfigurasi privasi dan keamanannya. Gunakan tanda ✓ jika sudah menerapkan konfigurasi privasi dan keamanan, dan tanda × jika belum.

Akun Peranti Digital	Konfigurasi Privasi dan Keamanan									
	1	2	3	4	5	6	7	8	9	10
Akun Google										
Akun Apple										
Akun Microsoft										
Game <i>online</i>										
Aplikasi lainnya, sebutkan										



Cocokkanlah istilah dan pengertiannya.

1.	Autentikasi	a	Antivirus
2.	<i>Firewall</i>	b	Informasi hanya dapat diakses oleh pihak yang sah dan berwenang.
3.	Enkripsi	c	Metode keamanan yang mengharuskan pengguna untuk memberikan lebih dari satu bentuk identifikasi untuk memverifikasi identitasnya.
4.	Autentikasi Multifaktor	d	Mengubah informasi atau data menjadi format yang tidak dapat dibaca atau sulit dipahami.
5.	Filter Konten Negatif	e	<i>SafeSearch</i>
6.	Manajer Sandi	f	Hak akses atau izin sesuai dengan tingkat hak akses yang diberikan.
7.	<i>Availability</i>	g	Proses verifikasi identitas pengguna.
8.	<i>Integrity</i>	h	Password
9.	<i>Otorisasi</i>	i	Informasi tetap utuh dan tidak berubah tanpa izin.
10.	<i>Confidentiality</i>	j	Informasi selalu tersedia untuk pihak yang berwenang ketika diperlukan.
		k	Perangkat lunak atau perangkat keras yang dirancang untuk mencegah aliran data yang masuk.

Uraian

1. Kalau dilihat dari sejarahnya, mengapa banyak kerawanan yang ada di internet?
2. Mengapa sistem operasi memiliki banyak celah keamanan? Sebutkan paling tidak tiga alasan.



3. Mengapa *cookie* sangat bermanfaat bagi situs web? Sebutkan minimal tiga kegunaannya.
4. Kamu menggunakan situs *e-banking* yang harus melakukan *login* dengan sidik jari dan menggunakan pin yang dikirimkan ke SMS. Apakah cara *e-banking* tersebut telah masuk dengan kriteria autentikasi multifaktor? Jelaskan jawaban kamu.



Pengayaan

Jika ingin mengetahui lebih rinci keamanan informasi di internet, dapat mengakses informasi di link ini.

1. Keamanan internet: <https://buku.kemdikbud.go.id/s/po0qn1>
2. Windows Security: <https://buku.kemdikbud.go.id/s/wq8svd>



Refleksi

1. Setelah kamu mempelajari materi tentang keamanan, apakah kamu dapat memahaminya? Berapa persen derajat pemahaman kamu?
2. Setelah kamu mempelajari materi tentang keamanan informasi ini, manakah materi yang menurut kamu paling menarik? Mengapa materi tersebut menarik?
3. Apakah kamu pernah mengalami gangguan ketika menggunakan teknologi informasi di dunia maya seperti terkena virus, *malware*, atau menerima *phishing email*? Apa yang kamu lakukan pada saat itu? Apa yang kamu lakukan setelah mendapatkan materi keamanan informasi ini?
4. Apakah kamu tertarik menjadi penjaga keamanan internet di masa depan?

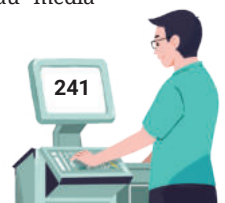


Glosarium

- abstraksi**
abstraction : proses memahami persoalan dengan berfokus pada ide utama/terpenting. Mengesampingkan hal rinci yang tidak relevan dan mengumpulkan hal yang relevan dalam suatu kesatuan; (produk); representasi baru dari suatu objek, sistem, atau masalah yang membingkai persoalan dengan menyembunyikan hal rinci yang tidak relevan.
- algoritma** *algorithm* : langkah-langkah dari proses untuk mencapai tujuan tertentu.
- alamat IP**
IP Address : sebuah identitas angka yang digunakan semua perangkat komputer agar saling berhubungan dalam jaringan internet.
- antrean**
queue : struktur data abstrak yang merupakan koleksi sekumpulan elemen, di mana penambahan elemen dilakukan di paling ujung, yaitu, pada “ekor” (*tail*), dan penghapusan elemen dilakukan pada titik pelayanan yang disebut “*head*”. Oleh karena itu, antrean juga disebut sebagai *first in first out* (FIFO).
- aplikasi**
application/ apps : jenis aplikasi perangkat lunak yang dirancang untuk dapat dijalankan pada *mobile device*, seperti *smartphone* atau tablet. Apps disebut juga *mobile apps*.
- aplikasi perkantoran**
office application : suatu paket aplikasi yang semula digunakan untuk perkantoran (pekerjaan menyiapkan dokumen yang zaman sebelum komputer memakai mesin ketik), yang oleh microsoft disebut *office application*.
- artefak komputasional**
computational artifact : objek apa pun yang dikembangkan oleh manusia dengan menggunakan proses berpikir komputasional dan peralatan komputer. Artefak komputasional dapat berupa (walaupun tidak terbatas): program, *image*, audio, video, *presentation*, atau *web page* (College Board, 2016);
- otentikasi**
authentication : proses untuk memastikan identitas pengguna yang mencoba mengakses suatu sistem atau layanan.
- otentikasi ganda** : juga dikenal sebagai verifikasi dua langkah: metode keamanan tambahan yang memerlukan dua bukti identitas untuk memverifikasi pengguna yang mencoba mengakses suatu akun atau layanan. Hal ini bertujuan untuk meningkatkan keamanan dengan membuatnya lebih sulit bagi penjahat *cyber* untuk masuk ke akun seseorang, bahkan jika mereka mengetahui kata sandi seseorang.
- otentikasi multifaktor** : juga dikenal sebagai verifikasi multi-langkah: metode keamanan yang memerlukan dua atau lebih bukti identitas untuk memverifikasi pengguna yang mencoba mengakses suatu akun atau layanan.
- badan program**
body program/ program definition : realisasi dari fungsi, kode program yang dituliskan di dalam sebuah fungsi.
- bahasa pemrograman**
programming language : kumpulan perintah, instruksi, dan sintaks lain yang digunakan untuk membuat suatu program.
- berpikir komputasional**
computational thinking : kemampuan manusia untuk memformulasikan masalah sehingga dapat dibuat penyelesaian yang diwujudkan dengan langkah-langkah komputasional/algoritma yang akan dieksekusi komputer (Lee, 2016); proses berpikir untuk mewujudkan solusi masalah dalam bentuk langkah-langkah komputasional atau algoritma yang dapat dieksekusi oleh komputer.



blok <i>block</i>	: sekumpulan kalimat C yang ditulis di antara { dan }
budaya <i>culture</i>	: lembaga manusia yang diwujudkan dalam perilaku orang yang dipelajari, termasuk sistem kepercayaan, bahasa, hubungan sosial, teknologi, lembaga, organisasi, dan sistem untuk menggunakan dan mengembangkan sumber daya.
bug <i>bug</i>	: <i>error</i> dalam program perangkat lunak yang dapat menyebabkan program berhenti atau memiliki perilaku yang tidak diinginkan; [Tech Terms] proses untuk menemukan dan mengoreksi <i>error</i> disebut dengan <i>debugging</i> [Wikipedia]
<i>Central Processing Unit (CPU)</i>	: peralatan dalam komputer yang mengeksekusi instruksi
diagram alir <i>flowchart</i>	: sebuah bagan atau diagram dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail serta hubungan antarproses.
data	: informasi yang dikumpulkan dan digunakan untuk referensi atau keperluan analisis; data dapat digital atau nondigital dan dapat dalam berbagai bentuk, termasuk angka, teks, gambar, suara, atau video.
<i>debugging</i>	: proses menemukan dan mengoreksi kesalahan (<i>bug</i>) dalam program.
dekomposisi <i>decomposition</i>	: decompose: untuk dipecah menjadi beberapa komponen; dekomposisi: menyelesaikan masalah atau sistem menjadi beberapa komponen.
ekosistem periksa data	: mengacu pada jaringan individu, organisasi, dan teknologi yang bekerja sama untuk memverifikasi dan memvalidasi informasi. Ekosistem ini memainkan peran penting dalam memerangi misinformasi dan memastikan akses ke informasi yang akurat dan andal.
ekspresi <i>expression</i>	: komponen bahasa pemrograman yang mendefinisikan suatu proses perhitungan/komputasi.
enkripsi <i>encryption</i>	: konversi data elektronik ke dalam bentuk lain yang disebut <i>ciphertext</i> , yang tidak dapat dengan mudah dipahami oleh siapa pun kecuali pihak yang berwenang.
<i>executable code</i>	: berkas yang saat dimuat ke dalam memori dapat langsung dieksekusi oleh komputer.
<i>firewall</i>	: perangkat lunak yang digunakan untuk menjaga keamanan jaringan pribadi. <i>Firewall</i> memblokir akses tidak sah ke atau dari jaringan pribadi dan sering digunakan untuk mencegah pengguna web yang tidak sah atau perangkat lunak terlarang mendapatkan akses ke jaringan pribadi yang terhubung ke internet.
<i>General Packet Radio Service (GPRS)</i>	: suatu teknologi yang memungkinkan pengiriman dan penerimaan data lebih cepat dibandingkan dengan penggunaan teknologi <i>Circuit Switch Data</i> atau CSD.
<i>header file</i>	: berisi deklarasi fungsi C dan definisi makro untuk dibagikan antara beberapa <i>file</i> sumber.
identifier <i>identifier</i>	: sekumpulan karakter yang digunakan untuk mengidentifikasi elemen-elemen dari program, seperti variabel dan fungsi.
integer	: tipe data yang merepresentasikan bilangan bulat.
internet <i>internet</i>	: jaringan komputer global yang koneksinya menggunakan protokol bersama (dalam hal struktur dan bahasa untuk permintaan <i>file</i> antara klien dan server) untuk berkomunikasi.
jaringan <i>network</i>	: sekelompok perangkat komputasi (komputer pribadi, telepon, server, sakelar, <i>router</i> , dll.) yang dihubungkan dengan kabel atau media nirkabel untuk pertukaran informasi dan sumber daya.



kata sandi <i>password</i>	: kumpulan karakter atau <i>string</i> yang digunakan oleh pengguna aplikasi, jaringan, atau sistem operasi yang mendukung banyak pengguna (<i>multiuser</i>) untuk memverifikasi identitas dirinya untuk keamanan aplikasi, jaringan atau sistem tersebut.
keluaran <i>output</i>	: hasil yang diperoleh dari suatu program yang berjalan yang dikirimkan ke luar dari program, misalnya kepada manusia atau program lainnya.
kesalahan sintaks <i>syntax error</i>	: kesalahan pada pemrograman yang terjadi ketika kode yang ditulis melanggar aturan sintaks dari suatu bahasa pemrograman; kesalahan ini dapat dideteksi oleh kompilator.
kesalahan semantik <i>semantic error</i>	: kesalahan pada pemrograman karena kesalahpahaman pada makna atau efek dari suatu elemen pada bahasa pemrograman.
kode sumber <i>source code</i>	: bentuk program yang diberikan kepada kompilator untuk dikonversi menjadi <i>object code</i> .
kode <i>code</i>	: kumpulan instruksi yang ditulis dalam bahasa pemrograman; <i>Koding/Coding</i> : Aksi untuk menulis program komputer dengan menggunakan bahasa pemrograman.
komputasional <i>computational</i>	: pendekatan atau metode yang berhubungan dengan komputer.
komputasi <i>computation</i>	: proses atau tindakan melakukan perhitungan atau pemrosesan data menggunakan komputer.
komputer <i>computer</i>	: mesin atau perangkat yang menjalankan proses, kalkulasi, dan operasi berdasarkan instruksi yang diberikan oleh program perangkat lunak atau perangkat keras [Techopedia].
kondisional <i>conditional</i>	: fitur bahasa pemrograman yang melakukan komputasi atau tindakan berbeda bergantung pada apakah kondisi Boolean yang dievaluasi bernilai benar atau salah; kondisional dapat merujuk ke pernyataan bersyarat, ekspresi bersyarat, atau konstruksi bersyarat.
koneksi <i>connection</i>	: hubungan fisik atau nirkabel antara beberapa sistem komputasi, komputer, atau perangkat komputasi.
konstanta <i>constant</i>	: suatu nilai yang tidak berubah
literasi digital	: kemampuan seseorang untuk menggunakan dan berinteraksi dengan media digital, alat komunikasi, atau jaringan.
lisensi	: izin atau persetujuan yang diberikan oleh pemilik hak atas suatu karya atau barang kepada pihak lain untuk menggunakan, menyebarkan, atau melakukan tindakan tertentu terhadap karya atau barang tersebut.
masukan <i>input</i>	: data yang dimasukkan ke dalam program untuk diproses lebih lanjut.
memori <i>memory</i>	: media penyimpanan data pada komputer.
<i>mnemonic</i>	: peranti seperti pola huruf, ide, atau asosiasi yang membantu dalam mengingat sesuatu.
membaca lateral	: teknik membaca yang bertujuan untuk mengevaluasi kredibilitas informasi, khususnya di era digital yang banjir informasi. Berbeda dengan membaca teks biasa secara linear dari atas ke bawah, membaca lateral melibatkan langkah keluar dari sumber informasi awal dan mencari informasi tambahan tentang sumber tersebut.
memori <i>memory</i>	: ruang penyimpanan fisik dalam perangkat komputasi, di mana data akan disimpan dan diproses dan instruksi yang diperlukan untuk pemrosesan juga disimpan.



<i>object code</i>	: berkas yang dihasilkan oleh kompilator
operasi biner	: suatu operasi yang melibatkan dua buah operan.
operasi uner	: operasi yang hanya melibatkan satu buah operan.
operator	: suatu fungsi yang dilakukan pada suatu operan.
operator aritmatika	: jenis operator yang digunakan untuk melambangkan suatu operasi aritmatika, seperti tambah, kurang, kali, dan bagi.
operator decrement	: operator yang digunakan untuk mengurangi nilai dari suatu variabel, misalnya '-'.
operator logika	: operator yang digunakan untuk menghasilkan nilai benar atau salah (boolean).
operator majemuk compound operator	: bentuk singkat penulisan operator penugasan, misalnya $x = x + 1$ ditulis $x += 1$.
otorisasi authorization	: proses keamanan yang menentukan akses pengguna ke sumber daya dalam suatu sistem setelah identitas mereka diverifikasi melalui proses autentikasi.
pengulangan loop	: struktur pemrograman yang mengulangi urutan instruksi selama kondisi tertentu benar.
penelusuran code tracing	: menginterpretasikan hasil dari setiap baris kode dan melacak dengan manual efek dari setiap pernyataan.
pengurutan sorting	: suatu jenis permasalahan pada komputasi untuk mengurutkan suatu himpunan objek berdasarkan kriteria tertentu.
penugasan assignment	: suatu pernyataan yang memberikan sebuah nilai pada suatu variabel.
perangkat keras hardware	: komponen fisik yang menyusun sistem komputasi, komputer, atau perangkat komputasi.
perangkat lunak software	: program yang berjalan di atas sistem komputasi, komputer, atau perangkat komputasi lainnya.
pernyataan statement	: unit bahasa pemrograman yang menyusun suatu program.
perulangan bersarang nested loop	: struktur kontrol perulangan yang berada di dalam struktur kontrol perulangan yang lain.
pembajakan piracy	: penyalinan, distribusi, atau penggunaan perangkat lunak ilegal.
pencarian searching	: suatu jenis permasalahan pada komputasi yang mencari suatu objek yang memenuhi kriteria tertentu dari sekumpulan objek.
prototip prototype	: dalam bahasa C, artinya deklarasi fungsi, menyatakan nama, tipe <i>return value</i> , nama dan tipe parameter formal (argumen).
pseudokode pseudocode	: deskripsi program informal yang tidak mengandung sintaks kode atau pertimbangan teknologi yang mendasari.
rekursi recursion	: suatu bentuk pendefinisian sebuah struktur yang mengandung dirinya sendiri. Fungsi/barisan rekursif adalah fungsi/barisan di mana nilainya tersebut ditentukan/tergantung dari nilai fungsi/barisan itu sendiri, pada urutan nilai-nilai sebelumnya.
SSL	: merupakan singkatan dari Secure Sockets Layer, suatu teknologi keamanan standar global yang memungkinkan komunikasi terenkripsi antara peramban web dan server web.



sistem operasi <i>operating system</i>	: perangkat lunak sistem yang mengatur sumber daya dari perangkat keras dan perangkat lunak, serta sebagai <i>daemon</i> untuk program komputer.
strategi <i>strategy</i>	: langkah terstruktur yang dilakukan untuk melakukan sesuatu. dalam hal ini, langkah tersebut digunakan untuk melakukan penebakan; strategi ini dapat ditulis juga sebagai sebuah algoritma.
simulasi <i>simulation</i>	: menyimulasikan: untuk meniru pengoperasian proses atau sistem di dunia nyata; tiruan operasi proses atau sistem dunia nyata
sistem komputer <i>computer system</i>	: pengaturan perangkat keras dan perangkat lunak lengkap dan fungsional dengan segala yang dibutuhkan untuk mengimplementasikan kinerja komputasi tertentu.
sistem operasi <i>operating-system</i>	: perangkat lunak sistem yang mengelola perangkat keras komputer, sumber daya perangkat lunak, dan menyediakan layanan umum untuk program komputer.
store (proses) <i>storage</i>	: suatu proses penyimpanan data digital dengan menggunakan teknologi komputasi. Penyimpanan ialah mekanisme yang memungkinkan komputer untuk menyimpan data, baik sementara maupun permanen.
struktur data <i>data structure</i>	: cara tertentu untuk menyimpan dan mengatur data dalam program komputer agar sesuai dengan tujuan tertentu sehingga dapat diakses dan dikerjakan dengan cara yang tepat; contoh struktur data termasuk <i>array</i> , <i>antrean</i> , <i>linked list</i> , <i>pohon</i> , dan <i>grafik</i> .
struktur kontrol	: blok kode yang menentukan urutan dan cara eksekusi program. Alat penting untuk pemrogram karena memungkinkan membuat program yang kompleks dan fleksibel.
tersarang <i>nested</i>	: struktur kontrol yang menjadi bagian dari suatu struktur kontrol lain.
tumpukan <i>stack</i>	: struktur konseptual yang terdiri atas sekumpulan elemen homogen dan didasarkan pada prinsip terakhir masuk pertama keluar (<i>last in first out - LIFO</i>).
unit kontrol <i>control unit</i>	: unit kontrol yang menangani semua sinyal kontrol prosesor.
unit logika aritmatika <i>arithmetic logic unit</i>	: komponen utama dari unit pengolah pusat dari sistem komputer yang melakukan semua proses yang berkaitan dengan aritmatika dan operasi logika yang perlu dilakukan pada kata-kata instruksi.
unit pusat pengolahan pusat <i>central processing unit</i>	: unit yang melakukan sebagian besar pemrosesan di dalam komputer dengan memproses semua instruksi yang diterima oleh perangkat lunak yang berjalan pada PC dan oleh komponen perangkat keras lainnya, dan bertindak sebagai kalkulator yang kuat.
variabel <i>variable</i>	: nama simbolik yang digunakan untuk melacak nilai yang dapat berubah saat program berjalan.
web scraping	: teknik untuk mengumpulkan data dalam jumlah besar dari berbagai situs web secara otomatis.
wifi <i>wifi</i>	: sebuah teknologi yang menggunakan gelombang radio (secara nirkabel) melalui jaringan komputer untuk bertukar data, termasuk koneksi internet yang memiliki kecepatan tinggi.



Daftar Pustaka

- Aaron. "How the Internet Works in 5 Minutes". 19 Februari 2009. Diakses dari https://www.youtube.com/watch?v=7_LPdtKXPc.
- AlgoRythmics. "Bubble-sort with Hungarian ("Csángó") folk dance. 29 Maret 2011. Diakses dari <https://www.youtube.com/watch?v=lyZQPjUT5B4>
- AlgoRythmics. "Insert-sort with Romanian folk dance". 29 Maret 2011. Diakses dari <https://www.youtube.com/watch?v=ROaU379I3U>
- Australian Curriculum Assessment and Reporting Authority. Computational Thinking in Practice (n.d). Diakses dari <https://v8.australiancurriculum.edu.au/media/5908/computational-thinking-in-practice-parent-teacher-cards.pdf>.
- Baase, S., and Timothy M. Henry. *A Gift of Fire, A: Social, Legal, and Ethical Issues for Computing Technology, Fifth Edition*. Pearson, 2018.
- Brodsky, M., and Getz, K. *The Data Literacy Cookbook*. ACRL, 2021.
- Deitel, P. & Deitel, H. C. *How to Program Edisi ke-8*. Pearson, 2016.
- Denning, P. J. The profession of IT. ACM, 2001. Diakses dari <https://dl.acm.org/doi/10.1145/359205.359239>
- EdGlossary. "The Glossary of Education Reform for Journalists, Parents, and Community Members". 2014, diakses dari <https://www.edglossary.org/>
- Fay-Wolfe, V. "How Computers Work: The CPU and Memory". 2005. Diakses melalui <https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading04.htm> pada 10 November 2020.
- George, B. *Digital Planet: Tomorrow's Technology and You Edisi 10*. Penerbit Pearson, 2011.
- ISTE. "Computational thinking: A digital age skill for everyone". 4 Januari 2012. Diakses dari <https://www.youtube.com/watch?v=VFcUgSYyRPg>
- John, S. (n.d). "Condition Statement in Python". Diakses dari <https://realpython.com/python-conditional-statements/> pada 22 Desember 2020.
- Jobs ID. "Info Lowongan Kerja Terbaru dan Populer 2020". Diakses dari <http://jobs.id> pada tanggal 18 November 2020.
- Kemdikbud. (n.d). KBBI, Kamus Besar Bahasa Indonesia. Diakses dari <https://kbbi.kemdikbud.go.id> tanggal 28 November 2020
- K12CS. (n.d). "Computational Thinking". Diakses melalui <https://k12cs.org/computational-thinking/>
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J. & Werner, L.(2011). "Computational Thinking for Youth in Practice". ACM Inroads Vol.2 (1) 2011: p.35. <https://doi.org/10.1145/1929887.1929902>.
- PyData. (n.d). Pandas: Python Data Analysis Library. Diakses dari <https://pandas.pydata.org/>
- Ramos, L.P. "Python while Loops: Repeating Tasks Conditionally". Diakses dari <https://realpython.com/python-while-loop/> pada 22 Desember 2020
- Sachs JD. "From Millennium Development Goals to Sustainable Development Goals". The Lancet. Vol. 379 (9832) 2012: p. 2206-2211.



- Sergeyev, Yaroslav D. "Numerical infinities and infinitesimals: Methodology, applications, and repercussions on two Hilbert problems". EMS Surveys in Mathematical Sciences Vol. 4 No. 2 (2017): 219–320.
- Sidik, S. "Siapa Minat, RI Butuh 17 Juta Tenaga IT". CNBC Indonesia, 12 Maret 2019. Diakses melalui <https://www.cnbcindonesia.com/tech/20190312195140-37-60263/siapa-minat-ributuh-17-juta-tenaga-kerja-it> pada 18 Desember 2020.
- TOKI. Learning Center Tim Olimpiade Komputer Indonesia (TOKI): <http://tlx.toki.id> Diakses pada tanggal 25 Desember 2020.
- Wikipedia. "HTTPS". 2021. Diakses dari Wikipedia: <https://id.wikipedia.org/wiki/HTTPS> pada tanggal 20 Januari 2021.
- Wikipedia. "Internet Sehat dan Aman". 2021. Diakses dari https://id.wikipedia.org/wiki/Internet_Sehat_dan_Aman pada tanggal 20 Januari 2021.
- Wikipedia. "Jaringan Komputer". 2021. Diakses dari https://id.wikipedia.org/wiki/Jaringan_komputer pada tanggal 20 Januari 2021.
- Wikipedia. "Operating System". 2021. Diakses dari https://en.wikipedia.org/wiki/Operating_system pada tanggal 20 Desember 2020.
- Wikipedia. "Queue (abstract_data_type)". 2021. Diakses dari [https://en.wikipedia.org/wiki/Queue_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Queue_(abstract_data_type))
- Wikipedia. "Rekursi". 2021. Diakses dari wikipedia: <https://en.wikipedia.org/wiki/Recursion> diakses pada tanggal 11 Januari 2021.
- Wikipedia. "Search Algorithm". 2021. Diakses dari https://en.wikipedia.org/wiki/Search_algorithm pada tanggal 11 Januari 2021.
- Wikipedia. "Stack (abstract data type)". 2021. Diakses dari [https://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type)) pada tanggal 11 Januari 2021.
- Wikipedia. "Web Scraping". 2021. Diakses dari wikipedia: https://en.wikipedia.org/wiki/Web_scraping diakses pada tanggal 11 Januari 2021.
- _____. (n.d.). Khan Academy. "Binary Search". Di akses melalui <https://khanacademy.org/computing/computer-science/algorithms/binary-search> pada 23 November 2020.
- _____. (n.d.). Dataquest. "Tutorial: Web Scraping with Python using BeautifulSoup". Diakses dari <https://www.dataquest.io/blog/web-scraping-tutorial-python/> pada 18 Desember 2020.
- _____. (n.d.). Google Colaboratory. Welcome to Colaboratory. Diakses dari <https://colab.research.google.com/>.



Daftar Sumber Gambar

Gambar 3.3 https://en.wikipedia.org/wiki/File:Crystal_Project_computer.png

Gambar 3.4 https://en.wikipedia.org/wiki/File:Intel_NUC8.jpg

Gambar 3.5 <https://www.computerhistory.org/collections/catalog/102691309>

Gambar 3.7 [https://en.wikipedia.org/wiki/File:Sierra_Supercomputer_\(48002385338\).jpg](https://en.wikipedia.org/wiki/File:Sierra_Supercomputer_(48002385338).jpg)

Indeks

A

algoritma 29, 30, 31, 32, 33, 36, 39, 43, 44,
53, 61, 79, 80, 83, 86, 87, 88, 89, 91, 94,
95, 106, 107, 108, 110, 121, 122, 123, 124,
206, 207, 240, 244

antivirus 204, 207, 208, 220, 224

antrean 96, 97, 98, 109, 122, 123, 124, 125,
240, 244

aplikasi 9, 82, 99, 100, 115, 116, 120, 121, 136,
137, 138, 139, 140, 141, 142, 143, 144,
145, 146, 155, 158, 160, 173, 177, 183,
184, 189, 190, 204, 205, 206, 207, 208,
209, 210, 213, 215, 220, 221, 225, 232,
233, 235, 236, 237, 240, 242

aplikasi terpercaya 210

arduino 116

artefak komputasional 190

assignment 37, 40, 50, 243

autentikasi 188, 210, 211, 212, 213, 214, 215,
219, 220, 232, 236, 239, 240, 243

autentikasi multifaktor 219, 239

B

bahasa pemrograman 8, 9, 10, 39, 42, 45, 49,
51, 53, 54, 59, 70, 78, 105, 106, 107, 115,
128, 136, 240, 241, 242, 243

berpikir komputasional iv, 29, 33, 240

biometrik 212, 213, 214, 219

bluetooth 197, 200, 201

bubble sort 83, 108

budaya digital 112

C

case 47, 57, 58

coding 45, 46

cookie 223, 224, 225, 239

cybersecurity 191

D

daftar isi 148, 150, 151, 152

data 2, 3, 4, 7, 8, 9, 12, 13, 14, 15, 16, 18, 19,
20, 21, 23, 24, 25, 26, 28, 29, 31, 32, 36,
37, 47, 63, 65, 71, 74, 75, 83, 91, 96, 97,
98, 100, 105, 107, 108, 109, 110, 114, 117,
119, 122, 126, 127, 128, 129, 130, 131,
132, 134, 135, 136, 137, 139, 140, 141,
143, 144, 145, 146, 148, 149, 150, 167,
170, 173, 174, 176, 181, 184, 189, 190,
191, 192, 194, 197, 199, 200, 201, 202,
204, 205, 206, 207, 208, 212, 213, 215,
219, 220, 222, 223, 224, 237, 238, 240,
241, 242, 244, 246, 250

decrement 48, 243

dekripsi 205, 206, 207

deskripsi soal 59, 60

diagram alir 28, 29, 30, 31, 32, 34, 35, 36, 37,
38, 39, 40, 43, 44, 51, 52, 59, 61, 62, 64,
65, 70, 74, 78, 105, 106, 241

digitalisasi 112

driver 115

E

ekspresi 28, 32, 37, 38, 45, 47, 48, 49, 50, 51,
53, 54, 64, 67, 68, 106, 133, 241, 242

enkripsi 204, 205, 206, 207, 221, 222, 241

excel 137, 142

F

firewall 204, 221, 241

for loop 63

format keluaran 46, 47, 107

format masukan 46, 47, 107

fungsi iii, 9, 73, 74, 75, 76, 77, 78, 79, 105, 113,
115, 120, 121, 137, 143, 158, 165, 183,
194, 240, 241, 243

fungsi rekursif 77, 79

H

hak atas kekayaan intelektual 112

hardware 114, 115, 120, 243

https ii, 8, 9, 14, 20, 21, 25, 26, 109, 116, 151,
166, 185, 186, 206, 239, 245, 246, 247

I

if-else 54, 56

increment 48

informasi iv, 1, 2, 3, 4, 6, 7, 11, 16, 17, 18, 19,
20, 22, 23, 24, 26, 29, 31, 47, 74, 80, 81,
111, 114, 118, 155, 156, 157, 158, 164,
167, 169, 170, 171, 172, 173, 174, 178,
181, 182, 184, 185, 187, 188, 189, 190,
191, 192, 194, 204, 205, 206, 207, 210,
211, 212, 213, 214, 215, 220, 221, 223,
226, 228, 235, 236, 237, 238, 239, 241, 242

insertion sort 83, 86, 87, 88

instruction set 127

integritas 169, 170, 191, 204, 208, 228

internet 3, 14, 15, 16, 19, 25, 81, 98, 154, 155,
175, 179, 182, 185, 188, 189, 190, 191,
192, 193, 194, 195, 196, 197, 198, 199,
200, 202, 203, 204, 205, 208, 210, 211,
215, 220, 221, 222, 223, 224, 226, 227,
228, 229, 230, 232, 235, 238, 239, 240,
241, 244

J

jaringan komputer dan internet 192, 204

jaringan lokal 156, 193, 194, 195, 200

K

karir 6, 8, 11, 12, 13, 113

kata sandi 188, 189, 204, 206, 211, 212, 213,
214, 215, 219, 232, 233, 236, 240, 242

keamanan informasi 185, 188, 189, 190, 191,
192, 204, 211, 220, 221, 239

kerahasiaan 191, 204

ketersediaan 191, 192, 204

komputer mikro 116, 117

konfigurasi keamanan 188

L

lisensi 113, 161, 162, 163, 164, 165, 166, 179,
184, 186, 242

lisensi gratis 162, 165

lisensi komersial 162, 165

literasi digital 242

lokasi memori 107, 128

M

mail merge 113, 148, 183

malware 204, 208, 209, 220, 221, 222, 224,
239

memanggil fungsi 75, 77

membaca lateral 2, 3, 19, 23, 24, 242

membuat fungsi 75, 76

memori 77, 107, 115, 118, 126, 127, 128, 129,
130, 131, 132, 134, 135, 209, 221, 241, 242

microprocessor 116

multitasking 120, 121, 124

O

open source 163, 165, 166, 184

operand 47, 48, 50

operator 5, 47, 48, 49, 50, 53, 106, 171, 243

operator logika 48, 50, 53, 243

P

parsing 10, 11, 12

pemrograman 7, 9, 10, 29, 30, 39, 40, 42, 45,
46, 47, 48, 49, 51, 53, 54, 59, 62, 70, 74, 78,
89, 95, 105, 106, 107, 115, 128, 136, 140,
167, 175, 240, 241, 242, 243

pencarian 2, 3, 4, 5, 6, 7, 8, 12, 16, 18, 22, 25,
80, 81, 92, 94, 108, 166, 179, 223, 229, 243

penelusuran 30, 36, 37, 38, 39, 42, 50, 54, 64,
66, 67, 69, 243

pengulangan 107, 243



pengurutan 82, 83, 85, 86, 88, 92, 95, 96, 105, 108, 243

perangkat keras iv, 112, 113, 114, 115, 116, 120, 156, 164, 169, 176, 183, 185, 186, 228, 238, 242, 243, 244

perangkat lunak iv, 112, 113, 114, 115, 116, 120, 136, 151, 156, 158, 160, 161, 162, 163, 164, 165, 166, 168, 169, 170, 172, 173, 174, 175, 176, 179, 184, 185, 186, 204, 205, 207, 208, 209, 210, 211, 213, 220, 221, 228, 229, 232, 236, 240, 241, 242, 243, 244, 256

perangkat lunak perkantoran 113, 136, 176, 184, 185

percabangan 28, 31, 33, 34, 40, 45, 53, 54, 55, 57, 58, 62, 69, 70, 106

percabangan bersarang 58

pop 138

profesi 112, 155, 160, 167, 169, 170, 171, 176, 184

pseudocode 29, 243

python 14, 25, 245, 246

Q

queue 96, 98, 99, 100, 101, 103, 104, 105, 109, 240

R

raspberry 116

register 127, 129, 130, 131, 132, 133, 134, 183

S

scheduler 122, 123, 124, 125

searching 243

selection sort 83, 89

single board circuit 116

sistem komputer 112, 113, 114, 115, 185, 186, 197, 209, 244

sistem operasi 66, 74, 113, 114, 115, 120, 121, 122, 125, 162, 163, 183, 190, 204, 209, 210, 220, 221, 229, 230, 236, 238, 242, 244

software 114, 115, 120, 172, 243

sorting 82, 109, 243

stack 96, 98, 99, 100, 101, 102, 105, 109, 244

struktur data 28, 29, 65, 96, 108, 109, 110, 240, 244

struktur kontrol 40, 45, 53, 54, 55, 58, 62, 67, 69, 106, 107, 108, 243, 244

switch 57, 58, 202, 203

switch-case 57, 58

T

teknologi iv, 14, 43, 112, 113, 154, 155, 156, 157, 158, 167, 169, 170, 171, 172, 173, 174, 175, 177, 179, 186, 189, 192, 196, 199, 202, 205, 207, 210, 214, 219, 220, 221, 228, 239, 241, 243, 244

tipe data 74, 75, 107, 108, 241

tracing 30, 36, 243

trusted application 207, 210

tumpukan 96, 97, 98, 102, 244

U

unit kontrol 244

unplugged iv, 7, 14, 215

V

variabel 2, 3, 4, 5, 6, 7, 16, 22, 32, 36, 37, 38, 45, 48, 49, 50, 58, 63, 64, 65, 77, 107, 108, 130, 241, 243, 244

variabel lokal 77

variabel pencarian 2, 3, 22

virus 204, 207, 208, 209, 210, 220, 221, 222, 239

W

web scraping 2, 25, 26, 244

while loop 63

windows 239

word 42, 137, 141



Profil Pelaku Perbukuan

Penulis

Dr. Mushthofa, S.Kom, M.Sc

Email : mush@apps.ipb.ac.id
Instansi : IPB University
Alamat Instansi : Kampus IPB Dramaga, Bogor
Bidang Keahlian : Ilmu komputer, *artificial intelligence*,
bioinformatika



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Dosen/peneliti di Departemen Ilmu Komputer IPB (2009 – sekarang)
2. Research Assistant di Ghent University, Belgium (2014 – 2018)

Riwayat Pendidikan dan Tahun Belajar

1. Sarjana Ilmu Komputer, IPB, Bogor (2005)
2. Master's in Computational Logic, TU Vienna, Austria (2009)
3. Doctor in Computer Science/ Bioinformatics, Ghent University, Belgia (2018)

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Fuzzy Answer Set Programming: From Theory to Practice (Chapter pada Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy etc. Methods and Their Applications (2020)
2. Network-based modelling of omics data (thesis), 2018

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

1. Comparison of TDNN and Factorized TDNN Approaches for Indonesian Speech Recognition, 2023, International Seminar on Intelligent Technology and Its Applications
2. Prediksi Interaksi Drug Target pada Gen Kanker Menggunakan Metode Lasso-XGBoost, 2023, Jurnal Teknologi Informasi dan Ilmu Komputer
3. Fuzzy Learning Vector Quantization for Classification of Mixed Meat Image Based on Character of Color and Texture, 2022, Jurnal RESTI
4. Large-Scale Oil Palm Trees Detection from High-Resolution Remote Sensing Images Using Deep Learning, 2022, Big Data and Cognitive Computing
5. Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen, 2019, Nature Communications
6. Quantitative analysis on porosity of reactive powder concrete based on automated analysis of back-scattered-electron images, 2019, Cement and Concrete Composite
7. Modelling multi-valued biological interaction networks using fuzzy answer set programming, 2018, Fuzzy Sets and Systems



Penulis

Auzi Asfarian, S.Komp., M.Kom.

Email : asfarian@apps.ipb.ac.id
Instansi : Dept. Ilmu Komputer, FMIPA, Institut
Pertanian Bogor
Alamat Instansi : Kampus IPB Dramaga, Bogor
Bidang Keahlian : Ilmu Komputer, Interaksi Manusia-Komputer



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Dosen, Ilmu Komputer, FMIPA, Institut Pertanian Bogor (2015 – sekarang).
2. Vice Chair, Indonesia ACM SIGCHI Chapter (2019 – sekarang).

Riwayat Pendidikan dan Tahun Belajar

1. Magister Ilmu Komputer, Institut Pertanian Bogor (2012 – 2014)
2. Sarjana Ilmu Komputer, Institut Pertanian Bogor (2007 – 2012)

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

1. Asfarian, A., Ibrahim, A., Anindhita, T., Nurhadryani, Y., Rum, S. N. M., & Nasharuddin, N. A. (2024). Towards IPB Immersive Field Trip Platform: Virtual Reality Field Trip to Introduce Digital Village Ecosystem to Computer Science Students. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 37(2), 82-93.
2. Rum, S. N. M., Mohamed, R., & Asfarian, A. (2024). Identifying Political Polarization in Social Media: A Literature Review. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 34(1), 80-89.
3. Ramadhan, D. A., Hidayati, S., Asfarian, A., & Ardiansyah, F. (2023). Bridging the Gap Between Education and Experience: A Comparative Analysis of IPB University Museum's Visitor Experience. *Khazanah al-Hikmah: Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan*, 11(1).
4. Integration of Nudging Techniques in the Context of Popular Digital Services to Increase Responsible Consumption Patterns in Society. 2023.
5. E-Initiative for Food Security: Design of Mobile Crowdfunding Platform to Reduce Food Insecurity in Indonesia Using The Wheel Methods – 2020
6. Thymun: Smart Mobile Health Platform for The Autoimmune Community to Improve the Health and Well-being of Autoimmune Sufferers in Indonesia – 2020



Penulis

Hanson Prihantoro Putro

Email : hanson @uii.ac.id
Instansi : Universitas Islam Indonesia
Alamat Instansi : Jl Kaliurang Km 14,5 Sleman Yogyakarta
Bidang Keahlian : Informatika/Rekayasa Perangkat Lunak



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Dosen Informatika, Universitas Islam Indonesia (2012 - sekarang)
2. Programmer, PT Lapi Divusi Bandung (2009 - 2011)

Riwayat Pendidikan dan Tahun Belajar

1. S2 Informatika, Institut Teknologi Bandung (2009 - 2011)
2. S1 Informatika, Institut Teknologi Bandung (2005 - 2009)

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Diktat Praktikum Pemrograman Berorientasi Obyek, Laboratorium Komputasi dan Sistem Cerdas, Jurusan Teknik Informatika, Universitas Islam Indonesia (2012).

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

1. Pengembangan Sistem Informasi Akademik MI-Gateway Berbasis Website, Kolokium Automata (2019).
2. Software Verification and Validation on Object Oriented Software Development Using Traceability Matrix, International Conference on Informatics and Computing (2018).

Penulis

Irya Wisnubhadra

Email : irya.wisnubhadra@uajy.ac.id
Instansi : Universitas Atma Jaya Yogyakarta
Alamat Instansi : Jl. Babarsari 44, Yogyakarta
Bidang Keahlian : Pemrograman, Database System, Business Intelligence



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Dosen Pengajar Tetap, Teknik Informatika Universitas Atma Jaya Yogyakarta

Riwayat Pendidikan dan Tahun Belajar

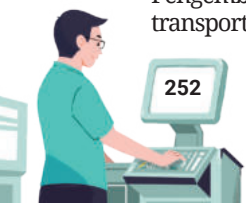
1. S1: Departemen Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada
2. S2: Teknik Informatika, Rekayasa Perangkat Lunak, Institut Teknologi Bandung
3. S3: Faculty of ICT, Universiti Teknikal Malaysia, Melaka

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Copyright Perangkat Lunak, BI-Transawit, 2021
2. Copyright Perangkat Lunak, LogTransawit, 2019
3. Jurnal - Publishing and Querying of Spatiotemporal Agriculture Production Data Warehouse on Semantic Web Using QB4MobOLAP, International Journal of Intelligent Engineering and Systems, 2023
4. Jurnal - Mobility Data Warehouse For Transportation of Oil Palm Fresh Fruit Bunches, ICIC Express Letters, Part B: Applications, 2022
5. Jurnal - Qb4mobolap: A vocabulary extension for mobility olap on the semantic web, Algorithms, 2021

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

Pengembangan mobility business intelligence untuk peningkatan produktivitas sistem transportasi TBS kelapa sawit secara berkelanjutan, Penelitian Terapan, Tahun 2020–2021, DIKTI



Penulis

Endang Purnama Giri, S.Kom., M.Kom.

Email : endang_pg@apps.ipb.ac.id

Instansi : Dept. Ilmu Komputer, FMIPA, IPB
University

Alamat Instansi : IPB University

Bidang Keahlian : Ilmu Komputer, Kriptografi, Machine learning



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Dosen, Ilmu Komputer, FMIPA, Institut Pertanian Bogor (2006 – sekarang)

Riwayat Pendidikan dan Tahun Belajar

1. Magister Ilmu Komputer, Universitas Indonesia (2006 – 2008)
2. Sarjana Ilmu Komputer, Institut Pertanian Bogor (2000 – 2004)

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

1. Perancangan Sistem Informasi Kuliner “Vokasi Laper” Berbasis Web dengan Metode Extreme Programming. Jurnal Rekayasa Sistem Informasi dan Teknologi (2024).
2. Penerapan Game-Based Retrospective Starfish pada Proses Scrum: Studi Kasus pada Pengembangan Aplikasi Zikirin. Prosiding Seminar Nasional Universitas Ma Chung (Informatika & Sistem Informasi; Bahasa dan Seni; Farmasi) (2023)
3. Perancangan Sistem Informasi Pemetaan Penjual Ternak Sapi dan Kambing di Kota Bogor Berbasis Website. Jurnal Rekayasa Sistem Informasi dan Teknologi (2023).
4. Exploring Computer Science Students’ Interest in the Role of Computer Science for Agriculture Digitalization: The Individual Interest Questionnaire Approach. Frontier in Sustainable Agro Maritime and Environmental Development Conference (2023)
5. Ekstraksi Fitur Rantai Markov untuk Klasifikasi Famili Protein. Jurnal Ilmiah SINUS (2023)
6. Ownership Protection on Digital Elevation Model (DEM) Using Transform-Based Watermarking. ISPRS International Journal of Geo-Information (2022)
7. Ischemic stroke identification based on EEG and EOG using ID convolutional neural network and batch normalization. 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (2016)
8. Pengembangan Model Pengenalan Wajah Manusia dengan Teknik Reduksi Dimensi Bi2D PCA dan Support Vector Machine sebagai Classifier. Ultimatics: Jurnal Teknik Informatika (2016)
9. Desain E-Voting Pilkada Kota Bogor Menggunakan Protokol Two Central Facilities yang Dimodifikasi. Krea-TIF: Jurnal Teknik Informatika (2015)



Penulis

Dean Apriana Ramadhan, S.Komp, M.Kom

Email : deanaprianaramadhan@apps.ipb.ac.id
Instansi : Departemen Ilmu Komputer, FMIPA IPB
Alamat Instansi : IPB University
Bidang Keahlian : Ilmu Komputer



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Dosen Departemen Ilmu Komputer IPB
2. Direktorat Sistem Informasi dan Transformasi Digital, IPB University

Riwayat Pendidikan dan Tahun Belajar

1. Sarjana Ilmu Komputer, Departemen Ilmu Komputer IPB. Lulus 2012
2. Magister Ilmu Komputer, Departemen Ilmu Komputer IPB. Lulus 2015

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

Ramadhan DA, Nurhadryani Y. Hermadi I. 2014. Campaign 2.0: Analysis Of Social Media Utilization In 2014 Jakarta Legislative Election. International Conference on Advanced Computer Science and Information Systems (JCACSIS) 2014. Jakarta (ID): Fasikom UI.

Penulis

Wahyono, Ph.D.

Email : wahyo@ugm.ac.id
Instansi : Universitas Gadjah Mada
Alamat Instansi : Sekip Utara Bulaksumur, Yogyakarta
Bidang Keahlian : Ilmu Komputer



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Staff Pengajar Prodi Ilmu Komputer, UGM, Yogyakarta (2012 – Sekarang)
2. Senior Developer, PT. Gamatechno Indonesia (2010-2012)

Riwayat Pendidikan dan Tahun Belajar

1. S3 Teknik Elektro, University of Ulsan, Korea (2012-2017)
2. S1 Ilmu Komputer, Universitas Gadjah Mada, Indonesia (2006-2010)

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Classification of Traffic Vehicle Density Using Deep Learning (Karya Ilmiah, 2020)
2. Perbandingan Perhitungan Jarak pada K-NN di Data Tekstual (Karya Ilmiah, 2020)

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

1. Pengembangan Sistem Surveilans Cerdas dan Terintegrasi Berbasis Kamera (2020)
2. Klasifikasi Tingkat Kepadatan Kendaraan Lalu Lintas Berbasis Convolutional Neural Network (2019)



Penelaah

Dr. Asep Wahyudin, S.Kom., M.T.

Email : away@upi.edu

Instansi : Universitas Pendidikan Indonesia

Alamat Instansi : Jl. Dr. Setiabudi No.229, Isola, Kec. Sukasari,
Kota Bandung, Jawa Barat 40154

Bidang Keahlian : Information System, Software Engineering, IS/ IT Strategic Plan,
Information Management, IS/IT Audit, IS/IT Governance, Business
Process Management



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Kepala KBK Software Engineering and Information Management Laboratory, Program Studi Ilmu Komputer UPI, 2021-Sekarang
2. Dosen Tetap Program S1 Studi Ilmu Komputer, 2006-sekarang
3. Dosen Tetap Program Studi S2 Pendidikan Ilmu Komputer, 2006-sekarang
4. Tim Pengembang Kurikulum Universitas Pendidikan Indonesia, 2021-sekarang
5. Asesor Lembaga Akreditasi Mandiri (LAM) INFOKOM, 2022-sekarang
6. Kepala Divisi Pengembangan Sistem Informasi pada Direktorat TIK UPI, 2007-2019

Riwayat Pendidikan dan Tahun Belajar

1. Ilmu Komputer, Universitas Indonesia, Program Doktorat S3, 2019
2. Teknik Informatika, Institut Teknologi Bandung, Program Magister S2, 2005
3. Sistem Informasi, STMIK Bandung, Program Sarjana S1, 2000

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Otomasi Perkantoran, Program Studi Manajemen Perkantoran S2 UPI
2. Modul Penerapan HOTS Dan TPACK (Technological Pedagogical Content Knowledge) Dalam Pembelajaran Informatika, Direktorat Jenderal Guru dan Tenaga Kependidikan, Kemendikbudristek
3. Pedoman Program Penguatan Profesional Kependidikan (P3K) Universitas Pendidikan Indonesia
4. Pedoman Pengembangan Kurikulum Program Studi di Lingkungan Universitas Pendidikan Indonesia
5. Panduan Pengakuan Pengalaman Belajar Mahasiswa Dalam Implementasi MBKM, Universitas Pendidikan Indonesia

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

1. The Development and Effectiveness of Business Strategy Model with A Partnership Approach for Growing Entrepreneurship, The 5th Global Conference on Business, Management and Entrepreneurship 2021



Penelaah

Leli Alhapip

Email : lelialthapip79@gmail.com
Instansi : Badan Riset dan Inovasi Nasional
Alamat Instansi : Jl. MH Thamrin No. 8 Jakarta
Bidang Keahlian : Teknologi Informasi



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. 2022 s.d. sekarang : Sebagai PNS di Badan Riset dan Inovasi Nasional, Jakarta
2. 2009 s.d. 2021 : Sebagai PNS, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, Jakarta

Riwayat Pendidikan dan Tahun Belajar

1. S1 Pendidikan Teknik Elektro, Universitas Pendidikan Indonesia Bandung, 2002
2. S2 Teknologi Informasi, Universitas Gajah Mada Yogyakarta, 2017
3. S3 Ilmu Komputer, Universitas Indonesia Depok

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Kajian Akademik Kurikulum Merdeka, 2024
2. Model Peningkatan Kompetensi Guru Melalui Program Peningkatan Kompetensi Guru Informatika SMA Fase E, 2023
3. Strategi Implementasi Kurikulum Merdeka pada Satuan Pendidikan, 2023
4. Kajian Akademik Kurikulum untuk Pemulihan Pembelajaran, 2022
5. Panduan Pembelajaran dan Asesmen, 2022
6. Panduan Pembelajaran dan Asesmen Kurikulum 2013, 2022
7. Naskah Rekomendasi Kebijakan PEMBANGUNAN MANUSIA INDONESIA 2045 Masukan *Background Studies* RPJPN 2025–2045: Aspek Agama, Pendidikan, dan Kebudayaan, 2022

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

1. Model pembelajaran efektif untuk sekolah dasar (SD) di Era digital 5.0 (studi kasus pada masa pandemic), DIPA Pusat Kurikulum dan Perbukuan 2021
2. Pengembangan Spektrum Keahlian dan Kurikulum Adaptif SMK Berbasis Revolusi Industri 4.0, DIPA Pusat Kurikulum dan Perbukuan 2020
3. Model penyelenggaraan sistem kredit semester di SMA/MA berbasis *blended learning* sebagai inovasi kurikulum diversifikasi, DIPA Pusat Kurikulum dan Pembelajaran 2019
4. Analisis komparatif dan eksperimen perangkat lunak kolaborasi daring untuk pengembangan dokumen publik pada instansi Pemerintah di Indonesia (Kasus di Pusat Kurikulum dan Perbukuan), Beasiswa Unggulan Tahun 2015



Ilustrator

Nana Maulana, S.Sn

Email : kartunaung@gmail.com
Instansi : Praktisi
Bidang Keahlian : Menulis, Ilustrasi, Komik, Karikatur,
Percetakan.



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Percetakan Citra Kharisma Bunda (2011)
2. Kartunis Penerbit Gerrmedia Komik (2011-2015)
3. Guru Seni dan Budaya SMA BINTARA DEPOK (2015-2018)
4. Owner Bikindong Creative Studio (2017-Sekarang)

Riwayat Pendidikan dan Tahun Belajar

1. S1 : Universitas Indraprasta PGRI (2007-2011)

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Buku Jakarta Penuh Hantu
2. Buku Komik Serba 90-an dalam Komik
3. Komik Siap Komandan

Informasi Lain

1. Juara 1 Lomba Komik Pembelajaran SD (2019)
2. Juara 2 Lomba Komik Polisi Nasional (2020)
3. Juara 2 Lomba Komik BCA (2018)
4. Juara 3 Lomba Komik PPATK (2020)
5. Juara 3 Lomba Komik Polisi Nasional (2019)
6. Juara 3 Lomba Komik Bank Indonesia (2021)
7. Juara Harapan Lomba Komik Jalur Rempah (2020)
8. Juara 1 Lomba Konten Kreatif HUT PT Pupuk Indonesia (2018)
9. Juara 1 Lomba Konten Kreatif HUT PT Pupuk Kujang (2019)
10. Juara 1 Lomba Komik BRI (2021)
11. Juara 1 Lomba Komik Silihtulungan Jawa Barat (2020)
12. Juara 2 Lomba Komik PPATK (2022)



Editor

Lenny Puspita Ekawaty

Email : lenny.ekawaty@gmail.com
Instansi : Pusat Perbukuan
Alamat Instansi : Jalan R.S. Fatmawati Gedung D Kompleks
Kemdikbudristek Cipete, Jakarta 12410
Bidang Keahlian : Penyuntingan



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Pusat Perbukuan, Badan Standar, Kurikulum, dan Asesmen Pendidikan, Kemdikbudristek (2021 s.d. sekarang)
2. Pusat Kurikulum dan Perbukuan, Badan Penelitian dan Pengembangan, Kemdikbudristek (2011 s.d. 2021)

Riwayat Pendidikan dan Tahun Belajar

1. Strata Dua (S2), Magister Manajemen Sistem Informasi, Universitas Gunadarma (2012 - 2015)
2. Strata Satu (S1), Teknik Komputer, Universitas Gunadarma (1993 - 1997)

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Tidak Bisa Tidak (2022)

Editor

Dr. Christina Tulalessy

Email : nonatula6@gmail.com
Instansi : Praktisi
Alamat Instansi : -
Bidang Keahlian : Penulisan BTP dan Penyuntingan



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Pusat Perbukuan (1988-2010)
2. Pusat Kurikulum dan Perbukuan, Badan Penelitian dan Pengembangan, Kemdikbudristek (2010–2021)

Riwayat Pendidikan dan Tahun Belajar

1. S3 Penelitian dan Evaluasi Pendidikan UNJ tahun 2017
2. S2 Penelitian dan Evaluasi Pendidikan UHAMKA tahun 2006
3. S1 Tata Busana IKIP Jakarta tahun 1988

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Penelitian Tindakan Kelas: Apa, Mengapa, Bagaimana: 2020

Informasi Lain dari Editor

1. Asesor Kompetensi Penulis dan Penyunting



Editor Visual

Is Yuniarto Nafawi

Email : isyuniarto@gmail.com
Instansi : Wind Rider Studio
Alamat Instansi : Krukah Selatan no 73, Surabaya
Bidang Keahlian : Ilustrasi, desain dan komik



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. Ilustrator dan komikus
2. Dosen honorer di Universitas Kristen Petra, Universitas Ciputra, iSTTS.
3. General Manager Bumilangit Comic Media

Riwayat Pendidikan dan Tahun Belajar

1. S1 jurusan Desain Komunikasi Visual, Universitas Kristen Petra. 1999-2003

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Grand Legend Ramayana – Reon Comics, 2013
2. Garudayana Saga vol 1-3 – CAB Publishing, 2014
3. Garudayana Saga vol 4-6 – CAB Publishing, 2015
4. Garudayana Saga Japan edition – Digital Catapult, 2015
5. Grand Legend Ramayana Japan edition – Digital Catapult, 2016
6. Garudayana Saga format scroll – Ciayo Comics, 2018
7. Gundala Son of Lightning – Line Webtoon2., 2019
8. Grand Legend Ramayana vol 5 – Reon Comics, 2020
9. Grand Legend Ramayana vol 6 – Reon Comics, 2022
10. Grand Legend Ramayana vol 7 – Reon Comics, 2023

Informasi Lain dari Editor Visual

Is Yuniarto adalah ilustrator dan komikus asal Surabaya, mengawali debut pada tahun 2005 dengan buku komik berjudul “Wind Rider” yang masuk dalam 3 nominasi Komikasia Award 2005 di kategori: Best Cover, Best Character, Best Comic. Is Yuniarto mendapat kesempatan bekerja sama dengan pihak Disney untuk membuat desain wayang kulit tokoh-tokoh Marvel Avengers dan memberikannya langsung sebagai cinderamata perwakilan Indonesia kepada sutradara dan aktor-aktor Avengers pada saat *event launching* film Avengers Infinity War pada tahun 2018 di Marina Bay Sand Singapura.



Desainer

Syndhi Renolarisa

Instansi : Praktisi
Alamat Instansi : Jakarta
Bidang Keahlian : Desain Grafis dan ilustrasi



Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. (2014–Sekarang) Freelance Ilustrator & Penata Letak/Desainer
2. (2014–2016) Quality Control & Internal Censorship, Transvision.
3. (2016–2023) Quality Control & Internal Censorship, UseeTV - Indihome.
4. (2023–sekarang) Digital Advertising & Financial Services Business Operation, Telkomsel.

Riwayat Pendidikan dan Tahun Belajar

1. S1 Jurusan Desain Komunikasi Visual, Sekolah Tinggi Media Komunikasi Trisakti, Jakarta (2014)

Judul Buku/Karya dan Tahun Terbit (10 Tahun Terakhir)

1. Desain isi Buku Panduan Guru dan Buku Siswa Buku Teks Pelajaran di Pusat Kurikulum dan Perbukuan (2014-sekarang)
2. Ilustrator Buku Panduan Guru dan Buku Siswa Buku Teks Pelajaran di Pusat Kurikulum dan Perbukuan (2014)

