

BUKU BAHAN AJAR SISWA

PEMROGRAMAN WEB DASAR

Program Keahlian Teknik Jaringan Komputer dan Telekomunikasi (TJKT)

Logika Pemrograman • HTML • CSS • JavaScript • Basis Data • Vibecoding

Disertai Contoh, Praktik, dan Relevansi Kehidupan Nyata

Untuk Jenjang Kelas X SMK

Yusuf Burhani,S.Pd., S.Kom

Kata Pengantar

Buku ini disusun sebagai bahan ajar bagi peserta didik mata pelajaran Pemrograman Web Dasar pada Program Keahlian Teknik Jaringan Komputer dan Telekomunikasi (TJKT) di Sekolah Menengah Kejuruan kelas X. Penyusunan materi mengacu pada referensi baku logika dan algoritma pemrograman serta referensi HTML komprehensif, dipadukan dengan pendekatan pembelajaran yang menekankan praktik langsung dan relevansi dengan kehidupan nyata.

Materi dalam buku ini disusun dalam lima bagian besar: logika pemrograman (konsep dasar, flowchart, percabangan, perulangan, variabel dan operator), dasar pengembangan web (HTML, CSS, JavaScript), praktik nyata (membangun landing page profil dan menyambungkannya ke database dengan form login), pengenalan vibecoding sebagai gambaran masa depan pemrograman berbantuan AI, serta proyek akhir yang mengintegrasikan seluruh kompetensi tersebut.

Setiap konsep logika pemrograman dijelaskan melalui analogi kehidupan sehari-hari — resep masakan, lampu lalu lintas, transaksi ATM — sebelum diterjemahkan menjadi kode program, dan setiap materi web langsung dipraktikkan menjadi bagian nyata dari sebuah website yang dibangun secara bertahap sepanjang buku ini. Pendekatan ini dimaksudkan agar peserta didik tidak sekadar menghafal sintaks, melainkan memahami esensi berpikir komputasional yang mendasarinya.

Semoga buku ini bermanfaat sebagai pegangan belajar yang sistematis, aplikatif, dan relevan dengan perkembangan dunia kerja di bidang teknologi informasi yang terus bertransformasi, termasuk kehadiran kecerdasan buatan sebagai mitra kerja baru bagi setiap pengembang perangkat lunak.

Penyusun

Daftar Isi

Kata Pengantar	2
Daftar Isi	3
Peta Kompetensi Buku	6
BAB 1 — Pendahuluan.....	7
1.1 Mengapa Teknisi Jaringan Perlu Memahami Pemrograman Web?	7
1.2 Gambaran Umum Materi	7
1.3 Perangkat Lunak dan Lingkungan Kerja.....	8
1.4 Cara Menggunakan Buku Ini	8
BAB 2 — Konsep Dasar Logika Pemrograman	10
2.1 Logika: Cara Berpikir Terstruktur	10
2.2 Algoritma: Resep untuk Menyelesaikan Masalah	10
2.3 Ciri-Ciri Algoritma yang Baik	10
2.4 Dari Aktivitas Sehari-hari Menuju Algoritma Formal	11
BAB 3 — Flowchart dan Pseudocode	13
3.1 Mengapa Perlu Merencanakan Sebelum Menulis Kode?	13
3.2 Simbol-Simbol Flowchart Standar	13
3.3 Contoh Flowchart: Menentukan Kelulusan Berdasarkan Nilai.....	13
3.4 Pseudocode: Kode Semu yang Mudah Dibaca	14
3.5 Latihan Konversi: Dari Algoritma ke Pseudocode.....	14
BAB 4 — Struktur Kontrol: Percabangan (Decision Making)	16
4.1 Percabangan dalam Kehidupan Sehari-hari	16
4.2 Struktur IF Tunggal.....	16
4.3 Struktur IF-ELSE.....	16
4.4 Struktur IF-ELSE IF Bertingkat	16
4.5 Operator Logika pada Kondisi Majemuk	17
BAB 5 — Struktur Kontrol: Perulangan (Looping)	19
5.1 Perulangan dalam Kehidupan Sehari-hari	19
5.2 Perulangan FOR: Ketika Jumlah Pengulangan Sudah Diketahui.....	19
5.3 Perulangan WHILE: Ketika Jumlah Pengulangan Belum Pasti	19
5.4 Perulangan Bersarang (Nested Loop)	20
5.5 Menghindari Perulangan Tak Berhingga (Infinite Loop).....	20
BAB 6 — Variabel, Tipe Data, dan Operator.....	22
6.1 Variabel: Kotak Penyimpan Data Bernama	22
6.2 Aturan Penamaan Variabel.....	22
6.3 Tipe Data Dasar.....	22
6.4 Operator Aritmatika	23
6.5 Operator Perbandingan dan Penugasan.....	23
BAB 7 — Pengenalan HTML: Struktur Dasar Dokumen Web	25

7.1 Apa Itu HTML dan Mengapa Penting?	25
7.2 Struktur Dasar Dokumen HTML	25
7.3 Tag Heading dan Paragraf	26
7.4 Tag Dasar Lainnya: Link, Gambar, dan List	26
7.5 Relevansi HTML dalam Kehidupan Nyata	27
BAB 8 — HTML Lanjutan: Tabel dan Form	28
8.1 Menyusun Tabel dengan HTML	28
8.2 Form HTML: Gerbang Interaksi Pengguna	29
8.3 Jenis-Jenis Input Form	29
8.4 Atribut Validasi Dasar pada Form	30
BAB 9 — Pengenalan CSS: Mempercantik Tampilan Web	32
9.1 Apa Itu CSS dan Tiga Cara Menerapkannya	32
9.2 Selector CSS: Menargetkan Elemen yang Ingin Distyling	32
9.3 Warna, Tipografi, dan Spacing	33
9.4 Dasar Layout dengan Flexbox	33
9.5 Relevansi CSS dalam Kehidupan Nyata	34
BAB 10 — Pengenalan JavaScript: Membuat Web Menjadi Interaktif	36
10.1 Dari Logika ke Web: Menyatukan Bab 2–6 dengan HTML/CSS	36
10.2 Menyisipkan JavaScript ke dalam HTML	36
10.3 DOM: Jembatan antara JavaScript dan HTML	36
10.4 Menangani Event Pengguna	37
10.5 Studi Kasus: Kalkulator Luas Sederhana	38
BAB 11 — Praktik: Membangun Landing Page Profil Sederhana	40
11.1 Apa Itu Landing Page dan Mengapa Penting?	40
11.2 Merancang Struktur Informasi (Wireframe Sederhana)	40
11.3 Membangun Struktur HTML Landing Page	41
11.4 Mempercantik dengan CSS	42
11.5 Menambahkan Interaktivitas dengan JavaScript	43
BAB 12 — Koneksi Database dan Form Login	45
12.1 Dari Website Statis Menuju Website Dinamis	45
12.2 Alur Kerja Website Dinamis	45
12.3 Membuat Database dan Tabel dengan MySQL	45
12.4 Menghubungkan PHP ke Database MySQL	46
12.5 Membangun Form Login Lengkap	46
12.6 Praktik Keamanan Dasar pada Form Login	47
BAB 13 — Pengenalan Vibecoding: Pemrograman Berbantuan AI	49
13.1 Apa Itu Vibecoding?	49
13.2 Mengapa Vibecoding Relevan bagi Siswa TJKT Saat Ini	49
13.3 Cara Berkomunikasi Efektif dengan Asisten AI	50

13.4 Sikap Kritis: AI sebagai Partner, Bukan Pengganti Pemahaman.....	50
13.5 Etika Penggunaan AI dalam Konteks Akademik	51
BAB 14 — Menyiapkan dan Menerbitkan Aplikasi Web di Ubuntu Server	52
14.1 Dari Localhost Menuju Server Sungguhan.....	52
14.2 Menyiapkan Server: LAMP Stack	52
14.3 Memverifikasi Instalasi.....	53
14.4 Memindahkan (Upload) Proyek ke Server.....	53
14.5 Mengonfigurasi Virtual Host.....	54
14.6 Menyambungkan Aplikasi ke Database di Server	54
14.7 Pengamanan Dasar Server Web	55
14.8 Troubleshooting Masalah Umum saat Deployment.....	55
BAB 15 — Rangkuman dan Proyek Akhir	58
14.1 Peta Keterkaitan Antar Bab	58
14.2 Rekapitulasi Capaian Pembelajaran	58
14.3 Proyek Akhir: Landing Page Profil dengan Sistem Login.....	59
14.4 Arah Pembelajaran Lanjutan	59
Glosarium	61
Lampiran: Referensi Cepat Sintaks JavaScript.....	63
Lampiran: Referensi Cepat Tag HTML dan Properti CSS.....	64
Tag HTML yang Sering Digunakan	64
Properti CSS yang Sering Digunakan	64
Lampiran: Proyek Studi Kasus Tambahan	65
Proyek A: Aplikasi Pengingat Tugas Sekolah (To-Do List).....	65
Proyek B: Kalkulator Nilai Rata-rata dan Predikat	65
Proyek C: Halaman Galeri Foto Kegiatan Sekolah	65
Lampiran: Ringkasan Perintah Deployment Ubuntu Server	66
Lampiran: Peta Jalan Belajar Mandiri Lanjutan.....	67
Lampiran: Checklist Proyek Akhir	68
Daftar Pustaka	69

Peta Kompetensi Buku

Tabel berikut memetakan setiap bab terhadap bagian besar dan fokus kompetensi dalam buku ini.

Bab	Judul	Bagian	Fokus Utama
1	Pendahuluan	-	Orientasi mata pelajaran dan lingkungan kerja
2	Konsep Dasar Logika Pemrograman	I. Logika	Logika dan algoritma dengan analogi kehidupan nyata
3	Flowchart dan Pseudocode	I. Logika	Perencanaan alur program sebelum menulis kode
4	Struktur Kontrol: Percabangan	I. Logika	if, if-else, if-else if, operator logika
5	Struktur Kontrol: Perulangan	I. Logika	for, while, nested loop
6	Variabel, Tipe Data, dan Operator	I. Logika	Fondasi penyimpanan dan manipulasi data
7	Pengenalan HTML	II. Web	Struktur dasar dokumen web
8	HTML Lanjutan	II. Web	Tabel dan form
9	Pengenalan CSS	II. Web	Styling, selector, dan layout Flexbox
10	Pengenalan JavaScript	II. Web	DOM dan event handling
11	Praktik: Landing Page	III. Praktik	Proyek web terintegrasi HTML+CSS+JS
12	Koneksi Database dan Login	III. Praktik	PHP, MySQL, dan keamanan dasar
13	Pengenalan Vibecoding	IV. Masa Depan	Kolaborasi efektif dengan AI
14	Setup Aplikasi Web di Ubuntu Server	III. Praktik	LAMP stack, virtual host, deployment, hardening dasar
15	Rangkuman dan Proyek Akhir	V. Penutup	Konsolidasi dan proyek integratif

BAB 1 — Pendahuluan

Pemrograman Web Dasar merupakan mata pelajaran kelas X pada Program Keahlian Teknik Jaringan Komputer dan Telekomunikasi (TJKT) yang memperkenalkan peserta didik pada dasar-dasar logika pemrograman serta pengembangan web menggunakan HTML, CSS, dan JavaScript. Mata pelajaran ini menjadi jembatan penting antara kompetensi jaringan yang menjadi fokus utama Program Keahlian TJKT dengan kemampuan dasar pemrograman yang semakin dibutuhkan oleh hampir seluruh profesi di bidang teknologi informasi, termasuk teknisi jaringan yang kini semakin sering dituntut memahami otomatisasi, dashboard monitoring berbasis web, dan konfigurasi perangkat melalui antarmuka web.

Buku ini disusun dengan filosofi 'belajar sambil mempraktikkan dan mengaitkan dengan kehidupan nyata'. Setiap konsep logika pemrograman dijelaskan melalui analogi sehari-hari sebelum diterjemahkan ke dalam kode, dan setiap materi HTML/CSS/JavaScript langsung dipraktikkan menjadi bagian nyata dari sebuah website, memuncak pada proyek akhir berupa landing page profil pribadi yang lengkap dengan form login dan koneksi database.

Tujuan Pembelajaran

- Peserta didik memahami kedudukan mata pelajaran Pemrograman Web Dasar dan relevansinya dengan Program Keahlian TJKT serta dunia kerja.
- Peserta didik memahami struktur dan alur belajar buku ini, dari logika pemrograman hingga proyek web nyata.
- Peserta didik mengenal perangkat lunak dan lingkungan kerja yang akan digunakan sepanjang pembelajaran.

1.1 Mengapa Teknisi Jaringan Perlu Memahami Pemrograman Web?

Banyak orang mengira bidang jaringan komputer sepenuhnya terpisah dari pemrograman, padahal kenyataannya sebagian besar perangkat jaringan modern — router, switch, access point, hingga sistem manajemen jaringan — dikonfigurasi dan dipantau melalui antarmuka berbasis web (web-based management interface). Seorang teknisi jaringan yang memahami dasar HTML, CSS, dan JavaScript akan jauh lebih percaya diri saat menghadapi dashboard konfigurasi perangkat, sistem monitoring jaringan berbasis web seperti Grafana atau LibreNMS, maupun ketika diminta membuat halaman status jaringan sederhana untuk timnya.

Lebih jauh, logika pemrograman yang dipelajari pada buku ini — cara berpikir sistematis, memecah masalah besar menjadi langkah-langkah kecil, dan menyusun alur keputusan — adalah keterampilan berpikir (computational thinking) yang bermanfaat di hampir semua aspek pekerjaan teknis, termasuk saat menyusun skrip otomatisasi jaringan atau troubleshooting yang runtut dan logis.

1.2 Gambaran Umum Materi

Bagian	Bab	Fokus Materi
I. Logika Pemrograman	2–6	Konsep logika, flowchart, pseudocode, percabangan, perulangan, variabel dan operator — seluruhnya dengan analogi kehidupan nyata
II. Dasar Web	7–10	HTML (struktur dokumen web), CSS (tampilan/styling), dan JavaScript (interaktivitas)
III. Praktik Nyata	11–12	Membangun landing page profil sederhana, menyambungkannya ke database, dan membuat form login
IV. Masa Depan Pemrograman	13	Pengenalan vibecoding — pemrograman berbantuan AI yang mengubah cara kerja pengembang modern
V. Penutup	14	Rangkuman dan proyek akhir terintegrasi

1.3 Perangkat Lunak dan Lingkungan Kerja

Seluruh praktik dalam buku ini dapat dikerjakan menggunakan perangkat lunak yang bersifat gratis dan ringan, sehingga dapat dijalankan pada komputer laboratorium sekolah dengan spesifikasi standar.

Kebutuhan	Rekomendasi	Fungsi
Code Editor	Visual Studio Code (VS Code)	Menulis dan mengedit kode HTML, CSS, JavaScript, dan PHP
Web Browser	Google Chrome / Mozilla Firefox	Menjalankan dan menguji hasil kode, dilengkapi Developer Tools untuk debugging
Server Lokal	XAMPP / Laragon	Menjalankan PHP dan MySQL secara lokal untuk praktik koneksi database pada Bab 12
Browser Extension	Live Server (ekstensi VS Code)	Melihat perubahan HTML/CSS secara langsung (live reload) tanpa perlu refresh manual

1.4 Cara Menggunakan Buku Ini

1. Kerjakan bab secara berurutan pada Bagian I (Bab 2–6) karena setiap konsep logika menjadi fondasi bagi konsep berikutnya, sebelum masuk ke Bagian II (Bab 7–10) yang memperkenalkan bahasa web secara bertahap.
2. Ketik ulang (jangan hanya membaca) setiap contoh kode pada buku ini di code editor masing-masing, karena keterampilan menulis kode terbentuk melalui praktik langsung, bukan sekadar membaca teori.

3. Kerjakan seluruh aktivitas pembelajaran di akhir bab sebelum melanjutkan ke bab berikutnya, karena banyak aktivitas yang hasilnya akan digunakan kembali pada proyek landing page di Bab 11.
4. Gunakan Bab 11 dan 12 sebagai proyek besar akhir semester yang mengintegrasikan seluruh kompetensi HTML, CSS, dan JavaScript yang telah dipelajari.

Rangkuman

Pemrograman Web Dasar membekali peserta didik TJKT dengan logika pemrograman dan kemampuan membangun web menggunakan HTML, CSS, dan JavaScript, keterampilan yang semakin relevan bagi teknisi jaringan modern. Buku ini disusun dalam lima bagian besar: logika pemrograman, dasar web, praktik nyata (landing page dan database), pengenalan vibecoding, dan proyek akhir, seluruhnya dikaitkan dengan konteks dan contoh kehidupan nyata.

BAB 2 — Konsep Dasar Logika Pemrograman

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan pengertian logika dan algoritma dalam konteks pemecahan masalah sehari-hari.
- Peserta didik mampu mengidentifikasi ciri-ciri algoritma yang baik.
- Peserta didik mampu menerjemahkan aktivitas sehari-hari menjadi langkah-langkah algoritmik yang runtut.

2.1 Logika: Cara Berpikir Terstruktur

Logika adalah cara berpikir yang runtut dan masuk akal untuk mengambil kesimpulan atau keputusan berdasarkan fakta dan aturan tertentu. Dalam konteks pemrograman, logika menjadi dasar bagaimana seorang programmer menyusun instruksi yang akan dijalankan komputer secara berurutan dan tepat, karena komputer — betapapun canggihnya — hanya dapat menjalankan instruksi persis seperti yang dituliskan, tanpa dapat menebak maksud programmer di baliknya.

Sebagai analogi sehari-hari, perhatikan bagaimana seseorang memutuskan apakah perlu membawa payung sebelum berangkat sekolah: 'Jika langit mendung ATAU ramalan cuaca menyebutkan hujan, maka bawa payung; jika tidak, maka tidak perlu membawa payung.' Kalimat ini sesungguhnya sudah merupakan sebuah pernyataan logika dengan struktur kondisional, persis seperti struktur percabangan (if-else) yang akan dipelajari pada Bab 4.

2.2 Algoritma: Resep untuk Menyelesaikan Masalah

Algoritma adalah urutan langkah-langkah logis dan sistematis untuk menyelesaikan suatu masalah atau mencapai tujuan tertentu. Analogi paling umum untuk memahami algoritma adalah resep masakan: sebuah resep menuliskan bahan-bahan yang diperlukan (input), langkah-langkah memasak secara berurutan (proses), dan hasil akhir berupa makanan yang siap disajikan (output). Algoritma dalam pemrograman bekerja dengan prinsip yang persis sama — data masukan diproses melalui serangkaian langkah untuk menghasilkan keluaran yang diinginkan.

Contoh algoritma sehari-hari lainnya adalah langkah-langkah menyeberang jalan dengan aman: (1) berhenti di tepi jalan, (2) tengok ke kanan dan ke kiri untuk memeriksa kendaraan, (3) jika jalan aman, mulai menyeberang; jika tidak, tunggu hingga aman, (4) menyeberang dengan tetap waspada, (5) sampai di seberang jalan dengan selamat. Perhatikan bahwa algoritma ini memiliki langkah kondisional (langkah 3) yang menentukan apakah proses menyeberang dapat dilanjutkan atau harus menunggu — pola yang akan sangat sering ditemui dalam pemrograman.

2.3 Ciri-Ciri Algoritma yang Baik

Ciri	Penjelasan	Contoh Analogi
Memiliki awal dan akhir yang jelas	Algoritma harus tahu kapan mulai dan kapan berhenti	Resep masakan berakhir saat makanan matang, bukan berlanjut tanpa henti
Setiap langkah jelas dan tidak ambigu	Instruksi tidak boleh menimbulkan lebih dari satu tafsiran	'Tambahkan gula secukupnya' bersifat ambigu, sedangkan '1 sendok makan gula' jelas dan pasti
Urutan langkah logis dan berurutan	Langkah berikutnya bergantung pada langkah sebelumnya secara masuk akal	Tidak mungkin menggoreng telur sebelum memecahkannya ke wajan
Menghasilkan output yang diharapkan	Algoritma yang benar harus mencapai tujuan yang dimaksud	Resep kue yang diikuti dengan benar akan menghasilkan kue, bukan sup
Efisien (jumlah langkah wajar)	Semakin sedikit langkah tanpa mengorbankan ketepatan, semakin baik	Rute tercepat ke sekolah lebih baik dibanding rute memutar tanpa alasan

2.4 Dari Aktivitas Sehari-hari Menuju Algoritma Formal

Kemampuan menerjemahkan aktivitas sehari-hari menjadi algoritma formal adalah keterampilan dasar yang akan terus digunakan sepanjang belajar pemrograman. Sebagai latihan, perhatikan bagaimana aktivitas 'membuat mi instan' diterjemahkan menjadi algoritma yang presisi.

5. Mulai.
6. Didihkan air sebanyak 400 ml dalam panci.
7. Masukkan mi ke dalam air mendidih.
8. Masak mi selama 3 menit.
9. Jika mi sudah lunak, tiriskan air; jika belum, tunggu hingga lunak.
10. Campurkan bumbu ke dalam mi yang telah ditiriskan.
11. Aduk hingga bumbu merata.
12. Sajikan mi ke dalam mangkuk.
13. Selesai.

Perhatikan bahwa algoritma di atas memiliki struktur sekuensial (langkah 1 hingga 4 berjalan berurutan tanpa cabang), struktur kondisional (langkah 5 memuat keputusan 'jika... maka...'), dan berpotensi memiliki struktur perulangan apabila proses menunggu pada langkah 5 diulang hingga kondisi terpenuhi. Ketiga struktur ini — sekuensial, kondisional, dan perulangan — adalah tiga struktur kontrol dasar yang membentuk hampir seluruh program komputer, dan akan dipelajari lebih mendalam pada Bab 4 dan Bab 5.

Aktivitas Pembelajaran

14. Peserta didik menuliskan algoritma langkah demi langkah untuk aktivitas sehari-hari pilihan mereka sendiri (misalnya membuat teh, mengisi pulsa, atau menyalakan komputer), kemudian menandai bagian mana yang bersifat sekuensial dan mana yang bersifat kondisional.
15. Peserta didik saling menukar algoritma yang telah dibuat dengan teman sebangku, lalu mencoba 'menjalankan' algoritma tersebut secara harfiah (persis seperti komputer menjalankan instruksi) untuk menemukan langkah yang ambigu atau kurang jelas.
16. Diskusi kelompok: identifikasi tiga contoh algoritma lain dalam kehidupan sehari-hari yang memiliki elemen kondisional (percabangan keputusan) yang jelas.

Rangkuman

Logika adalah cara berpikir runtut untuk mengambil keputusan, sedangkan algoritma adalah urutan langkah sistematis untuk menyelesaikan masalah, layaknya resep masakan. Algoritma yang baik memiliki awal-akhir yang jelas, langkah tidak ambigu, urutan logis, mencapai tujuan yang diharapkan, dan efisien. Tiga struktur kontrol dasar — sekuensial, kondisional, dan perulangan — akan menjadi fondasi seluruh materi logika pemrograman pada bab-bab berikutnya.

Soal Latihan / Refleksi

17. Jelaskan perbedaan antara logika dan algoritma, beserta hubungan keduanya.
18. Sebutkan lima ciri algoritma yang baik beserta contoh penerapannya masing-masing.
19. Tuliskan algoritma langkah demi langkah untuk aktivitas 'mengisi baterai ponsel hingga penuh', lengkap dengan elemen kondisionalnya.
20. Mengapa instruksi seperti 'masak sampai enak' dianggap sebagai instruksi yang buruk dalam sebuah algoritma?

BAB 3 — Flowchart dan Pseudocode

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan fungsi flowchart dan pseudocode dalam perencanaan program.
- Peserta didik mampu menggunakan simbol-simbol flowchart standar dengan benar.
- Peserta didik mampu menerjemahkan sebuah algoritma menjadi flowchart dan pseudocode.

3.1 Mengapa Perlu Merencanakan Sebelum Menulis Kode?

Sebelum seorang programmer menulis baris kode pertama, langkah yang sangat disarankan adalah merencanakan alur logika program terlebih dahulu menggunakan flowchart (diagram alir) atau pseudocode (kode semu). Analoginya serupa dengan seorang arsitek yang tidak langsung membangun rumah tanpa gambar cetak biru (blueprint) terlebih dahulu — perencanaan ini membantu menemukan kesalahan logika sejak dini, jauh lebih murah dan cepat diperbaiki dibanding ditemukan setelah ratusan baris kode ditulis.

3.2 Simbol-Simbol Flowchart Standar

Simbol	Bentuk	Fungsi
Terminator	Oval/elips	Menandai awal (Mulai) dan akhir (Selesai) sebuah program
Proses	Persegi panjang	Menunjukkan sebuah proses atau instruksi yang dijalankan
Keputusan (Decision)	Belah ketupat	Menunjukkan percabangan berdasarkan sebuah kondisi (Ya/Tidak)
Input/Output	Jajaran genjang	Menunjukkan proses memasukkan data (input) atau menampilkan hasil (output)
Garis Alir	Anak panah	Menunjukkan arah urutan proses yang dijalankan
Konektor	Lingkar kecil	Menghubungkan bagian flowchart yang terpisah halaman atau terlalu jauh

3.3 Contoh Flowchart: Menentukan Kelulusan Berdasarkan Nilai

Perhatikan studi kasus sederhana: sebuah program perlu menentukan apakah seorang siswa lulus atau tidak berdasarkan nilai ujian, dengan aturan nilai minimal kelulusan adalah 75. Alur logikanya digambarkan sebagai berikut dalam bentuk deskripsi berurutan simbol flowchart (Terminator → Input → Decision → Proses → Terminator).

21. [Terminator] Mulai
22. [Input/Output] Masukkan nilai ujian siswa

23. [Decision] Apakah nilai ≥ 75 ?
24. [Proses - jalur Ya] Tampilkan 'Lulus'
25. [Proses - jalur Tidak] Tampilkan 'Tidak Lulus'
26. [Terminator] Selesai

Kedua jalur dari simbol Decision (Ya dan Tidak) pada akhirnya bertemu kembali sebelum mencapai Terminator Selesai, mencerminkan bahwa program akan selalu berakhir dengan tepat setelah menampilkan salah satu dari dua kemungkinan hasil tersebut.

3.4 Pseudocode: Kode Semu yang Mudah Dibaca

Pseudocode adalah cara menuliskan algoritma menggunakan bahasa yang menyerupai kode pemrograman namun tidak terikat pada sintaks bahasa pemrograman tertentu, sehingga lebih mudah dipahami dan dikomunikasikan kepada siapa saja, termasuk kepada rekan kerja yang mungkin menggunakan bahasa pemrograman berbeda. Pseudocode menjadi jembatan antara algoritma dalam bahasa manusia dan kode program yang sesungguhnya.

```
MULAI
  INPUT nilai
  JIKA nilai  $\geq 75$  MAKA
    TAMPILKAN "Lulus"
  SELAIN ITU
    TAMPILKAN "Tidak Lulus"
  AKHIR JIKA
SELESAI
```

Perhatikan bagaimana pseudocode di atas merepresentasikan flowchart yang sama persis pada Subbab 3.3, namun dalam bentuk teks yang lebih ringkas untuk dituliskan. Kemampuan berpindah dengan lancar antara algoritma verbal, flowchart, dan pseudocode adalah keterampilan penting sebelum akhirnya menerjemahkan logika tersebut ke dalam kode JavaScript sesungguhnya pada Bab 10.

3.5 Latihan Konversi: Dari Algoritma ke Pseudocode

Sebagai latihan tambahan, perhatikan bagaimana algoritma 'menentukan bilangan genap atau ganjil' — sebuah masalah klasik dalam pembelajaran logika pemrograman — dikonversi menjadi pseudocode.

```
MULAI
  INPUT angka
  sisa = angka MOD 2
  JIKA sisa == 0 MAKA
    TAMPILKAN "Bilangan Genap"
  SELAIN ITU
    TAMPILKAN "Bilangan Ganjil"
  AKHIR JIKA
SELESAI
```

Operator MOD (modulus) menghasilkan sisa pembagian — sebuah angka dikatakan genap apabila sisa pembagiannya dengan 2 adalah 0. Konsep operator seperti MOD akan dibahas lebih mendalam pada Bab 6 mengenai variabel dan operator.

Aktivitas Pembelajaran

27. Peserta didik menggambar flowchart lengkap (menggunakan kertas atau aplikasi diagram sederhana) untuk algoritma 'menentukan apakah sebuah tahun adalah tahun kabisat', dengan aturan: tahun kabisat jika habis dibagi 4 dan (tidak habis dibagi 100 atau habis dibagi 400).
28. Peserta didik menerjemahkan flowchart yang telah dibuat pada aktivitas sebelumnya menjadi pseudocode menggunakan format MULAI-JIKA-SELESAI seperti contoh pada bab ini.
29. Peserta didik membuat flowchart dan pseudocode untuk algoritma sehari-hari pilihan sendiri yang memiliki minimal satu struktur keputusan (decision), kemudian mempresentasikannya secara singkat kepada kelas.

Rangkuman

Flowchart menggunakan simbol standar (terminator, proses, decision, input/output) untuk memvisualisasikan alur logika program secara grafis, sedangkan pseudocode menuliskan algoritma dalam bahasa semi-formal yang mudah dibaca tanpa terikat sintaks bahasa pemrograman tertentu. Keduanya berfungsi sebagai alat perencanaan sebelum penulisan kode sesungguhnya, membantu programmer menemukan kesalahan logika sejak dini.

Soal Latihan / Refleksi

30. Sebutkan empat simbol flowchart standar beserta fungsinya masing-masing.
31. Mengapa perencanaan menggunakan flowchart atau pseudocode disarankan sebelum menulis kode program yang sesungguhnya?
32. Buatlah pseudocode untuk menentukan nilai terbesar di antara dua buah bilangan yang diinput pengguna.
33. Jelaskan perbedaan mendasar antara flowchart dan pseudocode dalam merepresentasikan sebuah algoritma.

BAB 4 — Struktur Kontrol: Percabangan (Decision Making)

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan konsep percabangan dalam pemrograman menggunakan analogi kehidupan nyata.
- Peserta didik mampu membedakan struktur percabangan if, if-else, dan if-else if bertingkat.
- Peserta didik mampu menerapkan struktur percabangan pada kasus nyata menggunakan pseudocode dan JavaScript sederhana.

4.1 Percabangan dalam Kehidupan Sehari-hari

Struktur percabangan (branching/decision making) adalah struktur kontrol yang memungkinkan sebuah program mengambil jalur eksekusi berbeda tergantung pada suatu kondisi. Kehidupan sehari-hari dipenuhi dengan keputusan bercabang: 'Jika lampu lalu lintas hijau, jalan terus; jika kuning, bersiap berhenti; jika merah, berhenti.' Keputusan semacam ini memiliki lebih dari dua kemungkinan hasil, mencerminkan struktur percabangan bertingkat yang akan dipelajari pada subbab berikutnya.

4.2 Struktur IF Tunggal

Struktur IF tunggal menjalankan sebuah blok kode hanya apabila kondisi yang diperiksa bernilai benar (true), dan tidak melakukan apa pun apabila kondisi bernilai salah (false). Analoginya seperti aturan 'Jika hujan, bawa payung' — tanpa ada instruksi eksplisit mengenai apa yang harus dilakukan jika tidak hujan.

```
let hujan = true;

if (hujan) {
  console.log("Bawa payung");
}
```

4.3 Struktur IF-ELSE

Struktur IF-ELSE menyediakan dua jalur eksekusi: satu jalur dijalankan apabila kondisi benar, dan jalur lainnya (else) dijalankan apabila kondisi salah. Struktur ini persis seperti algoritma kelulusan pada Bab 3, dan dapat diterapkan pada berbagai skenario kehidupan nyata seperti menentukan tarif tiket berdasarkan kategori usia.

```
let umur = 15;

if (umur >= 17) {
  console.log("Tarif Dewasa: Rp 50.000");
} else {
  console.log("Tarif Anak: Rp 25.000");
}
```

4.4 Struktur IF-ELSE IF Bertingkat

Ketika sebuah keputusan memiliki lebih dari dua kemungkinan hasil — seperti lampu lalu lintas dengan tiga warna, atau sistem penilaian dengan lima kategori huruf — struktur IF-ELSE IF bertingkat digunakan untuk memeriksa beberapa kondisi secara berurutan hingga salah satu kondisi terpenuhi.

```
let nilai = 82;

if (nilai >= 90) {
  console.log("Nilai: A");
} else if (nilai >= 80) {
  console.log("Nilai: B");
} else if (nilai >= 70) {
  console.log("Nilai: C");
} else {
  console.log("Nilai: D");
}
```

Pada contoh di atas, JavaScript memeriksa kondisi secara berurutan dari atas ke bawah dan langsung berhenti begitu menemukan kondisi pertama yang bernilai benar — karena nilai 82 tidak memenuhi kondisi pertama (≥ 90) namun memenuhi kondisi kedua (≥ 80), maka program akan menampilkan 'Nilai: B' tanpa memeriksa kondisi-kondisi berikutnya.

4.5 Operator Logika pada Kondisi Majemuk

Kondisi pada struktur percabangan sering kali perlu menggabungkan lebih dari satu syarat sekaligus, menggunakan operator logika AND (&&), OR (||), dan NOT (!). Sebagai contoh nyata, sebuah sistem pendaftaran SIM mungkin memerlukan pemeriksaan: 'Jika umur ≥ 17 DAN memiliki KTP, maka boleh mendaftar.'

```
let umur = 18;
let punyaKTP = true;

if (umur >= 17 && punyaKTP) {
  console.log("Boleh mendaftar SIM");
} else {
  console.log("Belum memenuhi syarat");
}
```

Operator	Arti	Contoh Kehidupan Nyata
&& (AND)	Benar jika KEDUA kondisi benar	Lulus jika nilai ujian DAN kehadiran memenuhi syarat
(OR)	Benar jika SALAH SATU kondisi benar	Dapat diskon jika member ATAU berbelanja di atas Rp 100.000
! (NOT)	Membalik nilai kondisi	Tombol aktif jika form TIDAK kosong

Aktivitas Pembelajaran

34. Peserta didik menulis kode JavaScript sederhana (dijalankan melalui Console pada Developer Tools browser) untuk menentukan kategori Indeks Massa Tubuh (IMT) berdasarkan rentang nilai menggunakan struktur if-else if bertingkat.
35. Peserta didik membuat program penentu ongkos kirim sederhana berdasarkan berat paket menggunakan kombinasi if-else if dan operator perbandingan.
36. Peserta didik memodifikasi contoh program pendaftaran SIM pada bab ini untuk menambahkan syarat tambahan menggunakan operator logika OR, misalnya 'boleh mendaftar jika memiliki KTP ATAU surat keterangan domisili'.

Rangkuman

Struktur percabangan memungkinkan program mengambil jalur eksekusi berbeda berdasarkan kondisi tertentu, dengan tiga bentuk utama: IF tunggal (satu jalur), IF-ELSE (dua jalur), dan IF-ELSE IF bertingkat (banyak jalur). Operator logika AND, OR, dan NOT memungkinkan penggabungan beberapa syarat dalam satu kondisi. Struktur ini mencerminkan pola pengambilan keputusan yang lazim ditemui dalam kehidupan sehari-hari, dan menjadi salah satu struktur kontrol paling sering digunakan dalam pemrograman.

Soal Latihan / Refleksi

37. Jelaskan perbedaan antara struktur IF tunggal dan IF-ELSE.
38. Kapan sebaiknya digunakan struktur IF-ELSE IF bertingkat dibanding beberapa struktur IF tunggal yang terpisah?
39. Tuliskan kode JavaScript untuk menentukan apakah sebuah tahun adalah tahun kabisat (habis dibagi 4 dan (tidak habis dibagi 100 atau habis dibagi 400)) menggunakan operator logika yang sesuai.
40. Berikan satu contoh kehidupan nyata yang paling tepat direpresentasikan menggunakan operator OR (||), beserta penjelasannya.

BAB 5 — Struktur Kontrol: Perulangan (Looping)

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan konsep perulangan dalam pemrograman menggunakan analogi kehidupan nyata.
- Peserta didik mampu membedakan penggunaan perulangan for dan while.
- Peserta didik mampu menerapkan perulangan untuk menyelesaikan masalah berulang pada kasus nyata.

5.1 Perulangan dalam Kehidupan Sehari-hari

Perulangan (looping) adalah struktur kontrol yang menjalankan satu blok instruksi secara berulang selama kondisi tertentu masih terpenuhi, menghindari kebutuhan menuliskan instruksi yang sama berkali-kali. Aktivitas sehari-hari penuh dengan pola perulangan: 'Ulangi mengaduk adonan sebanyak 20 kali', atau 'Selama air belum mendidih, terus panaskan kompor'. Pola pertama (jumlah pengulangan sudah pasti diketahui di awal) mencerminkan perulangan for, sedangkan pola kedua (pengulangan bergantung pada kondisi yang belum pasti kapan terpenuhi) mencerminkan perulangan while.

5.2 Perulangan FOR: Ketika Jumlah Pengulangan Sudah Diketahui

Perulangan for digunakan ketika jumlah pengulangan sudah diketahui atau dapat dihitung sejak awal, seperti mencetak nomor absen siswa dari 1 hingga 30, atau menghitung total gaji karyawan yang jumlahnya tetap.

```
for (let i = 1; i <= 5; i++) {
  console.log("Absen ke-" + i);
}
// Output: Absen ke-1, Absen ke-2, ... Absen ke-5
```

Struktur for memiliki tiga bagian yang dipisahkan tanda titik koma: inisialisasi (let i = 1, nilai awal), kondisi (i <= 5, syarat perulangan berlanjut), dan increment (i++, perubahan nilai setiap putaran). Ketiga bagian ini bekerja sama menentukan kapan perulangan dimulai, berlanjut, dan berhenti.

5.3 Perulangan WHILE: Ketika Jumlah Pengulangan Belum Pasti

Perulangan while digunakan ketika jumlah pengulangan tidak diketahui sejak awal, melainkan bergantung pada suatu kondisi yang dapat berubah selama program berjalan. Contoh nyata adalah proses transaksi di ATM: 'Selama pengguna belum memasukkan PIN yang benar, minta pengguna memasukkan PIN kembali', di mana jumlah percobaan yang dibutuhkan tidak dapat diketahui sebelumnya.

```
let saldo = 100000;
let penarikan = 30000;

while (saldo >= penarikan) {
```

```

    saldo = saldo - penarikan;
    console.log("Penarikan berhasil, sisa saldo: " + saldo);
}
console.log("Saldo tidak mencukupi untuk penarikan berikutnya");

```

Perulangan di atas akan terus berjalan selama saldo masih mencukupi untuk penarikan, dan berhenti secara otomatis begitu saldo tidak lagi mencukupi — jumlah pengulangan yang terjadi bergantung sepenuhnya pada nilai saldo awal, bukan angka tetap yang ditentukan di awal kode.

5.4 Perulangan Bersarang (Nested Loop)

Perulangan bersarang adalah sebuah perulangan yang berada di dalam perulangan lainnya, umum digunakan untuk memproses data dua dimensi, seperti menampilkan denah kursi bioskop (baris dan kolom) atau membuat pola bintang berbentuk persegi.

```

for (let baris = 1; baris <= 3; baris++) {
    let teksBaris = "";
    for (let kolom = 1; kolom <= 3; kolom++) {
        teksBaris += "* ";
    }
    console.log(teksBaris);
}
// Output:
// * * *
// * * *
// * * *

```

5.5 Menghindari Perulangan Tak Berhingga (Infinite Loop)

Salah satu kesalahan paling umum bagi pemula adalah menciptakan perulangan tak berhingga (infinite loop), yaitu perulangan yang kondisinya tidak pernah menjadi salah sehingga program tidak pernah berhenti dan dapat menyebabkan browser atau aplikasi menjadi tidak responsif (hang). Kesalahan ini sering terjadi karena programmer lupa memperbarui variabel yang menjadi syarat perulangan.

```

// SALAH - infinite loop karena i tidak pernah bertambah
let i = 1;
while (i <= 5) {
    console.log(i);
    // lupa menuliskan i++, sehingga i selamanya bernilai 1
}

// BENAR
let j = 1;
while (j <= 5) {
    console.log(j);
    j++; // memperbarui nilai j agar perulangan akhirnya berhenti
}

```

Analoginya seperti instruksi 'terus panaskan kompor selama air belum mendidih', namun lupa menyalakan kompor sama sekali — air tidak akan pernah mendidih, dan instruksi tersebut akan terus berulang selamanya tanpa ada kemajuan menuju kondisi berhenti.

Aktivitas Pembelajaran

41. Peserta didik menulis kode JavaScript menggunakan perulangan for untuk menampilkan tabel perkalian 1 sampai 10 dari sebuah angka yang ditentukan.
42. Peserta didik menulis kode JavaScript menggunakan perulangan while untuk mensimulasikan proses menabung hingga mencapai target tertentu, menampilkan jumlah bulan yang dibutuhkan.
43. Peserta didik membuat pola bintang berbentuk segitiga menggunakan perulangan bersarang, kemudian memodifikasinya menjadi pola berbentuk terbalik.
44. Peserta didik dengan sengaja membuat sebuah infinite loop pada Developer Tools browser (dengan pengawasan guru), mengamati dampaknya, kemudian memperbaikinya.

Rangkuman

Perulangan memungkinkan sebuah blok instruksi dijalankan berulang kali tanpa perlu menuliskannya berkali-kali, dengan for digunakan ketika jumlah pengulangan sudah diketahui dan while ketika bergantung pada kondisi yang belum pasti. Perulangan bersarang memungkinkan pemrosesan data dua dimensi, sementara kesalahan lupa memperbarui variabel kondisi dapat menyebabkan infinite loop yang harus dihindari. Struktur kontrol sekuensial, kondisional (Bab 4), dan perulangan yang telah dipelajari menjadi fondasi lengkap sebelum mempelajari variabel dan operator secara lebih formal pada bab berikutnya.

Soal Latihan / Refleksi

45. Jelaskan kapan sebaiknya menggunakan perulangan for dibanding while, beserta contoh kasus masing-masing.
46. Apa yang menyebabkan terjadinya infinite loop, dan bagaimana cara mencegahnya?
47. Tuliskan kode JavaScript menggunakan perulangan untuk menjumlahkan seluruh bilangan genap dari 1 hingga 50.
48. Jelaskan kegunaan perulangan bersarang beserta satu contoh kasus nyata yang tepat menggunakannya.

BAB 6 — Variabel, Tipe Data, dan Operator

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan konsep variabel sebagai wadah penyimpanan data menggunakan analogi kehidupan nyata.
- Peserta didik mampu mengidentifikasi tipe data dasar dalam pemrograman.
- Peserta didik mampu menggunakan operator aritmatika, perbandingan, dan penugasan dengan tepat.

6.1 Variabel: Kotak Penyimpan Data Bernama

Variabel adalah tempat penyimpanan data dalam program yang diberi nama, memungkinkan data tersebut digunakan dan diubah kapan pun diperlukan sepanjang program berjalan. Analogi paling sederhana adalah kotak penyimpanan berlabel — misalnya kotak berlabel 'Uang Jajan' dapat diisi dengan nilai tertentu, digunakan (dikurangi saat berbelanja), dan diisi ulang (ditambah saat menerima uang jajan baru), namun label 'Uang Jajan' pada kotak tersebut tetap sama meskipun isinya berubah-ubah.

```
let uangJajan = 20000;
console.log(uangJajan); // 20000

uangJajan = uangJajan - 5000; // dipakai jajan
console.log(uangJajan); // 15000
```

Dalam JavaScript, variabel dideklarasikan menggunakan kata kunci `let` (nilai dapat berubah), `const` (nilai tetap/konstan setelah didefinisikan), atau `var` (cara lama, kini jarang digunakan karena memiliki beberapa kelemahan dibanding let/const).

6.2 Aturan Penamaan Variabel

Aturan	Contoh Benar	Contoh Salah
Dimulai huruf atau underscore, bukan angka	namaSiswa, _total	1nama (dimulai angka)
Tidak boleh mengandung spasi	hargaBarang	harga barang
Case-sensitive (huruf besar/kecil dianggap berbeda)	umur ≠ Umur	-
Sebaiknya deskriptif dan bermakna	totalHarga, jumlahSiswa	x, data1 (kurang jelas maknanya)
Konvensi umum: camelCase	hargaBarang, namaLengkap	harga_barang (snake_case, kurang lazim di JS)

6.3 Tipe Data Dasar

Tipe Data	Contoh Nilai	Kegunaan dalam Kehidupan Nyata
Number	17, 3.14, -5	Menyimpan umur, harga, nilai ujian, suhu
String	"Budi", "Jl. Merdeka No. 5"	Menyimpan nama, alamat, teks apa pun
Boolean	true, false	Menyimpan status seperti 'sudah bayar' (true) atau 'belum bayar' (false)
Array	["Senin", "Selasa", "Rabu"]	Menyimpan daftar/kumpulan data, seperti daftar nama siswa sekelas
Object	{nama: "Budi", umur: 17}	Menyimpan data terstruktur dengan banyak atribut, seperti data biodata siswa

```

let nama = "Budi Santoso";    // String
let umur = 17;                // Number
let sudahLulus = true;        // Boolean
let hobi = ["Membaca", "Coding", "Futsal"]; // Array
let siswa = {                 // Object
  nama: "Budi Santoso",
  umur: 17,
  kelas: "X TJKT 1"
};

```

6.4 Operator Aritmatika

Operator	Fungsi	Contoh
+	Penjumlahan	5 + 3 menghasilkan 8
-	Pengurangan	5 - 3 menghasilkan 2
*	Perkalian	5 * 3 menghasilkan 15
/	Pembagian	6 / 3 menghasilkan 2
%	Modulus (sisa bagi)	7 % 2 menghasilkan 1
**	Pemangkatan	2 ** 3 menghasilkan 8

Operator modulus (%) sering digunakan dalam skenario kehidupan nyata seperti menentukan genap-ganjil (sebagaimana dipelajari pada Bab 3), mengelompokkan data ke dalam beberapa kategori secara bergiliran, atau menghitung sisa uang kembalian dalam pecahan tertentu.

6.5 Operator Perbandingan dan Penugasan

Kategori	Operator	Contoh dan Hasil
Perbandingan	<code>== / ===</code>	5 === 5 menghasilkan true (=== memeriksa tipe data sekaligus nilai, lebih disarankan)
Perbandingan	<code>> < >= <=</code>	17 >= 17 menghasilkan true
Perbandingan	<code>!==</code>	5 !== 3 menghasilkan true (tidak sama dengan)
Penugasan	<code>=</code>	let umur = 17 (memberi nilai 17 ke variabel umur)
Penugasan	<code>+= -= *= /=</code>	umur += 1 setara dengan umur = umur + 1

Perbedaan antara operator `==` dan `===` penting dipahami: `==` hanya membandingkan nilai (dan dapat melakukan konversi tipe data secara otomatis, kadang menimbulkan hasil tak terduga), sedangkan `===` membandingkan nilai sekaligus tipe data secara ketat. Sebagai praktik terbaik, penggunaan `===` lebih disarankan untuk menghindari kesalahan logika yang sulit dilacak.

Aktivitas Pembelajaran

49. Peserta didik membuat variabel untuk menyimpan data biodata diri sendiri (nama, umur, kelas, hobi sebagai array) menggunakan tipe data yang sesuai, kemudian menampilkannya menggunakan `console.log`.
50. Peserta didik menulis program kalkulator sederhana menggunakan operator aritmatika yang menerima dua angka dan menampilkan hasil penjumlahan, pengurangan, perkalian, dan pembagian.
51. Peserta didik membuat program yang membandingkan dua nilai ujian menggunakan operator perbandingan, menampilkan nilai mana yang lebih besar atau apakah keduanya sama.

Rangkuman

Variabel berfungsi sebagai wadah penyimpanan data bernama yang nilainya dapat berubah selama program berjalan, dengan tipe data dasar meliputi Number, String, Boolean, Array, dan Object. Operator aritmatika (+, -, , /, %, *), perbandingan (===, >, <, !==), dan penugasan (=, +=, -=) menjadi alat dasar untuk memanipulasi dan membandingkan data dalam program. Ketiga bab sebelumnya (percabangan, perulangan, dan variabel/operator) bersama-sama membentuk fondasi logika pemrograman yang lengkap sebelum memasuki Bagian II buku ini mengenai dasar-dasar pengembangan web.

Soal Latihan / Refleksi

52. Jelaskan perbedaan antara variabel yang dideklarasikan dengan `let` dan `const`.
53. Sebutkan lima tipe data dasar dalam JavaScript beserta contoh penggunaannya dalam kehidupan nyata.
54. Apa perbedaan antara operator `==` dan `===`? Mengapa `===` lebih disarankan digunakan?
55. Tuliskan kode JavaScript yang menggunakan operator modulus untuk menentukan apakah sebuah tahun habis dibagi 4.

BAB 7 — Pengenalan HTML: Struktur Dasar Dokumen Web

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan peran HTML dalam pembangunan sebuah halaman web.
- Peserta didik mampu menyusun struktur dasar dokumen HTML yang valid.
- Peserta didik mampu menggunakan tag-tag dasar HTML untuk menyusun konten sederhana.

7.1 Apa Itu HTML dan Mengapa Penting?

HyperText Markup Language (HTML) adalah bahasa markup standar yang digunakan untuk menyusun struktur dan konten sebuah halaman web. Jika sebuah website diibaratkan sebagai rumah, HTML berperan sebagai kerangka bangunan (dinding, atap, ruangan) yang menentukan struktur dasar, sementara CSS (Bab 9) berperan seperti cat dan dekorasi yang mempercantik tampilan, dan JavaScript (Bab 10) berperan seperti sistem kelistrikan yang membuat rumah menjadi 'hidup' dan interaktif (lampu dapat dinyalakan, pintu otomatis dapat dibuka).

Hampir seluruh halaman web yang diakses sehari-hari — mulai dari media sosial, toko daring, hingga portal berita — dibangun di atas fondasi HTML. Memahami HTML adalah langkah pertama mutlak yang harus dikuasai sebelum mempelajari CSS maupun JavaScript.

7.2 Struktur Dasar Dokumen HTML

Setiap dokumen HTML memiliki struktur dasar yang wajib diikuti agar dapat ditampilkan dengan benar oleh browser.

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Halaman Pertamaku</title>
</head>
<body>
  <h1>Selamat Datang!</h1>
  <p>Ini adalah halaman web pertama saya.</p>
</body>
</html>
```

Elemen	Fungsi
<!DOCTYPE html>	Memberitahu browser bahwa dokumen ini menggunakan standar HTML5
<html>	Elemen pembungkus (root) seluruh isi dokumen HTML
<head>	Berisi informasi metadata halaman yang tidak tampil langsung di layar (judul tab, encoding, tautan CSS)

Elemen	Fungsi
<code><title></code>	Menentukan judul yang tampil pada tab browser
<code><body></code>	Berisi seluruh konten yang tampil dan terlihat oleh pengunjung website

7.3 Tag Heading dan Paragraf

HTML menyediakan enam tingkat heading (`<h1>` hingga `<h6>`) untuk menyusun hierarki judul dan subjudul konten, mirip dengan struktur bab dan subbab pada sebuah buku, di mana `<h1>` digunakan untuk judul utama halaman (idealnya hanya satu `<h1>` per halaman) dan `<h6>` untuk sub-subjudul paling detail.

```
<h1>Profil Sekolah</h1>
<h2>Visi dan Misi</h2>
<p>Sekolah kami berkomitmen mencetak lulusan yang kompeten dan berkarakter.</p>
<h2>Program Keahlian</h2>
<h3>Teknik Jaringan Komputer dan Telekomunikasi</h3>
<p>Program keahlian yang mempelajari jaringan komputer, keamanan siber, dan pemrograman web.</p>
```

7.4 Tag Dasar Lainnya: Link, Gambar, dan List

Tag	Fungsi	Contoh Penggunaan
<code><a></code>	Membuat tautan (hyperlink) ke halaman lain	<code>Kunjungi Website Sekolah</code>
<code></code>	Menampilkan gambar pada halaman	<code></code>
<code> / </code>	Membuat daftar tidak berurutan (bullet)	<code>HTMLCSS</code>
<code> / </code>	Membuat daftar berurutan (bernomor)	<code>Buka editorTulis kode</code>
<code><div></code>	Elemen pembungkus generik untuk mengelompokkan konten	<code><div class="header">...</div></code>
<code></code>	Elemen pembungkus generik untuk teks dalam satu baris	<code>teks penting</code>

Atribut `alt` pada tag `<img>` sangat penting untuk aksesibilitas — teks tersebut akan dibacakan oleh pembaca layar (screen reader) bagi pengguna dengan gangguan penglihatan, dan juga ditampilkan sebagai teks pengganti apabila gambar gagal dimuat.

7.5 Relevansi HTML dalam Kehidupan Nyata

Untuk memahami relevansi HTML secara nyata, coba klik kanan pada halaman web mana pun yang sering dikunjungi (misalnya halaman berita atau media sosial) dan pilih 'Lihat Kode Sumber Halaman' (View Page Source) atau tekan Ctrl+U pada browser. Peserta didik akan melihat bahwa di balik tampilan yang menarik dan kompleks, setiap halaman web pada dasarnya tersusun dari tag-tag HTML yang sama seperti yang baru saja dipelajari, hanya dengan skala dan kompleksitas yang jauh lebih besar.

Aktivitas Pembelajaran

56. Peserta didik membuat dokumen HTML pertama menggunakan struktur dasar yang telah dipelajari, berisi judul halaman, heading, dan minimal dua paragraf tentang diri mereka sendiri, kemudian membukanya di browser.
57. Peserta didik menambahkan sebuah daftar (list) hobi menggunakan dan sebuah gambar profil menggunakan pada dokumen HTML yang telah dibuat.
58. Peserta didik membuka halaman web favorit mereka, menggunakan fitur 'Lihat Kode Sumber Halaman', dan mengidentifikasi minimal lima tag HTML yang telah dipelajari pada bab ini di dalam kode sumber tersebut.

Rangkuman

HTML adalah bahasa markup yang membentuk struktur dan konten dasar sebuah halaman web, tersusun dari dokumen dengan struktur baku (DOCTYPE, html, head, body) dan berbagai tag seperti heading, paragraf, tautan, gambar, dan list. Pemahaman HTML menjadi fondasi mutlak sebelum mempelajari HTML lanjutan pada bab berikutnya, yang akan membahas formatting teks, tabel, dan form secara lebih mendalam.

Soal Latihan / Refleksi

59. Jelaskan fungsi elemen <head> dan <body> pada sebuah dokumen HTML.
60. Mengapa idealnya hanya terdapat satu tag <h1> dalam satu halaman HTML?
61. Tuliskan kode HTML untuk membuat daftar berurutan berisi tiga langkah membuat mi instan.
62. Jelaskan fungsi atribut alt pada tag dan mengapa atribut ini penting.

BAB 8 — HTML Lanjutan: Tabel dan Form

Tujuan Pembelajaran

- Peserta didik mampu menyusun tabel HTML untuk menampilkan data terstruktur.
- Peserta didik mampu membangun form HTML dengan berbagai jenis input.
- Peserta didik mampu menerapkan atribut form yang relevan untuk validasi dasar.

8.1 Menyusun Tabel dengan HTML

Tabel HTML digunakan untuk menampilkan data yang secara alami berbentuk baris dan kolom, seperti jadwal pelajaran, daftar nilai, atau tabel harga produk — situasi yang sering ditemui dalam konteks bisnis maupun akademik sehari-hari.

```
<table border="1">
  <thead>
    <tr>
      <th>Nama</th>
      <th>Kelas</th>
      <th>Nilai</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Budi Santoso</td>
      <td>X TJKT 1</td>
      <td>88</td>
    </tr>
    <tr>
      <td>Siti Aminah</td>
      <td>X TJKT 1</td>
      <td>92</td>
    </tr>
  </tbody>
</table>
```

Tag	Fungsi
<table>	Elemen pembungkus keseluruhan tabel
<thead> / <tbody>	Mengelompokkan bagian judul (header) dan isi (body) tabel
<tr>	Membuat satu baris (table row) baru
<th>	Sel judul kolom (table header), teksnya otomatis tebal dan rata tengah
<td>	Sel data biasa (table data)

8.2 Form HTML: Gerbang Interaksi Pengguna

Form adalah elemen HTML yang memungkinkan pengguna memasukkan data dan mengirimkannya ke server, menjadi fondasi hampir seluruh interaksi bermakna pada sebuah website — mulai dari kolom pencarian, formulir pendaftaran, kotak komentar, hingga form login yang akan dipraktikkan pada Bab 12. Tanpa form, sebuah website hanya akan bersifat satu arah (pengunjung hanya dapat membaca), sedangkan form memungkinkan komunikasi dua arah antara pengguna dan sistem.

```
<form action="proses.php" method="POST">
  <label for="nama">Nama Lengkap:</label>
  <input type="text" id="nama" name="nama" placeholder="Masukkan nama Anda"
  required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <button type="submit">Kirim</button>
</form>
```

8.3 Jenis-Jenis Input Form

Jenis Input	Kegunaan Nyata
<code><input type="text"></code>	Kotak isian teks bebas, seperti nama atau alamat
<code><input type="email"></code>	Kotak isian email dengan validasi format otomatis oleh browser
<code><input type="password"></code>	Kotak isian kata sandi, karakter yang diketik disamarkan (●●●●)
<code><input type="number"></code>	Kotak isian angka, seperti umur atau jumlah barang
<code><input type="radio"></code>	Pilihan tunggal dari beberapa opsi, seperti jenis kelamin
<code><input type="checkbox"></code>	Pilihan ganda (boleh lebih dari satu), seperti hobi yang diminati
<code><select> / <option></code>	Menu dropdown untuk memilih satu opsi dari daftar, seperti memilih kota domisili
<code><textarea></code>	Kotak isian teks panjang multi-baris, seperti kolom pesan atau komentar

```
<label>Jenis Kelamin:</label>
<input type="radio" id="lk" name="gender" value="L"> <label for="lk">Laki-
laki</label>
<input type="radio" id="pr" name="gender" value="P"> <label
for="pr">Perempuan</label>

<label for="kota">Kota Domisili:</label>
<select id="kota" name="kota">
  <option value="jakarta">Jakarta</option>
  <option value="bandung">Bandung</option>
  <option value="surabaya">Surabaya</option>
</select>
```

```
<label for="pesan">Pesan:</label>
<textarea id="pesan" name="pesan" rows="4" placeholder="Tulis pesan Anda di sini"></textarea>
```

8.4 Atribut Validasi Dasar pada Form

HTML5 menyediakan beberapa atribut validasi bawaan yang membantu memastikan data yang dimasukkan pengguna sesuai format yang diharapkan, tanpa perlu menulis kode JavaScript tambahan untuk validasi sederhana.

Atribut	Fungsi
required	Mewajibkan sebuah kolom diisi sebelum form dapat dikirim
placeholder	Menampilkan teks petunjuk yang hilang otomatis saat pengguna mulai mengetik
minlength / maxlength	Membatasi panjang minimal/maksimal teks yang dapat dimasukkan
pattern	Menentukan pola (regular expression) yang harus dipenuhi nilai input
min / max	Membatasi nilai minimal/maksimal pada input bertipe number

Atribut-atribut ini menjadi lapisan validasi pertama yang berjalan langsung di sisi browser (client-side), memberikan umpan balik instan kepada pengguna sebelum data benar-benar dikirim ke server — sebuah praktik penting yang akan relevan kembali saat mempelajari koneksi form login ke database pada Bab 12.

Aktivitas Pembelajaran

63. Peserta didik membuat tabel HTML berisi jadwal pelajaran pribadi selama satu minggu, lengkap dengan header hari dan mata pelajaran.
64. Peserta didik membuat form pendaftaran sederhana (nama, email, jenis kelamin, kota domisili, dan pesan) menggunakan berbagai jenis input yang telah dipelajari, lengkap dengan atribut required pada kolom wajib.
65. Peserta didik menguji form yang telah dibuat dengan sengaja mengosongkan kolom wajib dan mengamati pesan validasi otomatis yang ditampilkan browser.

Rangkuman

Tabel HTML (table, tr, th, td) digunakan untuk menampilkan data terstruktur berbentuk baris dan kolom, sementara form HTML dengan berbagai jenis input (text, email, password, radio, checkbox, select, textarea) menjadi gerbang interaksi dua arah antara pengguna dan website. Atribut validasi seperti

required dan pattern memberikan lapisan pemeriksaan dasar di sisi browser. Kompetensi form ini akan menjadi fondasi penting saat praktik membuat form login yang tersambung ke database pada Bab 12.

Soal Latihan / Refleksi

66. Jelaskan perbedaan fungsi tag <th> dan <td> dalam sebuah tabel HTML.
67. Sebutkan tiga jenis input form beserta kegunaannya masing-masing dalam konteks form pendaftaran online.
68. Apa fungsi atribut required pada sebuah elemen input, dan apa yang terjadi jika form dikirim tanpa mengisi kolom tersebut?
69. Tuliskan kode HTML untuk membuat sebuah dropdown pilihan ukuran baju (S, M, L, XL).

BAB 9 — Pengenalan CSS: Mempercantik Tampilan Web

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan peran CSS dalam pengembangan web dan cara menerapkannya pada HTML.
- Peserta didik mampu menggunakan selector, properti warna, tipografi, dan spacing pada CSS.
- Peserta didik mampu menerapkan konsep dasar layout menggunakan Flexbox.

9.1 Apa Itu CSS dan Tiga Cara Menerapkannya

Cascading Style Sheets (CSS) adalah bahasa yang mengatur tampilan visual sebuah halaman HTML — warna, ukuran font, jarak antar elemen, tata letak, hingga animasi sederhana. Jika HTML adalah kerangka bangunan sebagaimana dianalogikan pada Bab 7, CSS adalah cat, dekorasi interior, dan tata ruang yang membuat bangunan tersebut terlihat menarik dan nyaman dipandang.

Cara Penerapan	Contoh	Kapan Digunakan
Inline	<code><p style="color: red;">Teks</p></code>	Untuk perubahan kecil dan cepat pada satu elemen saja (jarang direkomendasikan)
Internal	<code><style></code> di dalam <code><head></code>	Untuk halaman tunggal dengan styling yang tidak terlalu banyak
External	<code><link rel="stylesheet" href="style.css"></code>	Praktik terbaik — memisahkan file CSS agar dapat digunakan di banyak halaman sekaligus

Metode external (file CSS terpisah) adalah praktik terbaik yang akan digunakan pada proyek landing page di Bab 11, karena memisahkan struktur (HTML) dari tampilan (CSS) membuat kode lebih rapi, mudah dipelihara, dan dapat digunakan kembali di banyak halaman.

9.2 Selector CSS: Menargetkan Elemen yang Ingin Distyling

```
/* Selector Tag - menargetkan semua elemen <p> */
p {
  color: #333333;
}

/* Selector Class - menargetkan elemen dengan class="highlight" */
.highlight {
  background-color: yellow;
}

/* Selector ID - menargetkan elemen dengan id="judul-utama" */
#judul-utama {
  font-size: 32px;
}
```

Selector	Sintaks	Karakteristik
Tag	p, h1, div	Menargetkan seluruh elemen dengan tag tersebut di halaman
Class	.nama-class	Menargetkan elemen ber-atribut class tertentu, dapat digunakan berulang kali
ID	#nama-id	Menargetkan satu elemen unik ber-atribut id tertentu, hanya boleh digunakan sekali per halaman

9.3 Warna, Tipografi, dan Spacing

```
.kartu-profil {
  color: #2E2E2E;           /* warna teks */
  background-color: #F5F5F5; /* warna latar belakang */
  font-family: 'Segoe UI', sans-serif;
  font-size: 16px;
  font-weight: bold;
  text-align: center;
  padding: 20px;             /* jarak antara konten dan tepi elemen */
  margin: 10px;              /* jarak antara elemen ini dan elemen lain di sekitarnya */
  border-radius: 8px;        /* membuat sudut elemen menjadi melengkung */
}
```

Perbedaan antara `padding` dan `margin` sering membingungkan pemula: padding adalah jarak di **dalam** elemen (antara konten dan tepi/border elemen tersebut), sedangkan margin adalah jarak di **luar** elemen (antara elemen tersebut dengan elemen lain di sekitarnya). Analoginya seperti bingkai foto — padding adalah jarak antara foto dan bingkainya, sedangkan margin adalah jarak antara bingkai tersebut dengan bingkai foto lain di dinding.

9.4 Dasar Layout dengan Flexbox

Flexbox adalah salah satu sistem layout CSS modern yang memudahkan penyusunan elemen secara horizontal maupun vertikal, sejajar maupun tersebar merata — kebutuhan tata letak yang sangat umum ditemui, misalnya menyusun menu navigasi secara horizontal atau menampilkan beberapa kartu produk sejajar dalam satu baris.

```
.navigasi {
  display: flex;             /* mengaktifkan flexbox pada elemen pembungkus */
  justify-content: space-between; /* menyebar elemen anak secara merata horizontal */
  /*
  align-items: center;       /* menyejajarkan elemen anak secara vertikal di tengah */
  */
}
```

Properti	Fungsi
display: flex;	Mengaktifkan mode flexbox pada sebuah elemen pembungkus (container)
justify-content	Mengatur perataan elemen anak pada sumbu horizontal (mis. center, space-between)
align-items	Mengatur perataan elemen anak pada sumbu vertikal (mis. center, flex-start)
flex-direction	Menentukan arah susunan elemen anak (row = horizontal, column = vertikal)

9.5 Relevansi CSS dalam Kehidupan Nyata

Perbedaan antara website yang terlihat profesional dan meyakinkan dengan website yang terlihat berantakan dan tidak dipercaya seringkali murni terletak pada penerapan CSS yang baik, bukan pada konten HTML-nya. Studi tentang pengalaman pengguna (user experience) secara konsisten menunjukkan bahwa pengunjung website membentuk kesan pertama terhadap kredibilitas sebuah bisnis hanya dalam hitungan detik berdasarkan tampilan visual — inilah sebabnya perusahaan rela menginvestasikan banyak sumber daya untuk memastikan CSS pada website mereka rapi, konsisten, dan mencerminkan identitas merek (branding) yang tepat.

Aktivitas Pembelajaran

70. Peserta didik membuat file CSS eksternal dan menghubungkannya ke dokumen HTML profil diri yang dibuat pada Bab 7, menerapkan warna, font, dan spacing pada heading dan paragraf.
71. Peserta didik menggunakan selector class untuk membuat sebuah 'kartu' (card) berisi foto dan biodata singkat, lengkap dengan padding, margin, dan border-radius.
72. Peserta didik menyusun sebuah menu navigasi horizontal sederhana (Beranda, Tentang, Kontak) menggunakan Flexbox dengan justify-content: space-between.

Rangkuman

CSS mengatur tampilan visual halaman HTML melalui tiga metode penerapan (inline, internal, external), dengan external menjadi praktik terbaik. Selector (tag, class, ID) digunakan untuk menargetkan elemen yang akan distyling, dilengkapi properti warna, tipografi, padding, dan margin. Flexbox menyediakan sistem layout modern untuk menyusun elemen secara fleksibel. Penguasaan CSS menjadi pembeda utama antara website yang terlihat profesional dengan yang tidak, dan akan diterapkan secara penuh pada proyek landing page di Bab 11.

Soal Latihan / Refleksi

73. Jelaskan perbedaan antara selector class dan selector ID pada CSS, beserta kapan masing-masing sebaiknya digunakan.

74. Apa perbedaan antara properti padding dan margin? Berikan analoginya.
75. Jelaskan fungsi properti display: flex; beserta dua properti flexbox lainnya yang sering digunakan bersamaan.
76. Mengapa metode external CSS (file terpisah) dianggap sebagai praktik terbaik dibanding inline atau internal CSS?

BAB 10 — Pengenalan JavaScript: Membuat Web Menjadi Interaktif

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan peran JavaScript dalam membuat halaman web menjadi interaktif.
- Peserta didik mampu memanipulasi elemen HTML menggunakan DOM (Document Object Model).
- Peserta didik mampu menangani event pengguna seperti klik dan input menggunakan JavaScript.

10.1 Dari Logika ke Web: Menyatukan Bab 2–6 dengan HTML/CSS

Bab ini menyatukan dua kompetensi besar yang telah dipelajari secara terpisah: logika pemrograman (Bab 2–6) dan struktur/tampilan web (Bab 7–9). JavaScript adalah bahasa pemrograman yang berjalan langsung di dalam browser, memungkinkan seluruh konsep percabangan, perulangan, variabel, dan operator yang telah dipelajari sebelumnya diterapkan langsung untuk mengubah, menganalisis, dan merespons interaksi pengguna pada halaman HTML yang telah disusun.

Tanpa JavaScript, sebuah halaman web bersifat statis — tampilannya sama persis setiap kali dimuat, tidak peduli apa yang dilakukan pengunjung. Dengan JavaScript, halaman web dapat merespons klik tombol, memvalidasi input form secara real-time, menampilkan/menyembunyikan konten, dan mengubah tampilan tanpa perlu memuat ulang seluruh halaman.

10.2 Menyisipkan JavaScript ke dalam HTML

```
<!DOCTYPE html>
<html lang="id">
<head>
  <title>Belajar JavaScript</title>
</head>
<body>
  <h1 id="judul">Halo!</h1>
  <button onclick="ubahTeks()">Klik Saya</button>

  <script>
    function ubahTeks() {
      document.getElementById("judul").innerText = "Anda telah mengklik tombol!";
    }
  </script>
</body>
</html>
```

Sebagaimana CSS, JavaScript juga dapat ditulis secara internal (di dalam tag `<script>`) maupun eksternal (file .js terpisah yang dihubungkan menggunakan `<script src="nama-file.js"></script>`), dengan pendekatan eksternal menjadi praktik terbaik untuk proyek yang lebih besar seperti landing page pada Bab 11.

10.3 DOM: Jembatan antara JavaScript dan HTML

Document Object Model (DOM) adalah representasi struktur HTML sebuah halaman dalam bentuk yang dapat diakses dan dimanipulasi oleh JavaScript. Melalui DOM, JavaScript dapat 'menjangkau' elemen HTML tertentu — misalnya sebuah heading, paragraf, atau tombol — untuk membaca isinya, mengubah teksnya, mengubah gaya CSS-nya, atau bahkan menghapusnya dari halaman.

```
// Mengambil elemen berdasarkan id
let judul = document.getElementById("judul");

// Mengubah isi teks elemen
judul.innerText = "Teks Baru";

// Mengubah gaya CSS elemen secara langsung
judul.style.color = "blue";
judul.style.fontSize = "40px";

// Mengambil elemen berdasarkan selector CSS (lebih fleksibel)
let tombol = document.querySelector(".btn-utama");
```

Method DOM	Fungsi
<code>document.getElementById("id")</code>	Mengambil satu elemen berdasarkan atribut id
<code>document.querySelector("selector")</code>	Mengambil elemen pertama yang cocok dengan selector CSS apa pun
<code>document.querySelectorAll("selector")</code>	Mengambil seluruh elemen yang cocok dengan selector, menghasilkan daftar (list)
<code>.innerText / .innerHTML</code>	Membaca atau mengubah isi teks/HTML sebuah elemen
<code>.style.namaProperti</code>	Membaca atau mengubah gaya CSS sebuah elemen secara langsung

10.4 Menangani Event Pengguna

Event adalah aksi yang dilakukan pengguna terhadap halaman web — mengklik tombol, mengetik pada kolom input, menggerakkan mouse, atau mengirim form. JavaScript memungkinkan program 'mendengarkan' event tersebut dan menjalankan fungsi tertentu sebagai responsnya, sebuah pola yang dikenal sebagai event handling.

```
let tombolHitung = document.getElementById("hitung");

tombolHitung.addEventListener("click", function() {
  let panjang = document.getElementById("panjang").value;
  let lebar = document.getElementById("lebar").value;
  let luas = panjang * lebar;
  document.getElementById("hasil").innerText = "Luas: " + luas + " m²";
});
```

Event	Terjadi Ketika
click	Pengguna mengklik sebuah elemen (tombol, tautan, dll.)
input / change	Pengguna mengetik atau mengubah nilai pada sebuah kolom form
submit	Pengguna mengirimkan sebuah form
mouseover / mouseout	Kursor mouse masuk atau keluar dari area sebuah elemen

10.5 Studi Kasus: Kalkulator Luas Sederhana

Sebagai penerapan menyeluruh dari struktur kontrol (Bab 4–5), variabel dan operator (Bab 6), HTML form (Bab 8), dan DOM/event (bab ini), perhatikan contoh kalkulator luas persegi panjang yang lengkap berikut, sebuah pola yang serupa dengan berbagai kalkulator sederhana yang umum ditemui di internet (kalkulator BMI, kalkulator zakat, kalkulator cicilan).

```
<input type="number" id="panjang" placeholder="Panjang (m)">
<input type="number" id="lebar" placeholder="Lebar (m)">
<button id="hitung">Hitung Luas</button>
<p id="hasil"></p>

<script>
document.getElementById("hitung").addEventListener("click", function() {
  let p = Number(document.getElementById("panjang").value);
  let l = Number(document.getElementById("lebar").value);

  if (p <= 0 || l <= 0) {
    document.getElementById("hasil").innerText = "Masukkan angka yang valid!";
  } else {
    let luas = p * l;
    document.getElementById("hasil").innerText = "Luas: " + luas + " m²";
  }
});
</script>
```

Perhatikan bagaimana fungsi `Number()` digunakan untuk mengonversi nilai input (yang secara default berupa `String`) menjadi tipe `Number` agar dapat dikalikan secara matematis, dan bagaimana struktur `if-else` dari Bab 4 digunakan untuk memvalidasi input sebelum melakukan perhitungan — sebuah pola integrasi lengkap yang akan menjadi dasar proyek landing page pada bab berikutnya.

Aktivitas Pembelajaran

77. Peserta didik membuat halaman HTML dengan sebuah tombol yang, ketika diklik, mengubah warna latar belakang halaman menggunakan JavaScript dan properti `.style.backgroundColor`.

78. Peserta didik membangun kalkulator sederhana serupa contoh pada bab ini, namun untuk menghitung Indeks Massa Tubuh (IMT) berdasarkan input berat dan tinggi badan, lengkap dengan kategori hasil menggunakan if-else if.
79. Peserta didik membuat form dengan satu kolom input nama dan tombol submit yang, ketika diklik, menampilkan pesan sapaan personal (misalnya 'Halo, Budi!') menggunakan event listener dan manipulasi DOM.

Rangkuman

JavaScript menghidupkan halaman HTML yang statis menjadi interaktif melalui manipulasi DOM (mengubah teks, gaya, dan struktur elemen) dan penanganan event pengguna (click, input, submit). Bab ini menyatukan seluruh kompetensi logika pemrograman dari Bab 2–6 dengan struktur HTML dari Bab 7–8, terbukti melalui studi kasus kalkulator luas sederhana yang menggabungkan variabel, percabangan, DOM, dan event handling sekaligus. Kompetensi ini menjadi bekal utama sebelum memasuki proyek landing page penuh pada Bab 11.

Soal Latihan / Refleksi

80. Jelaskan apa yang dimaksud dengan DOM dan mengapa JavaScript memerlukannya untuk berinteraksi dengan HTML.
81. Sebutkan perbedaan antara method `getElementById` dan `querySelector`.
82. Jelaskan pola umum event handling menggunakan `addEventListener`, beserta contoh sederhana.
83. Mengapa nilai dari sebuah elemen input perlu dikonversi menggunakan `Number()` sebelum digunakan dalam operasi perhitungan matematis?

BAB 11 — Praktik: Membangun Landing Page Profil Sederhana

Tujuan Pembelajaran

- Peserta didik mampu merancang struktur informasi sebuah landing page profil.
- Peserta didik mampu mengimplementasikan landing page menggunakan kombinasi HTML dan CSS yang telah dipelajari.
- Peserta didik mampu menambahkan interaktivitas dasar menggunakan JavaScript pada landing page yang dibangun.

11.1 Apa Itu Landing Page dan Mengapa Penting?

Landing page adalah sebuah halaman web tunggal yang dirancang khusus untuk memperkenalkan seseorang, produk, atau organisasi secara ringkas namun meyakinkan kepada pengunjung. Dalam dunia kerja nyata, landing page profil digunakan secara luas: seorang freelancer menggunakannya sebagai portofolio digital, sebuah startup menggunakannya untuk memperkenalkan produk sebelum peluncuran resmi, dan pencari kerja di bidang teknologi semakin sering diminta menyertakan landing page profil pribadi sebagai pelengkap CV konvensional.

Bab ini akan memandu peserta didik membangun landing page profil pribadi secara bertahap, mengintegrasikan seluruh kompetensi HTML (Bab 7–8), CSS (Bab 9), dan JavaScript (Bab 10) yang telah dipelajari menjadi satu proyek nyata dan utuh.

11.2 Merancang Struktur Informasi (Wireframe Sederhana)

Sebagaimana perencanaan flowchart mendahului penulisan kode program (Bab 3), perancangan struktur informasi sebaiknya mendahului penulisan kode landing page. Sebuah landing page profil yang efektif umumnya tersusun dari beberapa bagian (section) berikut, disusun secara berurutan dari atas ke bawah halaman.

Bagian (Section)	Isi	Tujuan
Header/Navigasi	Nama/logo dan menu navigasi (Beranda, Tentang, Skill, Kontak)	Memudahkan pengunjung berpindah antar bagian halaman
Hero Section	Foto profil, nama, dan tagline singkat yang menarik perhatian	Kesan pertama — harus langsung menjelaskan siapa Anda dalam hitungan detik
Tentang Saya	Deskripsi singkat latar belakang dan minat	Membangun konteks dan kepercayaan (trust)
Keahlian/Skill	Daftar kemampuan teknis yang dimiliki	Menunjukkan kompetensi secara ringkas dan terstruktur

Bagian (Section)	Isi	Tujuan
Kontak	Form atau tautan untuk dihubungi (email, media sosial)	Mengonversi ketertarikan pengunjung menjadi tindakan nyata

11.3 Membangun Struktur HTML Landing Page

```

<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Budi Santoso - Portofolio</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <nav class="navbar">
    <span class="logo">Budi Santoso</span>
    <div class="nav-links">
      <a href="#tentang">Tentang</a>
      <a href="#skill">Skill</a>
      <a href="#kontak">Kontak</a>
    </div>
  </nav>

  <section class="hero">
    
    <h1>Budi Santoso</h1>
    <p class="tagline">Siswa TJKT | Calon Network Engineer</p>
  </section>

  <section id="tentang" class="tentang">
    <h2>Tentang Saya</h2>
    <p>Saya siswa kelas X TJKT yang tertarik pada jaringan komputer dan pengembangan web.</p>
  </section>

  <section id="skill" class="skill">
    <h2>Keahlian</h2>
    <ul>
      <li>Dasar Jaringan Komputer</li>
      <li>HTML, CSS, JavaScript</li>
      <li>Konfigurasi Cisco Packet Tracer</li>
    </ul>
  </section>

  <section id="kontak" class="kontak">
    <h2>Hubungi Saya</h2>
    <form id="form-kontak">
      <input type="text" id="nama" placeholder="Nama Anda" required>
      <input type="email" id="email" placeholder="Email Anda" required>
      <textarea id="pesan" placeholder="Pesan Anda" rows="4" required></textarea>
      <button type="submit">Kirim Pesan</button>
    </form>
    <p id="status-pesan"></p>
  </section>

```

```

</section>

<script src="script.js"></script>
</body>
</html>

```

11.4 Mempercantik dengan CSS

```

body {
  font-family: 'Segoe UI', sans-serif;
  margin: 0;
  color: #2E2E2E;
}

.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 16px 40px;
  background-color: #1F4E79;
  color: white;
}

.nav-links a {
  color: white;
  text-decoration: none;
  margin-left: 20px;
}

.hero {
  text-align: center;
  padding: 60px 20px;
  background-color: #F5F5F5;
}

.foto-profil {
  width: 150px;
  height: 150px;
  border-radius: 50%; /* membuat gambar menjadi bulat sempurna */
  object-fit: cover;
}

.tentang, .skill, .kontak {
  max-width: 700px;
  margin: 0 auto;
  padding: 40px 20px;
}

#form-kontak input, #form-kontak textarea {
  display: block;
  width: 100%;
  margin-bottom: 12px;
  padding: 10px;
  box-sizing: border-box;
  border: 1px solid #CCCCCC;
  border-radius: 4px;
}

```

```

}

#form-kontak button {
  background-color: #1F4E79;
  color: white;
  padding: 10px 24px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

```

Perhatikan penggunaan Flexbox pada .navbar (dipelajari pada Bab 9) untuk menyusun logo dan menu navigasi secara horizontal, serta properti border-radius: 50% pada .foto-profil yang mengubah gambar persegi menjadi bulat sempurna — sebuah trik CSS umum yang sering ditemui pada foto profil website nyata.

11.5 Menambahkan Interaktivitas dengan JavaScript

```

document.getElementById("form-kontak").addEventListener("submit", function(e) {
  e.preventDefault(); // mencegah halaman reload saat form dikirim

  let nama = document.getElementById("nama").value;
  let email = document.getElementById("email").value;
  let pesan = document.getElementById("pesan").value;

  if (nama === "" || email === "" || pesan === "") {
    document.getElementById("status-pesan").innerText = "Mohon lengkapi semua kolom!";
  } else {
    document.getElementById("status-pesan").innerText =
      "Terima kasih, " + nama + "! Pesan Anda telah diterima.";
    document.getElementById("form-kontak").reset(); // mengosongkan kembali form
  }
});

```

Method `e.preventDefault()` mencegah perilaku default form HTML yang secara otomatis memuat ulang halaman saat dikirim, memungkinkan JavaScript menangani proses pengiriman secara penuh — sebuah pola fundamental yang akan digunakan kembali ketika form ini disambungkan ke database sesungguhnya pada Bab 12.

Aktivitas Pembelajaran

84. Peserta didik membangun landing page profil pribadi lengkap mengikuti struktur pada bab ini, menggunakan data dan foto diri sendiri, disimpan dalam tiga file terpisah (index.html, style.css, script.js).
85. Peserta didik menambahkan minimal satu section tambahan pada landing page mereka (misalnya 'Pengalaman Organisasi' atau 'Prestasi') yang belum dicontohkan pada bab ini, menerapkan HTML dan CSS secara mandiri.
86. Peserta didik saling bertukar landing page dengan teman sekelas untuk saling memberikan umpan balik terkait kejelasan informasi dan kerapian tampilan.

Rangkuman

Landing page profil dibangun melalui tahap perancangan struktur informasi (hero, tentang, skill, kontak), implementasi struktur menggunakan HTML, mempercantik tampilan menggunakan CSS (termasuk Flexbox), dan menambahkan interaktivitas form menggunakan JavaScript dengan penanganan event submit dan validasi dasar. Proyek ini mengintegrasikan seluruh kompetensi HTML, CSS, dan JavaScript yang telah dipelajari pada bab-bab sebelumnya menjadi satu produk web yang utuh dan siap digunakan sebagai portofolio nyata.

Soal Latihan / Refleksi

87. Sebutkan lima bagian (section) umum yang terdapat pada sebuah landing page profil yang efektif.
88. Jelaskan fungsi method `e.preventDefault()` pada penanganan event submit form.
89. Mengapa memisahkan kode menjadi tiga file (HTML, CSS, JavaScript) dianggap sebagai praktik yang lebih baik dibanding menuliskan semuanya dalam satu file?
90. Jelaskan bagaimana Flexbox digunakan pada bagian navbar landing page yang telah dipelajari pada bab ini.

BAB 12 — Koneksi Database dan Form Login

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan konsep dasar interaksi antara halaman web, server, dan database.
- Peserta didik mampu membuat database dan tabel sederhana menggunakan MySQL.
- Peserta didik mampu membangun form login sederhana yang tersambung ke database menggunakan PHP.

12.1 Dari Website Statis Menuju Website Dinamis

Landing page yang dibangun pada Bab 11 bersifat statis — data yang ditampilkan tetap sama untuk setiap pengunjung, dan data yang dikirimkan melalui form kontak tidak benar-benar tersimpan di mana pun. Website dinamis, sebaliknya, dapat menyimpan dan mengambil data dari database, memungkinkan fitur seperti login pengguna, penyimpanan pesan kontak, atau sistem pendaftaran online yang sesungguhnya — sebagaimana hampir seluruh website interaktif yang digunakan sehari-hari (media sosial, e-commerce, sistem akademik sekolah).

Untuk membangun website dinamis, dibutuhkan komponen tambahan di luar HTML/CSS/JavaScript yang berjalan di browser (client-side): bahasa pemrograman **server-side** seperti PHP yang berjalan di server, serta **database** seperti MySQL yang menyimpan data secara permanen.

12.2 Alur Kerja Website Dinamis

91. Pengguna mengisi form login pada halaman HTML (client-side) dan menekan tombol submit.
92. Data form dikirim ke server menggunakan method POST menuju sebuah file PHP.
93. Skrip PHP (server-side) menerima data, memeriksanya terhadap data yang tersimpan pada database MySQL.
94. Jika data cocok, PHP mengarahkan pengguna ke halaman berikutnya (login berhasil); jika tidak, PHP menampilkan pesan kesalahan.
95. Hasil akhir dikirim kembali ke browser dan ditampilkan kepada pengguna.

Alur ini melibatkan tiga komponen utama yang bekerja sama: **Client** (browser dan HTML/CSS/JS yang dijalankan pengguna), **Server** (PHP yang memproses logika di sisi server), dan **Database** (MySQL yang menyimpan data secara permanen) — arsitektur yang dikenal sebagai model client-server dan menjadi dasar hampir seluruh aplikasi web modern.

12.3 Membuat Database dan Tabel dengan MySQL

Menggunakan XAMPP/Laragon yang telah disiapkan pada Bab 1, peserta didik dapat mengakses phpMyAdmin (antarmuka pengelolaan MySQL berbasis web) untuk membuat database baru dan tabel pengguna.

```

CREATE DATABASE db_portofolio;

USE db_portofolio;

CREATE TABLE pengguna (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL
);

INSERT INTO pengguna (username, password)
VALUES ('budi', '$2y$10$exampleHashedPasswordXXXXXXXXXXXXXXXXXXXXXXXXXXXX');

```

Perhatikan bahwa kolom password disimpan dalam bentuk hash (terenkripsi satu arah), bukan teks biasa (plaintext). Menyimpan password dalam bentuk plaintext merupakan praktik keamanan yang sangat buruk dan berbahaya — apabila database bocor, seluruh password pengguna akan langsung terekspos. PHP menyediakan fungsi bawaan `password_hash()` untuk membuat hash password secara aman.

12.4 Menghubungkan PHP ke Database MySQL

```

<?php
// koneksi.php
$host = "localhost";
$user = "root";
$pass = "";
$db   = "db_portofolio";

$koneksi = new mysqli($host, $user, $pass, $db);

if ($koneksi->connect_error) {
  die("Koneksi database gagal: " . $koneksi->connect_error);
}
?>

```

Kode di atas menggunakan ekstensi MySQLi (MySQL Improved) bawaan PHP untuk membuka koneksi ke server database. File koneksi.php ini akan digunakan kembali (di-include) pada skrip proses login berikutnya, mengikuti praktik pemrograman yang baik yaitu tidak mengulang kode yang sama di banyak tempat.

12.5 Membangun Form Login Lengkap

```

<!-- login.html -->
<form action="proses_login.php" method="POST">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>

  <label for="password">Password:</label>
  <input type="password" id="password" name="password" required>

  <button type="submit">Login</button>
</form>

```

```

<?php
// proses_login.php
include "koneksi.php";

$username = $_POST['username'];
$password = $_POST['password'];

$stmt = $koneksi->prepare("SELECT * FROM pengguna WHERE username = ?");
$stmt->bind_param("s", $username);
$stmt->execute();
$hasil = $stmt->get_result();

if ($hasil->num_rows === 1) {
    $data = $hasil->fetch_assoc();
    if (password_verify($password, $data['password'])) {
        echo "Login berhasil! Selamat datang, " . $data['username'];
    } else {
        echo "Password salah.";
    }
} else {
    echo "Username tidak ditemukan.";
}
?>

```

Perhatikan penggunaan **prepared statement** (`$koneksi->prepare()`) dengan tanda tanya sebagai placeholder, diisi menggunakan `bind_param()`) alih-alih menggabungkan langsung nilai input pengguna ke dalam query SQL. Pendekatan ini bukan sekadar gaya penulisan, melainkan lapisan keamanan penting yang mencegah serangan **SQL Injection** — teknik peretasan di mana penyerang menyisipkan perintah SQL berbahaya melalui kolom input form untuk memanipulasi atau mencuri data dari database.

12.6 Praktik Keamanan Dasar pada Form Login

Praktik	Alasan
Gunakan <code>password_hash()</code> dan <code>password_verify()</code>	Password tidak pernah disimpan dalam bentuk asli yang dapat dibaca langsung
Gunakan prepared statement	Mencegah serangan SQL Injection melalui kolom input
Validasi input di sisi server, bukan hanya browser	Validasi client-side (atribut required pada Bab 8) dapat dilewati oleh pengguna yang mengubah kode HTML secara manual
Jangan tampilkan pesan error yang terlalu detail	Pesan seperti "Username tidak ditemukan" sebaiknya diseragamkan menjadi "Username atau password salah" agar tidak membocorkan informasi kepada penyerang

Aktivitas Pembelajaran

96. Peserta didik menginstal XAMPP/Laragon, mengaktifkan Apache dan MySQL, kemudian membuat database dan tabel pengguna melalui phpMyAdmin sesuai contoh pada bab ini.

97. Peserta didik membuat form login HTML dan skrip PHP `proses_login.php` yang tersambung ke database, kemudian menguji login menggunakan akun yang telah dibuat.
98. Peserta didik memodifikasi tabel pengguna untuk menambahkan kolom email, kemudian memperbarui form pendaftaran (registrasi) baru yang menyimpan data pengguna baru ke database menggunakan `password_hash()`.
99. Peserta didik mengintegrasikan form login yang telah dibuat ke dalam landing page dari Bab 11, misalnya sebagai halaman 'Admin Login' terpisah.

Rangkuman

Website dinamis melibatkan tiga komponen: client (HTML/CSS/JS), server (PHP), dan database (MySQL) yang bekerja sama melalui alur pengiriman form, pemrosesan di server, dan penyimpanan/pengambilan data. Form login yang aman menerapkan hashing password menggunakan `password_hash()`/`password_verify()`, prepared statement untuk mencegah SQL Injection, dan validasi di sisi server. Kompetensi ini melengkapi kemampuan peserta didik membangun website yang benar-benar fungsional, bukan sekadar tampilan statis.

Soal Latihan / Refleksi

100. Jelaskan perbedaan antara website statis dan website dinamis, beserta contoh masing-masing.
101. Mengapa password sebaiknya disimpan dalam bentuk hash, bukan plaintext, di dalam database?
102. Jelaskan apa itu SQL Injection dan bagaimana prepared statement dapat mencegahnya.
103. Sebutkan tiga komponen utama dalam arsitektur client-server pada website dinamis beserta peran masing-masing.

BAB 13 — Pengenalan Vibecoding: Pemrograman Berbantuan AI

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan konsep vibecoding dan bagaimana AI mengubah cara kerja pengembang web.
- Peserta didik mampu menggunakan asisten AI secara efektif untuk membantu menulis dan memahami kode.
- Peserta didik mampu menerapkan sikap kritis dan etis dalam memanfaatkan AI untuk pemrograman.

13.1 Apa Itu Vibecoding?

Vibecoding adalah istilah yang mulai populer sejak pertengahan dekade 2020-an untuk menggambarkan gaya pengembangan perangkat lunak di mana programmer berkolaborasi secara intensif dengan asisten kecerdasan buatan (AI) — mendeskripsikan apa yang ingin dibangun menggunakan bahasa alami, kemudian AI menghasilkan sebagian besar atau seluruh kode, sementara programmer berperan mengarahkan, meninjau, menguji, dan menyempurnakan hasil tersebut. Istilah 'vibe' merujuk pada pendekatan yang lebih mengalir dan eksploratif — programmer berfokus pada tujuan akhir dan 'rasa' (vibe) dari produk yang diinginkan, alih-alih menulis setiap baris kode secara manual dari nol.

Penting dipahami bahwa vibecoding bukan berarti pemrograman tanpa perlu memahami logika atau kode sama sekali — justru sebaliknya, peserta didik yang telah menguasai logika pemrograman (Bab 2–6) dan dasar HTML/CSS/JavaScript (Bab 7–10) akan jauh lebih mampu memanfaatkan AI secara efektif, karena mereka dapat membaca, memahami, mengevaluasi, dan memperbaiki kode yang dihasilkan AI — sebuah kemampuan yang mutlak diperlukan karena AI tidak selalu menghasilkan kode yang benar atau optimal.

13.2 Mengapa Vibecoding Relevan bagi Siswa TJKT Saat Ini

Industri teknologi informasi mengalami transformasi signifikan sejak asisten AI pemrograman seperti GitHub Copilot, Cursor, dan Claude menjadi alat yang lazim digunakan oleh pengembang profesional di seluruh dunia. Bagi siswa TJKT yang kelak akan bekerja di bidang jaringan maupun pengembangan sistem, kemampuan berkolaborasi secara efektif dengan AI bukan lagi sekadar nilai tambah, melainkan keterampilan dasar yang semakin dianggap setara pentingnya dengan kemampuan mengetik cepat pada dekade-dekade sebelumnya.

Sebagai contoh relevansi langsung: seorang teknisi jaringan yang perlu membuat skrip otomatisasi konfigurasi, dashboard monitoring sederhana, atau bahkan memperbaiki bug pada halaman web internal perusahaan, dapat bekerja jauh lebih cepat dan efisien dengan memanfaatkan AI sebagai partner kerja, asalkan ia memiliki fondasi pemahaman teknis yang cukup untuk mengarahkan dan memvalidasi hasil kerja AI tersebut.

13.3 Cara Berkomunikasi Efektif dengan Asisten AI

Kualitas hasil yang diberikan AI sangat bergantung pada kualitas instruksi (prompt) yang diberikan. Prinsip ini serupa dengan penyusunan algoritma yang baik pada Bab 2 — instruksi yang jelas, spesifik, dan tidak ambigu akan menghasilkan hasil yang jauh lebih baik dibanding instruksi yang samar.

Prompt Kurang Efektif	Prompt Lebih Efektif
"Buatkan form"	"Buatkan form HTML untuk pendaftaran webinar dengan kolom nama, email, dan pilihan sesi (pagi/siang), lengkap dengan validasi required"
"Kenapa kode ini error?"	"Kode JavaScript ini menghasilkan error 'Cannot read property of undefined' pada baris 12 saat tombol diklik — tolong jelaskan penyebabnya dan cara memperbaikinya"
"Buat website toko"	"Buatkan landing page HTML/CSS sederhana untuk toko kue online dengan bagian hero, daftar produk (3 kartu), dan form kontak, menggunakan skema warna coklat dan krem"

Pola umum yang efektif dalam menyusun prompt untuk keperluan pemrograman mencakup: menjelaskan **konteks** (apa proyek yang sedang dikerjakan), **tujuan spesifik** (apa yang ingin dicapai dari permintaan ini), dan **batasan/preferensi** (bahasa yang digunakan, gaya kode, atau fitur yang tidak diinginkan) — sebuah kerangka yang sejalan dengan kebiasaan menyusun kebutuhan (requirement) yang jelas sebelum memulai pekerjaan apa pun.

13.4 Sikap Kritis: AI sebagai Partner, Bukan Pengganti Pemahaman

Bahaya terbesar dari penggunaan AI secara tidak bijak dalam belajar pemrograman adalah ketergantungan tanpa pemahaman (copy-paste tanpa mengerti), yang pada akhirnya justru menghambat perkembangan kompetensi peserta didik. AI dapat menghasilkan kode yang terlihat benar namun sesungguhnya mengandung kesalahan logika, celah keamanan (seperti yang telah dipelajari terkait SQL Injection pada Bab 12), atau tidak sesuai dengan praktik terbaik — dan seseorang yang tidak memahami dasar pemrograman tidak akan mampu mengenali kesalahan tersebut.

- Selalu baca dan pahami kode yang dihasilkan AI sebelum menggunakannya, jangan sekadar menyalin-tempel (copy-paste) tanpa mengerti.
- Uji kode hasil AI secara menyeluruh — AI dapat membuat kesalahan, terutama pada logika yang kompleks atau kasus khusus (edge case).
- Gunakan AI untuk mempercepat pekerjaan rutin atau sebagai alat bantu belajar (misalnya meminta penjelasan konsep), bukan sebagai pengganti proses belajar logika pemrograman itu sendiri.
- Tetap kembangkan kemampuan menulis kode secara manual, karena kemampuan inilah yang memungkinkan seseorang menjadi pengarah (director) yang baik bagi AI, bukan sekadar operator yang pasif.

13.5 Etika Penggunaan AI dalam Konteks Akademik

Dalam konteks tugas sekolah, penggunaan AI perlu diimbangi dengan kejujuran akademik. Peserta didik sebaiknya menggunakan AI sebagai alat bantu belajar dan mempercepat eksplorasi, bukan sebagai jalan pintas untuk menghindari proses belajar yang sesungguhnya, mengingat tujuan utama mata pelajaran ini adalah membentuk kemampuan berpikir logis dan memecahkan masalah — kemampuan yang tidak dapat 'dipinjam' begitu saja dari AI tanpa proses internalisasi oleh peserta didik itu sendiri. Guru dapat mempertimbangkan kebijakan transparansi, misalnya meminta peserta didik mencantumkan bagian mana dari tugas yang dibantu AI beserta penjelasan pemahaman mereka terhadap kode tersebut.

Aktivitas Pembelajaran

104. Peserta didik mencoba meminta bantuan sebuah asisten AI (dengan pengawasan guru) untuk membuat sebuah komponen kecil pada landing page mereka (misalnya galeri foto sederhana), kemudian membandingkan dua versi prompt (kurang spesifik vs. spesifik) dan mengamati perbedaan kualitas hasilnya.
105. Peserta didik meminta AI menjelaskan sebuah potongan kode JavaScript yang belum mereka pahami sepenuhnya dari bab-bab sebelumnya, kemudian menuliskan pemahaman mereka sendiri dengan kata-kata sendiri sebagai bukti internalisasi konsep.
106. Diskusi kelas: peserta didik dengan sengaja meminta AI membuat sebuah fungsi sederhana, kemudian secara berkelompok mencoba menemukan potensi kesalahan atau kekurangan pada kode yang dihasilkan (jika ada), melatih sikap kritis terhadap keluaran AI.

Rangkuman

Vibecoding menggambarkan kolaborasi intensif antara programmer dan AI dalam proses pengembangan perangkat lunak, di mana AI menghasilkan kode berdasarkan instruksi bahasa alami sementara programmer mengarahkan dan memvalidasi hasilnya. Pemahaman fondasi logika pemrograman dan web yang telah dipelajari sepanjang buku ini menjadi prasyarat penting agar peserta didik dapat memanfaatkan AI secara efektif dan kritis, bukan sekadar bergantung tanpa pemahaman. Prinsip komunikasi prompt yang jelas dan etika akademik menjadi bagian penting dari kompetensi ini.

Soal Latihan / Refleksi

107. Jelaskan apa yang dimaksud dengan vibecoding dan mengapa pemahaman logika pemrograman tetap penting meskipun menggunakan AI.
108. Sebutkan tiga elemen yang sebaiknya terdapat dalam sebuah prompt yang efektif untuk meminta bantuan AI menulis kode.
109. Jelaskan risiko yang dapat terjadi apabila seseorang menggunakan kode hasil AI tanpa memahaminya terlebih dahulu.
110. Bagaimana sikap yang tepat dalam menggunakan AI untuk mengerjakan tugas sekolah agar tetap menjunjung kejujuran akademik?

BAB 14 — Menyiapkan dan Menerbitkan Aplikasi Web di Ubuntu Server

Tujuan Pembelajaran

- Peserta didik mampu menjelaskan perbedaan menjalankan aplikasi web secara lokal dengan menerbitkannya (deploy) di server nyata.
- Peserta didik mampu menginstal dan mengonfigurasi web server Apache, PHP, dan MySQL pada Ubuntu Server.
- Peserta didik mampu memindahkan (upload) proyek landing page dan sistem login ke server serta mengaksesnya melalui jaringan.
- Peserta didik mampu menerapkan pengamanan dasar pada server yang menjalankan aplikasi web.

14.1 Dari Localhost Menuju Server Sungguhan

Sepanjang Bab 11 dan 12, proyek landing page dan sistem login dijalankan menggunakan XAMPP/Laragon secara lokal (localhost) — hanya dapat diakses dari komputer sendiri. Agar sebuah website dapat diakses oleh siapa pun melalui internet atau jaringan sekolah, proyek tersebut perlu **diterbitkan (deploy)** ke sebuah server sungguhan yang menyala dan terhubung ke jaringan selama 24 jam. Bab ini memperkenalkan proses tersebut menggunakan **Ubuntu Server**, salah satu distribusi Linux paling populer untuk keperluan hosting web, sekaligus menjadi titik temu yang sangat relevan antara kompetensi pemrograman web dan kompetensi jaringan/Linux yang menjadi inti Program Keahlian TJKT.

Peserta didik yang telah mempelajari dasar perintah Linux pada mata pelajaran lain (atau akan mempelajarinya pada mata pelajaran Keamanan Jaringan dan Administrasi Sistem Jaringan) akan menemukan bahwa kemampuan tersebut langsung terpakai secara nyata dalam proses deployment aplikasi web, membuktikan bahwa kompetensi jaringan dan pemrograman web saling melengkapi, bukan dua bidang yang terpisah.

14.2 Menyiapkan Server: LAMP Stack

LAMP adalah singkatan dari **L**inux, **A**pache, **M**ySQL, dan **P**HP — empat komponen perangkat lunak yang bekerja sama menjadi fondasi paling umum untuk menjalankan aplikasi web dinamis, dan menjadi standar industri yang telah digunakan puluhan tahun. Berikut langkah instalasi LAMP stack pada Ubuntu Server yang masih baru diinstal.

```
# Perbarui daftar paket sistem
sudo apt update && sudo apt upgrade -y

# Instal Apache (web server)
sudo apt install apache2 -y
sudo systemctl enable apache2
sudo systemctl start apache2
```

```
# Instal MySQL (database server)
sudo apt install mysql-server -y
sudo mysql_secure_installation

# Instal PHP beserta modul pendukung Apache dan MySQL
sudo apt install php libapache2-mod-php php-mysql -y

# Restart Apache agar modul PHP aktif
sudo systemctl restart apache2
```

Perhatikan bahwa perintah `mysql\_secure\_installation` yang digunakan di atas persis sama dengan langkah pengamanan basis data yang telah dipelajari secara mendalam pada mata pelajaran Keamanan Jaringan — sebuah bukti nyata bagaimana kompetensi antar mata pelajaran saling terhubung dan digunakan berulang dalam konteks berbeda.

14.3 Memverifikasi Instalasi

Perintah	Fungsi
<code>sudo systemctl status apache2</code>	Memastikan layanan Apache berjalan (status: active/running)
<code>php -v</code>	Menampilkan versi PHP yang terinstal untuk memastikan instalasi berhasil
<code>sudo mysql -u root -p</code>	Masuk ke MySQL untuk memastikan layanan database dapat diakses
<code>curl http://localhost</code>	Menguji apakah Apache merespons permintaan HTTP dari command line server itu sendiri

Setelah instalasi berhasil, mengakses alamat IP server (misalnya `http://192.168.1.50`) dari browser komputer lain pada jaringan yang sama seharusnya menampilkan halaman default Apache ('Apache2 Ubuntu Default Page'), menandakan web server telah aktif dan siap menerima permintaan.

14.4 Memindahkan (Upload) Proyek ke Server

Direktori default tempat Apache menyajikan file website pada Ubuntu Server adalah `/var/www/html/`. Terdapat beberapa cara memindahkan proyek landing page dari komputer lokal ke direktori tersebut, tergantung apakah peserta didik bekerja langsung di server (misalnya melalui mesin virtual) atau memindahkan file dari komputer terpisah.

Metode	Perintah/Cara	Kapan Digunakan
SCP (Secure Copy)	<code>scp -r ./proyek-web user@192.168.1.50:/var/www/html/</code>	Memindahkan file dari komputer lokal ke server melalui SSH

Metode	Perintah/Cara	Kapan Digunakan
Git	<code>git clone https://github.com/user/proyek-web.git</code>	Jika proyek telah disimpan pada repositori Git (praktik profesional yang direkomendasikan)
Kerja Langsung di Server	Mengedit file langsung menggunakan nano/vim pada sesi SSH	Untuk perubahan kecil atau saat belajar langsung di mesin virtual server

```
# Contoh memindahkan folder proyek dari komputer lokal ke server menggunakan SCP
scp -r ./landing-page-profil user@192.168.1.50:/tmp/

# Login ke server melalui SSH, lalu pindahkan ke direktori web
ssh user@192.168.1.50
sudo mv /tmp/landing-page-profil/* /var/www/html/
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
```

Perintah ``chown -R www-data:www-data`` mengubah kepemilikan seluruh file menjadi milik user ``www-data``, yaitu user khusus yang digunakan Apache untuk menjalankan proses web server — langkah penting agar Apache memiliki izin membaca (dan pada kasus tertentu menulis) file-file tersebut tanpa perlu menjalankan Apache dengan hak akses root yang berisiko.

14.5 Mengonfigurasi Virtual Host

Virtual Host memungkinkan satu server Apache menjalankan beberapa website berbeda sekaligus, masing-masing diakses melalui domain atau subdomain yang berbeda pula — sangat berguna ketika sebuah sekolah ingin menghosting beberapa proyek siswa pada satu server yang sama.

```
# Membuat file konfigurasi virtual host baru
sudo nano /etc/apache2/sites-available/portofolio-budi.conf

<VirtualHost *:80>
    ServerName portofolio-budi.local
    DocumentRoot /var/www/html/portofolio-budi
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

# Mengaktifkan virtual host dan memuat ulang Apache
sudo a2ensite portofolio-budi.conf
sudo systemctl reload apache2
```

14.6 Menyambungkan Aplikasi ke Database di Server

Konfigurasi database dan tabel yang telah dipraktikkan pada Bab 12 perlu diulang pada server MySQL yang baru diinstal ini, karena data pada XAMPP/Laragon lokal tidak otomatis ikut berpindah. Peserta didik perlu menjalankan kembali perintah ``CREATE DATABASE`` dan ``CREATE TABLE`` pada server, kemudian memperbarui file ``koneksi.php`` sesuai kredensial database pada server tersebut.

```

<?php
// koneksi.php (versi server, kredensial disesuaikan)
$host = "localhost";          // tetap localhost karena PHP & MySQL berjalan di server
yang sama
$user = "user_web";           // gunakan user database khusus, bukan root, sesuai
praktik keamanan Bab 12
$pass = "PasswordAman!23";
$db   = "db_portofolio";

$koneksi = new mysqli($host, $user, $pass, $db);

if ($koneksi->connect_error) {
    die("Koneksi database gagal: " . $koneksi->connect_error);
}
?>

```

Perhatikan penggunaan user database khusus (`user\_web`) dengan hak akses terbatas, alih-alih akun `root`, sejalan dengan prinsip least privilege yang telah dipelajari pada mata pelajaran Keamanan Jaringan — sebuah praktik yang seharusnya diterapkan konsisten setiap kali aplikasi web terhubung ke database produksi.

14.7 Pengamanan Dasar Server Web

Server yang menjalankan aplikasi web dan dapat diakses dari jaringan luar memerlukan langkah pengamanan dasar, memperluas kompetensi hardening yang telah dipelajari secara mendalam pada mata pelajaran Keamanan Jaringan ke konteks server web secara spesifik.

Langkah Pengamanan	Perintah
Aktifkan firewall, izinkan hanya port esensial	<code>sudo ufw allow 'Apache Full' && sudo ufw allow OpenSSH && sudo ufw enable</code>
Nonaktifkan direktori listing Apache	Tambahkan Options -Indexes pada konfigurasi virtual host
Perbarui sistem dan paket secara berkala	<code>sudo apt update && sudo apt upgrade -y</code>
Gunakan HTTPS (bukan HTTP biasa)	<code>sudo apt install certbot python3-certbot-apache lalu sudo certbot --apache</code>

Sertifikat HTTPS (menggunakan Certbot dari Let's Encrypt, sebuah otoritas sertifikat gratis) memastikan seluruh data yang dikirim antara pengguna dan server — termasuk data form login yang sensitif — dikirim dalam bentuk terenkripsi, mencegah penyadapan data oleh pihak yang tidak berwenang selama proses transmisi, sebuah lapisan keamanan tambahan yang melengkapi hashing password yang telah diterapkan pada Bab 12.

14.8 Troubleshooting Masalah Umum saat Deployment

Proses deployment sering kali tidak berjalan mulus pada percobaan pertama, bahkan bagi pengembang berpengalaman sekalipun. Kemampuan menelusuri dan memperbaiki masalah (troubleshooting) secara sistematis — mengingat kembali pendekatan berlapis (layered troubleshooting) yang relevan dengan model OSI pada mata pelajaran Jaringan Dasar — menjadi keterampilan yang sama pentingnya dengan kemampuan melakukan konfigurasi itu sendiri.

Gejala	Kemungkinan Penyebab	Langkah Pemeriksaan
Halaman menampilkan error 403 Forbidden	Hak akses file/direktori tidak tepat	Periksa kembali hasil chown dan chmod pada direktori proyek
Halaman menampilkan error 500 Internal Server Error	Kesalahan sintaks pada kode PHP	Periksa log Apache: <code>sudo tail -f /var/log/apache2/error.log</code>
Website tidak dapat diakses sama sekali dari komputer lain	Firewall memblokir port, atau Apache tidak berjalan	Periksa <code>sudo ufw status</code> dan <code>sudo systemctl status apache2</code>
Form login menampilkan error koneksi database	Kredensial pada koneksi.php tidak sesuai dengan server	Periksa kembali host, user, password, dan nama database pada file konfigurasi
Perubahan pada file tidak terlihat di browser	Cache browser menyimpan versi lama halaman	Lakukan hard refresh (Ctrl+Shift+R) atau bersihkan cache browser

Log Apache (`/var/log/apache2/error.log`) menjadi sumber informasi paling berharga saat troubleshooting, karena hampir seluruh kesalahan pada sisi server — baik kesalahan konfigurasi Apache maupun kesalahan sintaks PHP — akan tercatat di sana secara rinci, sebuah kebiasaan memeriksa log yang sejalan dengan kompetensi analisis log yang telah dipelajari secara mendalam pada mata pelajaran Keamanan Jaringan.

Aktivitas Pembelajaran

111. Peserta didik menyiapkan sebuah mesin virtual Ubuntu Server (menggunakan VirtualBox atau sejenisnya) sebagai lingkungan praktik, kemudian instal LAMP stack lengkap mengikuti langkah-langkah pada bab ini.
112. Peserta didik memindahkan proyek landing page profil (Bab 11) ke direktori `/var/www/html/` pada mesin virtual tersebut, kemudian mengaksesnya dari browser komputer host menggunakan alamat IP mesin virtual.
113. Peserta didik memindahkan ulang database dan sistem login (Bab 12) ke MySQL pada server, memperbarui file koneksi.php, dan menguji proses login berfungsi dengan baik pada lingkungan server yang baru.
114. Peserta didik menerapkan firewall UFW pada mesin virtual server, mengizinkan hanya port yang diperlukan (SSH dan HTTP/HTTPS), kemudian memverifikasi website tetap dapat diakses setelah firewall diaktifkan.

Rangkuman

Menerbitkan (deploy) aplikasi web ke Ubuntu Server melibatkan instalasi LAMP stack (Linux, Apache, MySQL, PHP), pemindahan file proyek ke direktori web server, konfigurasi virtual host, penyesuaian koneksi database pada lingkungan server, serta penerapan pengamanan dasar seperti firewall dan HTTPS. Bab ini menjadi titik temu nyata antara kompetensi pemrograman web dan kompetensi jaringan/Linux yang menjadi inti Program Keahlian TJKT, membuktikan bahwa kedua bidang tersebut saling melengkapi dalam praktik dunia kerja yang sesungguhnya.

Soal Latihan / Refleksi

115. Jelaskan kepanjangan LAMP dan fungsi masing-masing komponennya dalam menjalankan aplikasi web.
116. Mengapa kepemilikan file pada direktori `/var/www/html/` perlu diubah menjadi `www-data` setelah proyek dipindahkan ke server?
117. Jelaskan fungsi Virtual Host pada Apache dan manfaatnya ketika sebuah server perlu menghosting beberapa proyek sekaligus.
118. Sebutkan tiga langkah pengamanan dasar yang sebaiknya diterapkan pada server web yang dapat diakses dari jaringan luar.
119. Mengapa penggunaan HTTPS penting khususnya untuk halaman yang memuat form login?

BAB 15 — Rangkuman dan Proyek Akhir

Bab ini menjadi bahan konsolidasi menyeluruh atas seluruh materi Pemrograman Web Dasar yang telah dipelajari, mencakup logika pemrograman (Bab 2–6), dasar pengembangan web (Bab 7–10), praktik nyata landing page, database, dan deployment server (Bab 11–14), serta pengenalan vibecoding (Bab 13). Bab ini dapat digunakan sebagai bahan review menjelang asesmen sumatif maupun sebagai panduan proyek akhir semester.

14.1 Peta Keterkaitan Antar Bab

Materi dalam buku ini disusun secara berjenjang dan saling terhubung erat. Konsep logika (Bab 2), flowchart/pseudocode (Bab 3), percabangan (Bab 4), perulangan (Bab 5), serta variabel/operator (Bab 6) membentuk fondasi cara berpikir sistematis yang kemudian diterapkan langsung ke dalam bahasa JavaScript. HTML (Bab 7–8) dan CSS (Bab 9) membentuk struktur dan tampilan visual, sementara JavaScript (Bab 10) menyatukan logika pemrograman dengan struktur web untuk menciptakan interaktivitas.

Bab 11 (landing page) mengintegrasikan seluruh kompetensi HTML/CSS/JavaScript menjadi satu proyek nyata, Bab 12 (database dan login) memperluasnya dengan kemampuan menyimpan dan mengambil data secara dinamis menggunakan PHP dan MySQL, dan Bab 14 (deployment di Ubuntu Server) melengkapi keduanya dengan kemampuan menerbitkan (publish) proyek tersebut agar dapat diakses secara nyata melalui internet. Bab 13 (vibecoding) melengkapi seluruh kompetensi teknis tersebut dengan keterampilan kolaborasi bersama AI yang semakin relevan dalam praktik industri masa kini.

14.2 Rekapitulasi Capaian Pembelajaran

Bagian	Bab	Kompetensi Utama
Logika Pemrograman	2	Konsep logika dan algoritma dengan analogi kehidupan nyata
Logika Pemrograman	3	Flowchart dan pseudocode sebagai alat perencanaan
Logika Pemrograman	4	Struktur percabangan (if, if-else, if-else if)
Logika Pemrograman	5	Struktur perulangan (for, while, nested loop)
Logika Pemrograman	6	Variabel, tipe data, dan operator
Dasar Web	7	Struktur dasar HTML dan tag umum
Dasar Web	8	Tabel HTML dan form dengan berbagai jenis input

Bagian	Bab	Kompetensi Utama
Dasar Web	9	CSS: selector, styling, dan layout Flexbox
Dasar Web	10	JavaScript: DOM dan event handling
Praktik Nyata	11	Membangun landing page profil terintegrasi
Praktik Nyata	12	Koneksi database dan form login aman
Praktik Nyata	14	Setup dan deployment aplikasi web di Ubuntu Server
Masa Depan	13	Vibecoding dan kolaborasi efektif dengan AI

14.3 Proyek Akhir: Landing Page Profil dengan Sistem Login

Sebagai proyek akhir yang mengintegrasikan seluruh kompetensi buku ini, peserta didik diminta mengembangkan lebih lanjut landing page yang telah dibangun pada Bab 11 dengan menambahkan sistem login sederhana dari Bab 12, menghasilkan sebuah produk web yang lengkap dan dapat dipresentasikan sebagai portofolio nyata.

120. Perbarui landing page profil pribadi (Bab 11) dengan konten yang lebih lengkap dan tampilan yang telah disempurnakan.
121. Tambahkan halaman 'Admin Login' terpisah yang tersambung ke database MySQL, mengikuti praktik keamanan yang telah dipelajari pada Bab 12 (password hashing, prepared statement).
122. Tambahkan sebuah halaman 'Dashboard' sederhana yang hanya dapat diakses setelah login berhasil, menampilkan pesan sapaan personal menggunakan data dari database.
123. Unggah (deploy) seluruh proyek ke sebuah server Ubuntu Server — baik server fisik/virtual laboratorium sekolah maupun mesin virtual pribadi — mengikuti langkah-langkah pada Bab 14, sehingga proyek dapat diakses melalui alamat IP atau domain server tersebut, bukan hanya berjalan secara lokal di komputer sendiri.
124. Dokumentasikan proses pengerjaan, termasuk bagian mana yang dibantu menggunakan AI (vibecoding) beserta penjelasan pemahaman pribadi terhadap kode tersebut, sejalan dengan prinsip etika akademik pada Bab 13.
125. Presentasikan hasil proyek di depan kelas, menjelaskan struktur HTML, styling CSS, logika JavaScript, serta alur koneksi database yang telah dibangun.

14.4 Arah Pembelajaran Lanjutan

Setelah menuntaskan buku ini, peserta didik memiliki fondasi kuat untuk melanjutkan eksplorasi pengembangan web maupun mengaitkannya kembali dengan kompetensi jaringan yang menjadi fokus utama Program Keahlian TJKT — misalnya membangun dashboard monitoring jaringan sederhana, halaman konfigurasi perangkat berbasis web, atau portofolio digital profesional yang mengintegrasikan

kompetensi jaringan dan web sekaligus. Peserta didik yang tertarik mendalami pengembangan web lebih lanjut dapat mempelajari framework modern seperti Bootstrap (mempercantik CSS dengan lebih cepat), atau bahasa/framework backend lain di luar PHP seperti Node.js, sebagai pengembangan kompetensi lanjutan.

Soal Latihan / Refleksi

126. Jelaskan bagaimana konsep percabangan (Bab 4) dan perulangan (Bab 5) diterapkan secara nyata pada studi kasus kalkulator luas di Bab 10.
127. Sebuah sekolah ingin membuat halaman pendaftaran ekstrakurikuler online yang menyimpan data pendaftar ke database. Jelaskan alur kerja lengkap dari pengisian form hingga data tersimpan, mengacu pada arsitektur client-server yang telah dipelajari.
128. Jelaskan mengapa pemahaman HTML, CSS, dan JavaScript tetap penting bagi seorang siswa TJKT meskipun AI (vibecoding) kini dapat membantu menulis sebagian besar kode.
129. Rancang sebuah landing page sederhana (deskripsikan section-section yang akan dibuat) untuk mempromosikan sebuah kegiatan pameran karya siswa TJKT di sekolah, lengkap dengan alasan pemilihan setiap section.
130. Refleksikan: dari seluruh materi buku ini (logika, HTML, CSS, JavaScript, database, vibecoding), bagian mana yang menurutmu paling menantang, dan bagaimana strategi mempelajarinya lebih lanjut?

Glosarium

Algoritma — urutan langkah logis dan sistematis untuk menyelesaikan suatu masalah.

API — Application Programming Interface, cara program berkomunikasi dengan program lain.

Array — tipe data yang menyimpan kumpulan/daftar nilai dalam satu variabel.

Client-side — kode yang dijalankan di browser pengguna (HTML, CSS, JavaScript).

CSS — Cascading Style Sheets, bahasa untuk mengatur tampilan visual halaman web.

DOM — Document Object Model, representasi struktur HTML yang dapat dimanipulasi JavaScript.

Event — aksi yang dilakukan pengguna terhadap halaman web, seperti klik atau input.

Flowchart — diagram alir yang memvisualisasikan alur logika program menggunakan simbol standar.

Flexbox — sistem layout CSS modern untuk menyusun elemen secara fleksibel.

Hashing — proses mengubah data (seperti password) menjadi bentuk terenkripsi satu arah.

Firewall (UFW) — perangkat lunak pengontrol lalu lintas jaringan pada server, dibahas dalam konteks pengamanan dasar server web.

HTML — HyperText Markup Language, bahasa markup untuk struktur dokumen web.

Landing Page — halaman web tunggal yang dirancang memperkenalkan sesuatu secara ringkas.

LAMP Stack — kombinasi Linux, Apache, MySQL, dan PHP sebagai fondasi umum menjalankan aplikasi web.

Looping/Perulangan — struktur kontrol yang menjalankan instruksi berulang kali.

MySQL — sistem manajemen basis data relasional yang populer untuk aplikasi web.

Percabangan — struktur kontrol yang memilih jalur eksekusi berdasarkan suatu kondisi.

PHP — bahasa pemrograman server-side yang umum digunakan bersama MySQL.

Prepared Statement — teknik query database yang mencegah serangan SQL Injection.

Pseudocode — penulisan algoritma dengan bahasa semi-formal, tidak terikat sintaks bahasa tertentu.

Selector (CSS) — penanda yang menentukan elemen HTML mana yang akan distyling.

Server-side — kode yang dijalankan di server, bukan di browser pengguna (seperti PHP).

SQL Injection — teknik serangan yang menyisipkan perintah SQL berbahaya melalui input form.

Tipe Data — kategori nilai yang dapat disimpan variabel (Number, String, Boolean, dsb.).

Variabel — wadah penyimpanan data bernama yang nilainya dapat berubah.

Vibecoding — gaya pengembangan perangkat lunak berbasis kolaborasi intensif dengan AI.

Virtual Host — konfigurasi Apache yang memungkinkan satu server menjalankan beberapa website berbeda.

Lampiran: Referensi Cepat Sintaks JavaScript

Kategori	Sintaks	Fungsi
Variabel	<code>let x = 5; const y = 10;</code>	Mendeklarasikan variabel yang dapat/tidak dapat diubah
Percabangan	<code>if (kondisi) {...} else {...}</code>	Menjalankan blok kode berdasarkan kondisi
Perulangan	<code>for (let i=0; i<n; i++) {...}</code>	Mengulang blok kode sebanyak n kali
Perulangan	<code>while (kondisi) {...}</code>	Mengulang blok kode selama kondisi benar
Fungsi	<code>function nama(param) {...}</code>	Mendefinisikan blok kode yang dapat dipanggil ulang
DOM	<code>document.getElementById("id")</code>	Mengambil elemen HTML berdasarkan id
DOM	<code>document.querySelector("selector")</code>	Mengambil elemen HTML berdasarkan selector CSS
DOM	<code>elemen.innerText = "teks"</code>	Mengubah isi teks sebuah elemen
Event	<code>elemen.addEventListener("click", fn)</code>	Menjalankan fungsi saat elemen diklik
Array	<code>let arr = [1,2,3]; arr.push(4);</code>	Membuat array dan menambahkan elemen baru
Operator	<code>===, !==, &&, , !</code>	Operator perbandingan dan logika

Lampiran: Referensi Cepat Tag HTML dan Properti CSS

Tag HTML yang Sering Digunakan

Tag	Fungsi
<h1>...<h6>	Heading/judul dengan enam tingkat hierarki
<p>	Paragraf teks
	Tautan (hyperlink)
	Menampilkan gambar
/ /	Daftar tidak berurutan/berurutan
<div>/	Elemen pembungkus generik (blok/inline)
<table>/<tr>/<th>/<td>	Struktur tabel
<form>/<input>/<button>	Form dan elemen input
<nav>/<section>/<header>/<footer>	Elemen struktural semantik HTML5

Properti CSS yang Sering Digunakan

Properti	Fungsi
color	Warna teks
background-color	Warna latar belakang
font-size / font-family	Ukuran dan jenis huruf
padding / margin	Jarak dalam/luar elemen
display: flex	Mengaktifkan layout Flexbox
justify-content / align-items	Perataan elemen pada Flexbox
border / border-radius	Garis tepi dan sudut melengkung
width / height	Lebar dan tinggi elemen

Lampiran: Proyek Studi Kasus Tambahan

Lampiran ini menyediakan tiga ide proyek tambahan yang dapat dikerjakan peserta didik sebagai latihan mandiri atau tugas pengayaan setelah menyelesaikan materi utama, masing-masing mengintegrasikan kompetensi logika, HTML, CSS, dan JavaScript.

Proyek A: Aplikasi Pengingat Tugas Sekolah (To-Do List)

Bangun sebuah halaman web sederhana yang memungkinkan pengguna menambahkan daftar tugas sekolah, menandainya sebagai selesai, dan menghapusnya dari daftar. Proyek ini melatih penggunaan array (Bab 6), manipulasi DOM (Bab 10) untuk menambah/menghapus elemen secara dinamis, dan event handling untuk tombol tambah/hapus/selesai.

Proyek B: Kalkulator Nilai Rata-rata dan Predikat

Bangun sebuah kalkulator yang menerima input beberapa nilai ujian (menggunakan perulangan atau array), menghitung rata-ratanya, kemudian menampilkan predikat (A/B/C/D) menggunakan struktur percabangan bertingkat sebagaimana dipelajari pada Bab 4. Tambahkan validasi agar nilai yang dimasukkan berada pada rentang 0–100.

Proyek C: Halaman Galeri Foto Kegiatan Sekolah

Bangun sebuah halaman galeri yang menampilkan beberapa foto kegiatan sekolah menggunakan Flexbox atau CSS Grid (dapat dieksplorasi mandiri sebagai pengembangan dari Flexbox pada Bab 9), dilengkapi fitur JavaScript sederhana untuk memperbesar tampilan foto ketika diklik (modal/lightbox sederhana).

Ketiga proyek di atas dapat pula dikembangkan lebih lanjut dengan menyambungkannya ke database menggunakan teknik yang telah dipelajari pada Bab 12, misalnya menyimpan daftar tugas atau nilai secara permanen alih-alih hilang setiap kali halaman dimuat ulang.

Lampiran: Ringkasan Perintah Deployment Ubuntu Server

Lampiran ini merangkum perintah-perintah utama yang digunakan pada Bab 14 untuk menyiapkan dan menerbitkan aplikasi web di Ubuntu Server, sebagai referensi cepat saat praktik.

Kategori	Perintah	Fungsi Singkat
Instalasi	<code>sudo apt install apache2 -y</code>	Menginstal web server Apache
Instalasi	<code>sudo apt install mysql-server -y</code>	Menginstal database server MySQL
Instalasi	<code>sudo apt install php libapache2-mod-php php-mysql -y</code>	Menginstal PHP beserta modul Apache dan MySQL
Layanan	<code>sudo systemctl status apache2</code>	Memeriksa status layanan Apache
Layanan	<code>sudo systemctl restart apache2</code>	Memulai ulang layanan Apache setelah perubahan konfigurasi
File	<code>sudo chown -R www-data:www-data /var/www/html/</code>	Mengubah kepemilikan file agar dapat dibaca Apache
File	<code>sudo chmod -R 755 /var/www/html/</code>	Mengatur hak akses file dan direktori web
Transfer	<code>scp -r ./folder user@ip:/tmp/</code>	Memindahkan folder proyek dari lokal ke server
Virtual Host	<code>sudo a2ensite nama.conf</code>	Mengaktifkan konfigurasi virtual host
Virtual Host	<code>sudo systemctl reload apache2</code>	Memuat ulang konfigurasi Apache tanpa memutus koneksi aktif
Keamanan	<code>sudo ufw allow 'Apache Full'</code>	Mengizinkan lalu lintas HTTP dan HTTPS pada firewall
Keamanan	<code>sudo certbot --apache</code>	Memasang sertifikat HTTPS gratis dari Let's Encrypt

Lampiran: Peta Jalan Belajar Mandiri Lanjutan

Bagi peserta didik yang ingin terus mengasah kemampuan pemrograman web setelah menuntaskan buku ini, lampiran berikut memberikan gambaran peta jalan (roadmap) belajar mandiri yang dapat diikuti secara bertahap, disusun dari yang paling relevan untuk segera dipelajari hingga topik yang lebih lanjut.

Tahap	Topik	Alasan Relevansi
1	CSS Framework (Bootstrap/Tailwind CSS)	Mempercepat proses styling tanpa menulis CSS dari nol, banyak digunakan di industri
2	JavaScript Lanjutan (Fetch API, JSON)	Memungkinkan halaman web mengambil data dari server tanpa reload penuh (AJAX)
3	Version Control dengan Git dan GitHub	Standar industri untuk mengelola dan mendokumentasikan riwayat perubahan kode
4	Framework Backend (Laravel/Node.js/Express)	Mempermudah pengembangan aplikasi web berskala lebih besar dan kompleks
5	Responsive Design dan Mobile-First	Memastikan website tampil optimal di berbagai ukuran layar, termasuk ponsel
6	Deployment ke Cloud Hosting	Memperluas kompetensi Bab 14 ke platform seperti VPS cloud atau layanan hosting modern

Peserta didik disarankan tidak terburu-buru mempelajari seluruh topik lanjutan sekaligus, melainkan memilih satu topik yang paling menarik minat dan mendalaminya melalui praktik nyata — misalnya dengan mengembangkan lebih lanjut proyek landing page dan sistem login yang telah dibangun sepanjang buku ini, sebuah pendekatan belajar berkelanjutan yang jauh lebih efektif dibanding mempelajari banyak hal secara dangkal sekaligus.

Lampiran: Checklist Proyek Akhir

Gunakan checklist berikut sebagai panduan memastikan proyek akhir (Bab 15) telah memenuhi seluruh aspek yang diharapkan sebelum dipresentasikan.

Aspek	Kriteria	Sudah?
Struktur HTML	Menggunakan heading berjenjang, semantik (nav, section) dan atribut alt pada gambar	☐
Styling CSS	Tampilan rapi, konsisten, menggunakan Flexbox pada minimal satu bagian	☐
Interaktivitas JS	Form kontak berfungsi dengan validasi dan event handling yang tepat	☐
Form Login	Password di-hash, menggunakan prepared statement, tersambung ke database	☐
Deployment	Proyek dapat diakses melalui alamat IP/domain server Ubuntu	☐
Keamanan Server	Firewall aktif, hanya port esensial yang terbuka	☐
Dokumentasi	Bagian yang dibantu AI (vibecoding) dicantumkan beserta penjelasan pemahaman pribadi	☐
Presentasi	Mampu menjelaskan alur kerja HTML-CSS-JS-PHP-MySQL secara runtut	☐

Daftar Pustaka

Irawan, Bei Harira & Prihadi, Deddy. (2022). Logika dan Algoritma Pemrograman. JABE Kreasi Indonesia. ISBN 978-623-99921-4-9.

HTML Untuk Pemula — referensi komprehensif elemen dan atribut HTML.

Mozilla Developer Network (MDN Web Docs). Dokumentasi HTML, CSS, dan JavaScript.

W3Schools. Tutorial dan referensi HTML, CSS, JavaScript, PHP, dan MySQL.

PHP Documentation Group. Manual Resmi PHP — mysqli, password_hash, dan prepared statement.

Ubuntu Documentation / Canonical. Official Ubuntu Server Documentation — Apache2, MySQL, dan UFW.

Let's Encrypt / Certbot Documentation. Panduan Instalasi Sertifikat HTTPS Gratis.

Kemdikbudristek. Capaian Pembelajaran Mata Pelajaran Pemrograman Web Dasar, Kurikulum Merdeka SMK Program Keahlian TJKT.