

Syntax Champions

Empowering Code

With AI



Ts. Izwan Suhadak Ishak | Ts. Roziyani Setik

Prakata

***“Ilmu itu bagaikan pokok, amal itu bagaikan buahnya.
Maka tiada guna ilmu tanpa amal.”***

Ibn Sina (Avicenna)

Buku *Syntax Champions: Empowering Code With AI* ini ditulis dengan tujuan untuk memberikan pemahaman yang mendalam tentang pengaturcaraan sambil memperkenalkan peranan kecerdasan buatan (AI) dalam memperkukuhkan kemahiran pengaturcaraan. Di dalam dunia teknologi yang semakin berkembang pesat, penguasaan bahasa pengaturcaraan sahaja tidak mencukupi. Penggunaan AI dalam proses pembangunan perisian kini menjadi elemen penting yang dapat membantu pelajar dan profesional mempercepatkan pemahaman mereka dalam membina aplikasi web, sistem interaktif, dan banyak lagi.

Kami percaya bahawa setiap pelajar mampu menjadi *champion* dalam dunia pengaturcaraan, dengan bantuan alat-alat canggih seperti AI. Buku ini disusun dengan pendekatan yang mudah difahami dan interaktif, di mana pelajar bukan sahaja belajar menulis kod, tetapi juga memahami bagaimana AI boleh digunakan untuk mempercepatkan proses pengaturcaraan serta memudahkan tugas-tugas yang kompleks. Bab-bab dalam buku ini diatur dengan teliti untuk membimbing pelajar daripada konsep asas seperti HTML, CSS, dan PHP, sehinggalah kepada penggunaan alat AI untuk menjana kod yang lebih efisien dan inovatif. Dengan setiap bab yang praktikal, kami berharap pelajar dapat memperkaya kemahiran teknikal mereka serta mengaplikasikan ilmu ini ke dalam projek-projek sebenar.

Sebagai tenaga pengajar dalam bidang teknologi, kami berbesar hati dapat berkongsi pengalaman kami dalam buku ini. Kami berharap bahawa buku ini bukan sahaja menjadi rujukan pembelajaran tetapi juga sumber inspirasi kepada semua pembaca untuk menjadi lebih kreatif, kritis, dan inovatif dalam dunia pengaturcaraan.

Selamat belajar dan semoga berjaya menjadi juara dalam dunia kod!

Ts. Izwan Suhadak Ishak & Ts Roziyani Setik

Isu Kandungan

Prakata	i
Bab 1: Pembelajaran Pengaturcaraan melalui AI	3
1.1 Kecerdasan Buatan (AI) memudahkan Pengaturcaraan	3
1.2 Kepentingan Menggunakan AI dalam Pembelajaran Coding	4
1.3 Alatan AI untuk Membantu Menjana Kod PHP, HTML, dan CSS	5
1.4 Membina Proses Pembelajaran Efektif dengan AI	5
1.5 Contoh Penggunaan AI dalam Pengaturcaraan	6
Bab 2: Pembangunan Aplikasi Web	7
2.1 Apa itu Aplikasi Web?	7
2.2 Struktur dan Fungsi Utama Aplikasi Web	8
2.3 Alatan Pembangunan: HTML, CSS, PHP	8
2.4 Pengenalan kepada XAMPP dan PhpMyAdmin	9
2.5 Peranan AI dalam Membantu Pembangunan Web	11
Bab 3: Pembangunan Frontend dengan HTML dan CSS	12
3.1 Struktur Asas HTML: Elemen, Tag, dan Atribut	12
3.2 Membina Halaman Web dengan HTML	15
3.3 Penggayaan Halaman Web Menggunakan CSS	16
3.4 Membina Layout Responsif dengan CSS	17
3.5 Automasi Penggayaan dengan Bantuan AI	18
3.6 Membina Laman Login dan Logout	19
3.7 Membina Laman Web dan Menu	23
3.8 Membina Laman Borang Pendaftaran	27
3.9 Paparan Senarai Maklumat Pendaftaran	30

Bab 4: Pembangunan Backend dengan PHP	34
4.1 Asas PHP: Syntax, Pembolehubah, dan Fungsi	34
4.2 Menghubungkan PHP dengan HTML	35
4.3 Membina Fungsi Dinamik: Borang dan Interaksi Pengguna	39
4.4 Menangani Data dan Proses Logik di Backend	48
4.5 Laman Login dan Logout dengan PHP	42
4.6 Laman Borang Pendaftaran dengan PHP	48
4.7 Paparan Senarai Maklumat Pendaftaran dengan PHP	52
Bab 5: Pengurusan Pangkalan Data dengan PhpMyAdmin	59
5.1 Pengenalan kepada Pangkalan Data MySQL	59
5.2 Mencipta dan Mengurus Jadual dalam PhpMyAdmin	60
5.3 Menghubungkan Pangkalan Data dengan PHP	61
5.4 CRUD (Create, Read, Update, Delete) Operasi Asas	62
5.5 Penggunaan AI untuk Automasi Operasi Pangkalan Data	64
Bab 6: Gabungan Komponen dan Pembangunan Projek	66
6.1 Gabungan Frontend dan Backend	66
6.2 Pembangunan Sistem Login Menggunakan PHP dan MySQL	68
6.3 Menambah Fungsi AI dalam Projek Aplikasi Web	70
6.4 Ujian dan Penambahbaikan Projek	72
6.5 Memasang Aplikasi Web ke Pelayan Tempatan dengan XAMPP	74

Bab 1: Pembelajaran Pengaturcaraan melalui AI

"Kewajipan manusia adalah untuk mencari kebenaran dalam segala hal, dan kebenaran tidak boleh didapati tanpa menggunakan akal dan ilmu pengetahuan."

AI-Kindi (Alkindus)

1.1 Kecerdasan Buatan (AI) memudahkan Pengaturcaraan

Kecerdasan Buatan (AI) merujuk kepada teknologi yang meniru fungsi otak manusia untuk menjalankan tugas-tugas seperti pengenalan corak, pemrosesan bahasa semula jadi, dan pembelajaran dari data. Dalam konteks pengaturcaraan, AI mampu mempelajari pola dan struktur dari kod yang ditulis oleh manusia dan menggunakannya untuk mencadangkan penyelesaian yang lebih pantas dan cekap. Teknologi AI seperti ini menggunakan teknik pembelajaran mesin (machine learning) dan rangkaian neural (neural networks) untuk meningkatkan keupayaannya dalam mengenali pelbagai aspek dalam pengaturcaraan.

Dalam pembangunan perisian, AI berfungsi sebagai pembantu automatik yang mampu membantu pengaturcara menghasilkan kod dengan lebih cepat. Sebagai contoh, GitHub Copilot, salah satu alat AI yang popular, boleh menulis kod, menjangka penyelesaian, dan memberikan cadangan kepada pengaturcara berdasarkan konteks masalah yang dihadapi. AI juga berfungsi dengan baik dalam menangani kesilapan kecil, seperti sintaksis yang salah, dan ia boleh membetulkannya tanpa campur tangan pengguna.

Selain daripada membantu menulis kod, AI juga boleh digunakan untuk menguji dan menilai kod yang ditulis. AI boleh mengesan kesilapan atau kelemahan dalam kod dan memberikan cadangan tentang cara untuk memperbaikinya. Dengan ini, pengaturcara boleh meminimumkan masa yang dihabiskan untuk debugging dan lebih fokus kepada aspek kreatif dalam pembangunan perisian. Penggunaan AI dalam pengaturcaraan memberikan kelebihan dari segi

kecekapan dan kelajuan, membolehkan pelajar dan profesional untuk meningkatkan kemahiran mereka dengan lebih pantas.

1.2 Kepentingan Menggunakan AI dalam Pembelajaran Coding

Menggunakan AI dalam pembelajaran *coding* sangat penting kerana ia mempercepatkan proses pembelajaran dan memudahkan pemahaman pelajar. AI boleh membantu dengan memberi cadangan segera berdasarkan input pengguna, yang mempercepatkan proses belajar dengan cara yang lebih interaktif. Sebagai contoh, pelajar boleh menulis beberapa baris kod PHP, dan AI seperti GitHub Copilot boleh memberi cadangan tentang baris kod seterusnya, sekaligus memandu pelajar ke arah penyelesaian yang betul tanpa perlu banyak kajian manual.

Di samping itu, AI membolehkan pelajar belajar dengan lebih berdikari. Biasanya, pelajar mungkin memerlukan bantuan guru atau sumber luar seperti tutorial dan dokumentasi untuk mempelajari sesuatu konsep. Dengan AI, mereka boleh mendapatkan maklum balas serta-merta dan mencuba pelbagai cara untuk menyelesaikan masalah secara kreatif. Ini mempercepatkan pemahaman mereka terhadap logik pengaturcaraan dan bagaimana pelbagai elemen kod berinteraksi.

Selain daripada itu, AI juga membantu meningkatkan motivasi dan keyakinan pelajar. Pelajar yang baru mempelajari pengaturcaraan mungkin mudah merasa tertekan apabila menghadapi kesilapan atau kekurangan pemahaman terhadap konsep-konsep yang kompleks. AI boleh menyediakan sokongan berterusan dan memastikan pelajar tidak berasa terasing apabila menghadapi masalah teknikal. Dengan adanya bantuan ini, pelajar lebih yakin dalam menyelesaikan masalah dan mencapai objektif pengaturcaraan mereka dengan lebih cepat.

1.3 Alatan AI untuk Membantu Menjana Kod PHP, HTML, dan CSS

Terdapat pelbagai alatan AI yang tersedia untuk membantu dalam penjanaan kod PHP, HTML, dan CSS. Salah satu yang paling popular ialah **GitHub Copilot**, yang dibangunkan oleh GitHub dan OpenAI. Alat ini berfungsi seperti pembantu automatik yang mencadangkan kod berdasarkan konteks yang ditulis oleh pengguna. Ia boleh mengenal pasti struktur yang diperlukan dalam HTML, CSS, atau PHP dan memberikan cadangan kod yang sesuai untuk melengkapkan fungsi tertentu dalam pembangunan aplikasi web.

Selain GitHub Copilot, **OpenAI Codex** juga merupakan alat yang sangat berkuasa. Codex mampu menjana kod berdasarkan input bahasa semula jadi. Contohnya, jika pengguna meminta Codex untuk menjana halaman HTML dengan CSS yang responsif, ia boleh melakukannya hanya

dengan arahan ringkas dalam bahasa manusia. Ini membolehkan pelajar atau pengaturcara yang baru belajar menumpukan lebih kepada aspek logik dan kreativiti, sambil AI menangani penulisan kod teknikal yang kompleks.

Kite adalah satu lagi alat yang memberikan cadangan kod dalam masa nyata ketika pengguna menaip di editor mereka. Alat ini boleh digunakan untuk pelbagai bahasa pengaturcaraan, termasuk PHP, HTML, dan CSS. Keupayaan Kite untuk memberikan cadangan yang relevan dan tepat waktu membantu pengaturcara menulis kod dengan lebih efisien. Alatan AI ini bukan sahaja mempermudah penulisan kod, tetapi juga mengurangkan masa yang diperlukan untuk pembelajaran dan pembangunan projek.

1.4 Membina Proses Pembelajaran Efektif dengan AI

Untuk membina proses pembelajaran yang berkesan menggunakan AI, pendekatan berstruktur perlu diterapkan. Pertama, pelajar perlu diperkenalkan kepada alatan AI yang relevan seperti GitHub Copilot atau OpenAI Codex dan diajar bagaimana untuk menggunakannya secara strategik dalam pembangunan aplikasi. Ini termasuk memahami bagaimana alat ini berfungsi, bagaimana ia memberikan cadangan, dan bagaimana pelajar boleh menggunakan cadangan ini untuk memperbaiki kod mereka.

Kedua, pelajar perlu diberi peluang untuk bereksperimen dengan AI dalam pelbagai senario pengaturcaraan. Contohnya, pelajar boleh diberi tugas untuk menulis halaman web dengan HTML dan CSS, di mana mereka perlu meminta bantuan AI untuk mempercepatkan proses penulisan kod atau memperbaiki kesilapan. Proses ini membolehkan pelajar melihat sendiri bagaimana AI boleh membantu dalam menjana kod yang betul dan mengatasi cabaran pengaturcaraan secara cekap.

Akhir sekali, AI juga boleh digunakan untuk mengesan kemajuan pelajar dan menyesuaikan pembelajaran mereka. Dengan adanya AI, para pengajar boleh menggunakan sistem yang memantau bagaimana pelajar berinteraksi dengan kod dan alatan AI, seterusnya menyediakan maklum balas yang bersesuaian dengan keperluan pembelajaran individu. Ini membantu memastikan bahawa setiap pelajar menerima perhatian yang diperlukan untuk menguasai kemahiran pengaturcaraan dengan baik.

1.5 Contoh Penggunaan AI dalam Pengaturcaraan

AI telah terbukti sangat berguna dalam menyelesaikan pelbagai masalah pengaturcaraan yang dihadapi oleh pelajar dan profesional. Sebagai contoh, apabila pelajar menulis kod PHP untuk menghubungkan ke pangkalan data MySQL, mereka mungkin menghadapi masalah sintaks atau kesilapan logik. GitHub Copilot boleh mengenali kesilapan ini dan mencadangkan kod yang betul untuk menyelesaikan masalah tersebut. Ini menjimatkan masa yang biasanya digunakan untuk menyemak dokumentasi atau meminta bantuan daripada guru.

AI juga membantu dalam debugging, satu proses yang sering mengambil masa yang lama bagi pelajar. Dengan menggunakan alat seperti Codex, pelajar boleh meminta AI untuk mencari punca kesilapan dalam kod mereka dan memberikan cadangan bagaimana untuk membaikinya. Sebagai contoh, jika ada masalah dengan borang HTML yang tidak menghantar data dengan betul ke skrip PHP, AI boleh mengenal pasti masalahnya dan memberikan penyelesaian yang sesuai.

Tambahan lagi, AI boleh digunakan untuk mengoptimumkan prestasi kod. Dalam banyak kes, kod yang ditulis oleh pelajar mungkin berfungsi tetapi tidak efisien. AI boleh mengesan cara yang lebih baik untuk menulis kod yang sama, mengurangkan redundansi dan meningkatkan kecekapan aplikasi. Ini bukan sahaja menyelesaikan masalah yang sedia ada, tetapi juga membantu pelajar belajar tentang amalan terbaik dalam pengaturcaraan.

Bab 2: Pembangunan Aplikasi Web

“Tujuan tertinggi manusia adalah mencapai kebahagiaan melalui ilmu pengetahuan dan kebijaksanaan.”

Al-Farabi

2.1 Apa itu Aplikasi Web?

Aplikasi web ialah perisian yang dijalankan melalui pelayar web, seperti Google Chrome, Firefox, atau Safari, yang membolehkan pengguna berinteraksi dengan data melalui antara muka pengguna (UI). Tidak seperti aplikasi desktop yang memerlukan pemasangan pada komputer tempatan, aplikasi web boleh diakses melalui internet dari mana-mana peranti yang disambungkan. Contoh-contoh aplikasi web termasuk laman media sosial seperti Facebook, alat produktiviti seperti Google Docs, dan platform e-dagang seperti Lazada dan Shopee.

Aplikasi web biasanya mempunyai tiga komponen utama: frontend, backend, dan pangkalan data. Frontend ialah bahagian aplikasi yang dilihat dan digunakan oleh pengguna, yang sering kali dibangunkan menggunakan bahasa seperti HTML, CSS, dan JavaScript. Backend pula ialah bahagian aplikasi yang menguruskan logik, pemprosesan data, dan komunikasi dengan pangkalan data. Pangkalan data menyimpan semua maklumat yang diperlukan oleh aplikasi, seperti data pengguna, produk, dan transaksi.

Salah satu kelebihan utama aplikasi web ialah kebolehaksesannya yang tinggi. Oleh kerana ia berjalan di pelayar, aplikasi web tidak terikat kepada platform tertentu, membolehkan pengguna dari pelbagai sistem operasi seperti Windows, macOS, atau Linux untuk mengaksesnya tanpa sebarang masalah keserasian. Aplikasi web juga lebih mudah dikemas kini kerana perubahan pada pelayan terus berkuat kuasa kepada semua pengguna tanpa perlu mengemaskini aplikasi pada peranti mereka.

2.2 Struktur dan Fungsi Utama Aplikasi Web

Struktur asas aplikasi web boleh dibahagikan kepada tiga lapisan utama: **lapisan pembentangan (presentation layer)**, **lapisan logik perniagaan (business logic layer)**, dan **lapisan pangkalan data (database layer)**. **Lapisan pembentangan** merangkumi antaramuka pengguna yang dilihat oleh pengguna. Ini adalah bahagian di mana elemen visual seperti butang, menu, dan borang ditempatkan, dan ia dibina dengan HTML untuk struktur, CSS untuk reka bentuk, dan JavaScript untuk interaktiviti.

Lapisan logik perniagaan ialah bahagian yang menguruskan bagaimana data diproses dan dihantar di antara frontend dan backend. Kod PHP sering digunakan dalam lapisan ini untuk mengurus permintaan pengguna, memproses borang, atau menjalankan logik yang lebih kompleks. Sebagai contoh, apabila pengguna log masuk ke dalam aplikasi, lapisan logik ini akan memproses butiran log masuk dan mengesahkan kredensial pengguna melalui pangkalan data.

Lapisan pangkalan data ialah bahagian yang menyimpan dan mengurus data yang digunakan oleh aplikasi. Pangkalan data boleh menjadi relasi (seperti MySQL atau PostgreSQL) atau NoSQL (seperti MongoDB), bergantung pada keperluan aplikasi. Dalam struktur ini, setiap lapisan mempunyai peranan tersendiri tetapi berinteraksi antara satu sama lain untuk menyampaikan pengalaman pengguna yang lancar. Fungsi utama aplikasi web termasuk pengurusan data pengguna, autentikasi pengguna, pemrosesan transaksi, dan penyampaian kandungan yang dinamik.

2.3 Alatan Pembangunan: HTML, CSS, PHP

Dalam pembangunan aplikasi web, HTML, CSS, dan PHP adalah alatan utama yang digunakan untuk membina frontend dan backend aplikasi. **HTML (HyperText Markup Language)** digunakan untuk mencipta rangka dan struktur asas laman web. Ia adalah bahasa yang menyusun elemen-elemen seperti tajuk, perenggan, gambar, dan pautan pada halaman web. Tanpa HTML, tidak akan ada kerangka untuk aplikasi web untuk dipersembahkan kepada pengguna.

CSS (Cascading Style Sheets) berfungsi sebagai alat penggayaan yang menjadikan laman web kelihatan menarik. Ia digunakan untuk menetapkan warna, jenis tulisan, margin, padding, dan reka bentuk keseluruhan laman web. Dengan CSS, pembangun dapat memastikan laman web kelihatan kemas dan seragam merentas pelbagai pelayar web dan peranti. CSS juga membantu

membuat laman web responsif, membolehkan ia menyesuaikan dengan saiz skrin peranti mudah alih.

PHP (Hypertext Preprocessor) adalah bahasa pengaturcaraan pelayan yang digunakan untuk membina logik backend aplikasi. PHP membolehkan aplikasi web berinteraksi dengan pangkalan data, mengurus sesi pengguna, dan memproses input yang dinamik. PHP sering digabungkan dengan pangkalan data seperti MySQL untuk menyimpan dan mendapatkan maklumat yang diperlukan untuk aplikasi. Sebagai contoh, apabila pengguna mengisi borang pada aplikasi web, PHP digunakan untuk memproses data borang dan menyimpannya dalam pangkalan data.

Dalam proses pembangunan aplikasi web menggunakan HTML, CSS, dan PHP, penggunaan persekitaran pembangunan bersepadu (IDE) atau editor kod yang sesuai adalah sangat penting untuk meningkatkan kecekapan dan produktiviti pembangun. **Visual Studio Code (VS Code)** adalah salah satu IDE yang paling popular untuk pembangunan web. VS Code menawarkan pelbagai ciri yang membantu pembangun, termasuk autolengkap kod, integrasi Git, dan pelbagai sambungan (extensions) seperti **PHP Intelephense** untuk pengaturcaraan PHP dan **Live Server** untuk melihat hasil HTML dan CSS secara langsung. Ciri-ciri ini membolehkan pembangun melihat perubahan yang dilakukan pada kod mereka secara masa nyata, menjadikan debugging dan pengujian lebih mudah dan efisien.

Selain VS Code, **Sublime Text** juga merupakan pilihan yang popular, terutamanya bagi mereka yang mencari editor yang ringan tetapi berkuasa. Sublime Text menyokong pelbagai bahasa pengaturcaraan, termasuk HTML, CSS, dan PHP, serta menyediakan fungsi seperti multi-line editing, autolengkap kod, dan tema yang boleh disesuaikan mengikut keperluan pengguna. Satu kelebihan utama Sublime Text adalah kelajuan dan keupayaannya untuk mengendalikan fail besar dengan cepat, menjadikannya pilihan yang sesuai bagi projek berskala besar. Walaupun ia tidak sekompleks VS Code, kelebihan dari segi kelajuan dan kebolegunaan menjadikannya pilihan yang popular di kalangan pembangun.

2.4 Pengenalan kepada XAMPP dan PhpMyAdmin

XAMPP ialah alat yang menyediakan persekitaran pembangunan tempatan untuk pembangun web. Ia mengandungi pelayan Apache, pangkalan data MySQL (sekarang MariaDB), dan beberapa alat lain seperti PHP dan Perl. XAMPP membolehkan pembangun untuk menjalankan pelayan web dan menguji aplikasi web secara tempatan pada komputer mereka tanpa memerlukan akses ke pelayan jarak jauh. Ini sangat membantu dalam pembangunan aplikasi

web kerana ia menyediakan persekitaran yang selamat dan boleh diubah suai untuk pembangunan dan ujian.

Dalam XAMPP, **PhpMyAdmin** adalah antara muka pengguna grafik (GUI) yang memudahkan pengurusan pangkalan data MySQL atau MariaDB. PhpMyAdmin membolehkan pembangun membuat dan mengurus jadual pangkalan data, menyimpan data, mengubah data, dan menjalankan query SQL dengan mudah. Alat ini sangat berguna bagi mereka yang baru belajar pangkalan data kerana ia mempunyai antara muka yang mesra pengguna untuk mengurus pangkalan data tanpa perlu menggunakan baris perintah.

XAMPP dan PhpMyAdmin adalah alat yang sangat penting dalam pembangunan web kerana ia menyediakan kemudahan kepada pembangun untuk menjalankan aplikasi web dan mengurus pangkalan data secara setempat. Alat-alat ini juga membolehkan pembangun menguji aplikasi mereka dalam persekitaran yang realistik sebelum dipindahkan ke pelayan pengeluaran.

XAMPP bukan sahaja berguna untuk menjalankan pelayan web dan pangkalan data di persekitaran tempatan, tetapi ia juga sangat berguna untuk pembangunan **offline**. Ini bermaksud pembangun tidak perlu bergantung kepada sambungan internet atau pelayan jarak jauh semasa fasa pembangunan. XAMPP menyediakan semua komponen penting seperti pelayan **Apache**, **MySQL** (MariaDB), dan **PHP** yang membolehkan pembangun menjalankan aplikasi web sepenuhnya pada komputer tempatan mereka. Ini membolehkan pembangun menguji aplikasi, mengesan dan membetulkan kesilapan, serta membuat perubahan tanpa perlu risau tentang isu latensi atau kelajuan internet. Selain itu, ia menawarkan persekitaran yang selamat dan boleh dikawal sepenuhnya oleh pembangun sebelum aplikasi dipindahkan ke pelayan sebenar.

Namun, untuk menjadikan aplikasi yang dibangunkan menggunakan XAMPP dapat diakses oleh pengguna **online**, pembangun perlu menggunakan **web hosting**. Web hosting adalah perkhidmatan yang menyediakan pelayan berasaskan internet untuk menyimpan dan menjalankan aplikasi web anda secara **dalam talian**. Contoh hosting yang popular termasuk **Hostinger**, **Bluehost**, dan **GoDaddy**. Setelah aplikasi siap di XAMPP, ia boleh dipindahkan ke persekitaran hosting ini melalui **FTP (File Transfer Protocol)**, membolehkan pengguna mengakses aplikasi melalui internet. Di hosting, pangkalan data juga boleh disetup menggunakan **PhpMyAdmin** yang biasanya disediakan oleh penyedia hosting, dan aplikasi yang sebelum ini dibangunkan secara lokal kini boleh berjalan secara penuh dalam talian.



2.5 Peranan AI dalam Membantu Pembangunan Web

AI telah membawa revolusi dalam pembangunan web dengan membantu pembangun menghasilkan kod dengan lebih cepat dan cekap. Dalam pembangunan frontend, AI boleh digunakan untuk menghasilkan reka bentuk halaman web berdasarkan input pengguna atau menganalisis susun atur terbaik untuk pengalaman pengguna yang optimum. Sebagai contoh, alat seperti OpenAI Codex mampu menjana kod HTML dan CSS berdasarkan penerangan yang diberikan dalam bahasa semulajadi, yang mempercepatkan proses pembangunan laman web.

Di samping itu, AI membantu dalam pembangunan backend dengan menjana kod PHP untuk tugas seperti pengurusan pangkalan data, logik perniagaan, dan pengesahan pengguna. AI seperti GitHub Copilot dapat mengurangkan masa yang dihabiskan untuk menulis kod yang berulang dengan memberi cadangan yang bijak, seperti ketika mengendalikan fungsi CRUD (Create, Read, Update, Delete). Pembangun hanya perlu memberikan beberapa input asas dan AI akan mengisi selebihnya, menjimatkan masa dan usaha.

Tambahan lagi, AI membantu dalam pengujian dan debugging aplikasi web. Dengan alat-alat automasi AI, pembangun dapat mengenal pasti kesilapan dengan lebih pantas, menganalisis corak yang mungkin menyebabkan masalah prestasi, dan memberikan cadangan penyelesaian yang lebih baik. Ini membolehkan pembangun memperbaiki aplikasi mereka dengan lebih cepat dan melancarkan produk yang lebih stabil dan berkualiti tinggi.

Bab 3: Pembangunan Fronted dengan HTML dan CSS

“Kebahagiaan sejati terletak dalam ilmu yang benar, ilmu yang mendekatkan manusia kepada Tuhan.”

Al-Ghazali

3.1 Struktur Asas HTML: Elemen, Tag, dan Atribut

HTML (HyperText Markup Language) adalah asas bagi semua halaman web, dan ia terdiri daripada elemen-elemen yang digunakan untuk menentukan struktur dan kandungan laman web. **Elemen** dalam HTML biasanya didefinisikan menggunakan **tag** yang diletakkan di antara kurungan sudut. Contohnya, tag `<p>` digunakan untuk menandakan perenggan, manakala tag `<h1>` digunakan untuk tajuk utama. Setiap tag HTML mempunyai pembuka (`<p>`) dan penutup (`</p>`) kecuali elemen tertentu seperti `` yang bersifat kosong.

Selain daripada tag, elemen HTML juga mempunyai **atribut**, yang digunakan untuk memberikan maklumat tambahan kepada elemen tersebut. Contohnya, atribut `src` digunakan untuk menentukan lokasi imej dalam tag ``, manakala atribut `href` dalam tag `<a>` digunakan untuk menentukan pautan URL. Atribut diletakkan dalam tag pembuka dan mengikuti format seperti ini: ``. Struktur ini adalah penting kerana ia menentukan bagaimana kandungan dipaparkan kepada pengguna di pelayar web.

Setiap elemen HTML mempunyai fungsi tertentu yang membantu membina keseluruhan struktur halaman web. Dari elemen-elemen struktur seperti `<div>` dan `<section>` hingga elemen teks seperti `<h1>` dan `<p>`, HTML menawarkan pelbagai alat untuk mencipta kandungan yang dinamik dan teratur. Dengan pemahaman yang kuat mengenai elemen, tag, dan atribut, pelajar dapat membina laman web asas yang berfungsi dengan baik. Sila rujuk Jadual 1.0 Penggunaan Elemen, Tag, dan Atribut HTML.

Jadual 1.0 Penggunaan Elemen, Tag, dan Atribut HTML

Kategori	Elemen HTML / Tag / Atribut	Penerangan	Contoh Penggunaan
Tag	<h1>	Menentukan tajuk utama terbesar, biasanya untuk tajuk utama.	<h1>Tajuk Utama</h1>
Tag	<h2>	Menentukan tajuk sekunder, biasanya untuk tajuk bahagian.	<h2>Tajuk Bahagian</h2>
Tag	<p>	Menentukan perenggan teks.	<p>Ini adalah perenggan.</p>
Tag	<a>	Menentukan pautan hiperteks, digunakan untuk memaut ke halaman lain atau URL.	Pautan
Tag		Menyisipkan imej ke dalam dokumen.	
Tag		Menentukan bekas kecil untuk elemen sebaris seperti teks atau ikon.	Teks Sebaris
Tag		Menentukan senarai tidak teratur (senarai dengan peluru).	 ...
Tag	<th>	Menentukan sel tajuk dalam jadual.	<th>Sel Tajuk</th>
Tag	<td>	Menentukan sel data dalam jadual.	<td>Sel Data</td>

Elemen	<html>	Elemen utama dalam dokumen HTML.	<html> ... </html>
Elemen	<head>	Mengandungi metadata dan pautan ke stylesheet atau skrip.	<head> ... </head>
Elemen	<title>	Menentukan tajuk dokumen, dipaparkan di bar tajuk atau tab pelayar.	<title>Judul Dokumen</title>
Elemen	<body>	Mengandungi kandungan laman web (teks, gambar, dll.).	<body> ... </body>
Elemen	<div>	Menentukan bahagian atau seksyen dalam dokumen, sering digunakan untuk tujuan susun atur.	<div> ... </div>
Elemen		Menentukan item dalam senarai teratur atau tidak teratur.	Item 1
Elemen		Menentukan senarai teratur (senarai bernombor).	 ...
Elemen	<table>	Menentukan jadual dalam HTML.	<table> ... </table>
Elemen	<tr>	Menentukan baris dalam jadual.	<tr> ... </tr>
Elemen	<form>	Menentukan borang HTML untuk input pengguna.	<form> ... </form>
Atribut	<input type='text'>	Menentukan medan input dalam borang, seperti teks, kata laluan, atau butang.	<input type='text'>

3.2 Membina Halaman Web dengan HTML

Setelah memahami struktur asas HTML, langkah seterusnya adalah membina halaman web yang lengkap. Setiap halaman web HTML bermula dengan **deklarasi dokumen**, iaitu `<!DOCTYPE html>` yang memberitahu pelayar bahawa dokumen ini menggunakan HTML5. Selepas itu, elemen utama dalam dokumen HTML adalah elemen `<html>`, di mana seluruh kandungan halaman ditempatkan. Elemen `<head>` digunakan untuk menyimpan maklumat meta seperti tajuk dokumen, pautan ke fail CSS, atau kod JavaScript, manakala elemen `<body>` mengandungi kandungan sebenar yang akan dipaparkan kepada pengguna.

Contoh struktur asas HTML untuk membina laman web adalah seperti berikut:

```
<!DOCTYPE html>

<html>

<head>

  <title>Halaman Utama</title>

</head>

<body>

  <h1>Selamat Datang ke Laman Web Saya</h1>

  <p>Ini adalah contoh perenggan yang diletakkan dalam laman
web.</p>

  <a href="https://www.contoh.com">Klik di sini untuk maklumat
lanjut</a>

</body>

</html>
```

Elemen `<h1>` digunakan untuk tajuk besar, manakala elemen `<p>` digunakan untuk perenggan teks. Elemen `<a>` pula digunakan untuk pautan ke laman web lain. Dalam contoh di atas, pelajar telah membina halaman web yang ringkas dengan tajuk, teks, dan pautan.

Selepas membina struktur asas, pembangun boleh menambahkan elemen-elemen lain seperti imej dengan menggunakan tag ``, senarai dengan `` atau ``, dan jadual dengan `<table>`. Kesemua elemen ini membantu mengatur kandungan halaman web agar lebih tersusun dan mudah difahami oleh pengguna.

3.3 Penggayaan Halaman Web Menggunakan CSS

CSS (Cascading Style Sheets) digunakan untuk menentukan rupa dan gaya halaman web. Dengan CSS, pembangun dapat mengawal warna, saiz fon, margin, padding, dan pelbagai elemen visual lain untuk menjadikan laman web kelihatan lebih menarik dan profesional. CSS boleh digunakan dalam tiga cara utama: **inline**, **internal**, dan **external**. Penggunaan **CSS external** biasanya lebih disukai kerana ia memisahkan kod HTML daripada kod penggayaan, menjadikan halaman web lebih teratur dan mudah diselenggara.

Sebagai contoh, fail **CSS external** boleh ditautkan dalam elemen `<head>` seperti berikut:

Fail .html

```
<head>
  <link rel="stylesheet" href="style.css">
</head>
```

Di dalam fail **style.css**, kod penggayaan untuk elemen HTML boleh ditulis seperti berikut:

Fail .css

```
body {
  background-color: lightblue;
  font-family: Arial, sans-serif;
}

h1 {
```

```
color: navy;
text-align: center;
}

p {
font-size: 16px;
line-height: 1.5;
}
```

Dalam contoh ini, latar belakang halaman web diwarnakan biru muda, teks pada elemen `<h1>` diberi warna biru gelap, dan teks dalam perenggan diberi saiz fon 16px dengan jarak baris 1.5. Dengan CSS, pembangun dapat mengawal elemen-elemen visual secara menyeluruh dan memberikan identiti visual kepada laman web mereka.

3.4 Membina Layout Responsif dengan CSS

Dengan pelbagai jenis peranti yang digunakan untuk melayari laman web, dari telefon pintar hingga komputer meja, adalah penting untuk membina **layout responsif** yang boleh menyesuaikan diri dengan saiz skrin yang berbeza. CSS menawarkan alat yang kuat seperti **media queries** untuk memastikan laman web dapat berfungsi dengan baik pada skrin kecil dan besar. Media queries membolehkan pembangun menentukan aturan gaya yang berbeza berdasarkan lebar skrin peranti.

Sebagai contoh, menggunakan media query untuk menyesuaikan gaya laman web pada skrin telefon pintar:

Fail .css

```
@media (max-width: 600px) {
body {
```

```
background-color: lightgrey;
}

h1 {
  font-size: 24px;
}
}
```

Dalam contoh ini, jika lebar skrin peranti kurang daripada 600px, latar belakang laman web akan bertukar menjadi kelabu muda, dan saiz fon untuk tajuk `<h1>` akan dikecilkan menjadi 24px untuk kelihatan lebih sesuai pada skrin kecil. Ini memastikan laman web tidak hanya kelihatan baik pada komputer meja tetapi juga pada peranti mudah alih.

Selain media queries, **CSS Grid** dan **Flexbox** adalah teknik yang popular untuk membina layout yang fleksibel. **Flexbox** sangat berguna untuk mencipta layout satu dimensi, seperti baris atau lajur, manakala **CSS Grid** digunakan untuk mencipta layout dua dimensi yang lebih kompleks. Kedua-dua teknik ini membantu dalam mencipta laman web yang responsif, moden, dan mesra pengguna.

3.5 Automasi Penggayaan dengan Bantuan AI

Dengan perkembangan teknologi, AI kini memainkan peranan yang semakin penting dalam membantu pembangun membina dan menggayakan laman web dengan lebih cekap. Alat seperti **GitHub Copilot** dan **OpenAI Codex** mampu menjana kod CSS dan HTML secara automatik berdasarkan input yang diberikan. Sebagai contoh, jika seorang pembangun ingin mencipta butang dengan reka bentuk yang menarik, AI boleh menjana kod yang sesuai hanya dengan memberikan arahan dalam bahasa semula jadi.

Sebagai contoh, pembangun boleh meminta AI untuk mencipta butang berwarna biru dengan sudut bulat dan bayang pada teks. AI akan menjana kod CSS seperti berikut:

Fail .css

```
button {  
    background-color: blue;  
    color: white;  
    border-radius: 10px;  
    box-shadow: 2px 2px 5px grey;  
    padding: 10px 20px;  
    font-size: 16px;  
}
```

Penggunaan AI dalam penjaanan kod bukan sahaja menjimatkan masa tetapi juga membantu pembangun untuk fokus kepada aspek kreatif dalam pembangunan laman web. Selain itu, AI juga dapat membantu dalam mengoptimumkan kod dengan memberikan cadangan untuk kod yang lebih efisien dan responsif. Dengan bantuan AI, proses pembangunan frontend menjadi lebih pantas dan produktif.

3.6 Membina Laman Login dan Logout

Laman login dan logout adalah penting untuk mana-mana laman web yang melibatkan pengguna berdaftar. Laman ini membolehkan pengguna mengakses kandungan atau fungsi yang disesuaikan untuk mereka. HTML digunakan untuk mencipta borang yang membolehkan pengguna memasukkan maklumat mereka seperti nama pengguna dan kata laluan. Elemen `<input>` dengan jenis `text` dan `password` digunakan untuk menerima input dari pengguna, manakala butang submit menghantar data tersebut untuk diproses di server.

CSS memainkan peranan penting dalam menjadikan borang login lebih menarik dan mudah digunakan. Anda boleh menambah gaya pada medan input dan butang untuk menjadikan borang kelihatan profesional. Sebagai contoh, padding, margin, dan border-radius boleh digunakan untuk memberikan ruang yang mencukupi di sekitar elemen dan menjadikan borang kelihatan lebih kemas. CSS juga boleh digunakan untuk menambah gaya hover pada butang submit,

memberikan maklum balas visual kepada pengguna apabila mereka menggerakkan kursor ke atas butang tersebut.

Selepas pengguna log masuk, sistem backend akan memproses maklumat yang dimasukkan untuk mengesahkan identiti pengguna. Setelah berjaya log masuk, laman web boleh memaparkan butang logout untuk pengguna keluar dari sesi mereka. Fungsi logout biasanya dilakukan dengan menghapuskan sesi pengguna di server, dan ini memastikan keselamatan maklumat pengguna setelah mereka selesai menggunakan laman web.

Kod aturcara berikut menunjukkan cara untuk membina borang **Login** dan **Logout** menggunakan HTML dan CSS:

HTML (Laman Login):

Fail .html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width,
initial-scale=1.0">

  <title>Login</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      background-color: #f0f0f0;

      display: flex;

      justify-content: center;

      align-items: center;

      height: 100vh;
```



```
}
```

```
form {  
    background-color: white;  
    padding: 20px;  
    border-radius: 8px;  
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);  
}
```

```
input[type="text"], input[type="password"] {  
    width: 100%;  
    padding: 10px;  
    margin: 10px 0;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
}
```

```
input[type="submit"] {  
    background-color: #4CAF50;  
    color: white;  
    padding: 10px;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;
```

```
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <form action="/login" 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>

        <input type="submit" value="Login">
    </form>
</body>
</html>
```


HTML (Laman Logout):

Fail .html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Logout</title>
</head>
<body>
  <h1>Anda telah log keluar.</h1>
  <a href="/login">Log Masuk Semula</a>
</body>
</html>
```

3.7 Membina Laman Web dengan Menu

Membina laman web dengan **menu navigasi** adalah satu cara yang biasa untuk memastikan pengguna boleh berpindah dari satu halaman ke halaman lain dengan mudah. Menu navigasi ini biasanya terletak di bahagian atas laman web dan membolehkan akses cepat kepada halaman utama seperti **Home**, **About**, dan **Contact**. Menu ini boleh dibina menggunakan elemen `<nav>` dan `` untuk menyusun pautan secara teratur.

CSS digunakan untuk menambah pengayaan pada menu ini agar ia kelihatan kemas dan menarik. Contohnya, anda boleh menggunakan `display: inline-block;` untuk memastikan menu dipaparkan dalam satu baris dan `text-decoration: none;` untuk menghilangkan garis

bawah pada pautan. Gaya hover juga boleh ditambahkan untuk memberikan maklum balas visual kepada pengguna, contohnya dengan menukar warna latar atau teks apabila kursor bergerak di atas item menu.

Selain itu, penggunaan **media queries** memastikan menu responsif dan mudah diakses di pelbagai peranti. Sebagai contoh, pada skrin telefon pintar, menu boleh ditukar kepada **hamburger menu** yang lebih sesuai dengan skrin kecil. Ini memastikan laman web mesra pengguna tidak kira peranti yang digunakan, memberikan pengalaman navigasi yang lancar kepada semua jenis pengguna.

Kod berikut menunjukkan bagaimana untuk membina laman web dengan **menu** menggunakan HTML dan CSS:

Fail .html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Laman dengan 3 Menu</title>
  <style>
    nav ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      background-color: #333;
      overflow: hidden;
    }
```



```
nav ul li {
    float: left;
}

nav ul li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 20px;
    text-decoration: none;
}

nav ul li a:hover {
    background-color: #111;
}
</style>
</head>
<body>

<nav>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
```



```
        </ul>
</nav>

<section id="home">
    <h1>Selamat Datang ke Halaman Utama</h1>
    <p>Ini adalah halaman utama.</p>
</section>

<section id="about">
    <h1>Tentang Kami</h1>
    <p>Ini adalah halaman tentang kami.</p>
</section>

<section id="contact">
    <h1>Hubungi Kami</h1>
    <p>Ini adalah halaman untuk menghubungi kami.</p>
</section>

</body>
</html>
```

3.8 Membina Laman Borang Pendaftaran

Borang pendaftaran adalah komponen asas dalam laman web yang membolehkan pengguna mendaftar akaun untuk mengakses fungsi-fungsi khas seperti pembelian, komen, atau penggunaan perkhidmatan tertentu. Elemen HTML seperti `<form>`, `<input>`, dan `<label>` digunakan untuk membina struktur borang ini. Medan input seperti `text` dan `password` membantu pengguna memasukkan maklumat mereka, manakala `type="submit"` digunakan untuk menghantar borang ke server untuk diproses.

Borang ini boleh disesuaikan dengan CSS untuk memastikan ia mudah digunakan dan menarik. Contohnya, anda boleh menambah `padding` dan `margin` pada setiap medan input untuk memberikan ruang yang mencukupi antara medan, menjadikan borang lebih mudah dibaca. Warna latar pada butang submit juga boleh disesuaikan untuk menarik perhatian pengguna dan meningkatkan kebolehgunaan.

Selain itu, untuk laman web yang lebih maju, **validasi form** boleh ditambah menggunakan HTML5 atau JavaScript. HTML5 menyediakan atribut seperti `required` dan `pattern` untuk memastikan pengguna memasukkan data yang sah, seperti alamat e-mel yang betul atau kata laluan yang cukup kuat. Ini meningkatkan keselamatan dan kebolehgunaan laman web serta memastikan data pengguna dihantar dengan betul kepada server.

Contoh berikut menunjukkan borang pendaftaran yang ringkas menggunakan HTML dan CSS:

Fail .html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width,
initial-scale=1.0">

  <title>Borang Pendaftaran</title>

<style>

  form {
```



```
width: 300px;
margin: 0 auto;
padding: 20px;
background-color: #f9f9f9;
border-radius: 8px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
```

```
input[type="text"], input[type="email"],
input[type="password"] {
width: 100%;
padding: 10px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
}
```

```
input[type="submit"] {
width: 100%;
background-color: #4CAF50;
color: white;
padding: 14px 20px;
```

```
        margin: 8px 0;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>

<h2>Borang Pendaftaran</h2>

<form action="/register" method="post">
    <label for="name">Nama:</label>
    <input type="text" id="name" name="name" required>

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

    <label for="password">Kata Laluan:</label>
```

```
<input type="password" id="password" name="password"
required>

<input type="submit" value="Daftar">

</form>

</body>

</html>
```

3.9 Paparan Senarai Maklumat Pendaftaran

Selepas pengguna mendaftar di laman web, maklumat mereka seperti nama, e-mel, atau tarikh pendaftaran boleh dipaparkan dalam bentuk senarai atau jadual untuk tujuan pengurusan. HTML menyediakan elemen `<table>`, `<tr>`, dan `<td>` untuk membina struktur jadual yang memaparkan data pengguna dalam format yang teratur. Jadual ini berguna dalam pelbagai situasi seperti pengurusan pengguna atau paparan senarai peserta bagi sesuatu acara.

CSS memainkan peranan dalam menambah gaya pada jadual untuk menjadikannya lebih kemas dan mudah dibaca. Sebagai contoh, anda boleh menambah **border** pada sel jadual untuk membezakan setiap baris dan lajur. **Hover effect** juga boleh ditambah untuk memberi kesan visual apabila pengguna menggerakkan cursor mereka di atas baris jadual. Penggunaan warna latar yang berbeza untuk baris genap dan ganjil juga boleh membantu meningkatkan keterbacaan jadual.

Selain itu, anda boleh menambah **sort dan filter** menggunakan JavaScript untuk membolehkan pengguna mengatur maklumat dalam jadual berdasarkan nama atau tarikh. Ini menjadikan paparan senarai maklumat lebih interaktif dan mesra pengguna. Dengan penggunaan HTML, CSS, dan JavaScript yang tepat, paparan maklumat pendaftaran akan menjadi lebih profesional dan mudah diuruskan oleh pengguna atau pentadbir laman web.

Berikut adalah contoh kod HTML untuk memaparkan senarai maklumat pendaftaran dalam bentuk jadual:

Fail .html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Senarai Pendaftaran</title>
  <style>
    table {
      width: 50%;
      margin: 20px auto;
      border-collapse: collapse;
    }

    table, th, td {
      border: 1px solid black;
    }

    th, td {
      padding: 10px;
      text-align: left;
    }
  </style>
</head>
<tbody>
  <tr>
    <td></td>
  </tr>
</tbody>
</html>
```



```
        th {
            background-color: #4CAF50;
            color: white;
        }

        tr:nth-child(even) {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>

<h2>Senarai Maklumat Pendaftaran</h2>

<table>
    <tr>
        <th>Nama</th>
        <th>Email</th>
    </tr>
    <tr>
        <td>Ali Bin Abu</td>
        <td>ali@example.com</td>
    </tr>
    <tr>
```



```
        <td>Aisyah Binti Ahmad</td>
        <td>aisyah@example.com</td>
    </tr>
</table>

</body>
</html>
```

Bab 4: Pembangunan Backend dengan PHP

"Ilmu adalah cahaya yang menerangi jalan hidup manusia, dan tiada ilmu tanpa usaha dan kesungguhan yang terus-menerus."

Ibn Khaldun

4.1 Asas PHP: Syntax, Pembolehubah, dan Fungsi

PHP ialah bahasa skrip yang digunakan terutamanya untuk pembangunan web. Salah satu ciri utama PHP ialah ia berfungsi di bahagian pelayan (server-side), bermakna ia memproses permintaan di pelayan sebelum menghantar hasilnya kepada pelayar web pengguna. PHP dikenali kerana fleksibilitinya, dan ia boleh digunakan untuk membina pelbagai jenis aplikasi web, dari laman web peribadi yang mudah hingga aplikasi e-dagang yang kompleks. Sintaks PHP adalah ringkas dan mudah difahami, terutamanya bagi mereka yang telah terbiasa dengan bahasa pengaturcaraan seperti C atau JavaScript.

Pembolehubah dalam PHP didefinisikan dengan simbol `$` diikuti dengan nama pembolehubah. PHP tidak memerlukan pengisytiharan jenis data (seperti integer atau string) kerana ia ialah bahasa yang ditaip secara dinamik. Ini memudahkan pembangun untuk menulis kod dengan lebih cepat tanpa perlu risau tentang pengisytiharan jenis pembolehubah. Sebagai contoh, anda boleh menetapkan pembolehubah `$nama = "Ali"`; tanpa menentukan bahawa ia adalah string. Di samping itu, PHP menyokong pelbagai jenis data seperti integer, string, array, dan objek.

Fungsi dalam PHP didefinisikan menggunakan kata kunci `function`, dan ia digunakan untuk membolehkan kod diatur dalam blok yang boleh digunakan semula. Fungsi ini boleh menerima argumen dan memulangkan nilai. Sebagai contoh, jika anda mempunyai banyak bahagian laman web yang memerlukan penukaran format tarikh, anda boleh menggunakan fungsi untuk menangani tugas ini dengan lebih efisien. Fungsi boleh menjimatkan masa pembangun dan membantu mengurangkan redundansi kod.

Selain itu, PHP menyokong pelbagai jenis struktur kawalan seperti pernyataan bersyarat (`if`, `else`, `switch`) dan gelung (`for`, `while`, `foreach`). Struktur ini membolehkan pembangun mengawal aliran kod dengan cara yang dinamik dan kondisional berdasarkan input atau data dari pangkalan data. Sebagai contoh, anda boleh menggunakan `if` untuk memeriksa status pengguna sebelum membenarkan akses ke halaman tertentu, atau menggunakan gelung `foreach` untuk memproses setiap item dalam senarai. Berikut adalah contoh kod asas PHP yang menggunakan pembolehubah dan fungsi:

Fail .php

```
<?php

// Definisi pembolehubah

$nama = "Ali";

$umur = 25;

// Fungsi untuk memaparkan mesej selamat datang

function sapaPengguna($nama, $umur) {

    return "Selamat datang, " . $nama . "! Anda berumur " . $umur

    . " tahun.";

}

// Memanggil fungsi

echo sapaPengguna($nama, $umur);

?>
```

4.2 Menghubungkan PHP dengan HTML

PHP memberikan fleksibiliti untuk mencampurkan kod PHP dengan HTML, menjadikannya bahasa yang ideal untuk mencipta laman web dinamik. PHP boleh diselitkan dalam dokumen HTML dengan menggunakan tag `<?php ?>`, di mana kod PHP dijalankan di pelayan, dan

hasilnya dihantar kembali kepada pelayar pengguna sebagai HTML. Ini bermakna anda boleh menggunakan PHP untuk menghasilkan kandungan yang dinamik, seperti memaparkan maklumat yang berbeza untuk pengguna yang berbeza atau berdasarkan input masa nyata.

Integrasi PHP dan HTML membolehkan pembangun menyesuaikan kandungan berdasarkan keadaan tertentu. Sebagai contoh, anda boleh menggunakan PHP untuk memeriksa status log masuk pengguna dan memaparkan mesej yang berbeza jika pengguna sudah log masuk atau belum. Di samping itu, PHP boleh digunakan untuk memproses data yang dihantar melalui borang HTML. Sebagai contoh, pengguna boleh menghantar maklumat mereka melalui borang pendaftaran, dan PHP akan memproses input tersebut serta menyimpannya di pangkalan data.

Selain itu, PHP boleh digunakan untuk memaparkan data dari pangkalan data terus ke halaman HTML. Ini amat berguna dalam situasi di mana anda perlu memaparkan produk dari pangkalan data e-dagang atau senarai artikel dari pangkalan data blog. PHP juga membolehkan anda mengawal bagaimana kandungan tersebut dipaparkan dengan menggunakan struktur bersyarat, seperti hanya memaparkan artikel terbaru atau produk terlaris. Dengan PHP, anda boleh memastikan bahawa kandungan laman web anda sentiasa dikemas kini tanpa perlu mengedit HTML secara manual.

Penggunaan PHP bersama HTML juga membolehkan pembangunan templat dinamik di mana satu fail PHP boleh digunakan untuk menjana pelbagai halaman dengan data yang berbeza. Sebagai contoh, laman web berita boleh menggunakan satu templat untuk memaparkan berita yang berbeza berdasarkan parameter yang diberikan melalui URL. Ini bukan sahaja menjimatkan masa pembangunan, tetapi juga memudahkan penyelenggaraan kod.

Fail .php

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>PHP dalam HTML</title>

</head>

<body>
```

```
<h1>Selamat Datang!</h1>

<p>Hari ini adalah: <?php echo date("l, d F Y"); ?></p>

</body>

</html>
```

4.3 Membina Fungsi Dinamik: Borang dan Interaksi Pengguna

Salah satu penggunaan utama PHP ialah pemrosesan borang HTML, yang membolehkan interaksi dinamik antara pengguna dan pelayan. Dalam aplikasi web dinamik, borang sering digunakan untuk mengumpulkan input dari pengguna seperti maklumat pendaftaran, komen, atau input carian. PHP menggunakan superglobal `$_POST` dan `$_GET` untuk mengakses data yang dihantar melalui borang. Data ini kemudian boleh diproses oleh PHP, seperti menyimpan maklumat dalam pangkalan data atau memberikan maklum balas kepada pengguna.

Borang boleh direka menggunakan HTML dan CSS untuk pengumpulan data, dan PHP akan mengambil alih pemrosesan setelah pengguna menghantar borang tersebut. Sebagai contoh, jika pengguna mengisi borang pendaftaran dengan nama dan alamat e-mel, PHP akan memeriksa data tersebut, membersihkannya, dan kemudian menyimpannya dalam pangkalan data. Ini juga merupakan tempat di mana anda boleh menambah logik pengesahan untuk memastikan bahawa input pengguna sah sebelum data dihantar ke pangkalan data.

Proses pengesahan input pengguna adalah penting dalam mengelakkan serangan keselamatan seperti **SQL Injection** atau **Cross-Site Scripting (XSS)**. PHP menyediakan pelbagai fungsi yang membantu memastikan bahawa input pengguna adalah sah dan selamat. Contohnya, fungsi `htmlspecialchars()` boleh digunakan untuk membersihkan input pengguna sebelum ia dipaparkan semula di pelayar web, manakala `mysqli_real_escape_string()` membantu mengelakkan SQL Injection ketika berinteraksi dengan pangkalan data.

Selepas memproses borang, PHP juga boleh memberikan maklum balas kepada pengguna. Sebagai contoh, jika borang pendaftaran berjaya diproses, anda boleh memaparkan mesej kejayaan. Jika terdapat ralat dalam input, PHP boleh memberikan maklum balas yang berguna, seperti meminta pengguna memasukkan alamat e-mel yang sah atau memastikan kata laluan mereka memenuhi syarat tertentu. Di bawah adalah contoh sintaksnya.

Fail .php

```
<!-- Borang Pendaftaran HTML -->
<form action="proses.php" method="post">
    <label for="nama">Nama:</label>
    <input type="text" id="nama" name="nama" required>

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

    <input type="submit" value="Daftar">
</form>

<!-- Proses.php untuk memproses data -->
<?php
$nama = $_POST['nama'];
$email = $_POST['email'];

// Menggunakan htmlspecialchars untuk membersihkan input
$nama = htmlspecialchars($nama);
$email = htmlspecialchars($email);

echo "Nama: " . $nama . "<br>";
echo "Email: " . $email;
?>
```


4.4 Menangani Data dan Proses Logik di Backend

PHP memainkan peranan penting dalam menangani data dan menjalankan logik backend yang kompleks. Sebagai bahasa skrip pelayan, PHP digunakan untuk memproses data yang dihantar oleh pengguna melalui borang, seperti maklumat log masuk atau pendaftaran, dan berinteraksi dengan pangkalan data untuk menyimpan atau mendapatkan data. Salah satu tugas utama PHP ialah menangani pengesahan input pengguna, memastikan bahawa data yang diterima adalah sah sebelum ia diproses atau disimpan dalam pangkalan data.

Selepas data dihantar oleh pengguna, PHP boleh menjalankan logik yang lebih kompleks untuk memprosesnya. Contohnya, dalam sistem e-dagang, PHP boleh digunakan untuk memproses pesanan pengguna, memeriksa stok produk, mengira harga keseluruhan, dan menyimpan butiran pesanan ke dalam pangkalan data. Jika pesanan berjaya diproses, PHP boleh menghantar e-mel pengesahan kepada pengguna. PHP juga boleh mengurus pengesahan kata laluan dan menyulitkan data sensitif seperti kata laluan pengguna sebelum disimpan.

Pembangun juga boleh menggunakan PHP untuk mengurus sesi pengguna. Dengan menggunakan `$_SESSION`, anda boleh menyimpan maklumat pengguna seperti ID pengguna, nama, dan status log masuk dalam sesi yang berterusan di seluruh halaman web. Ini membolehkan pengguna log masuk sekali sahaja, dan mereka boleh menavigasi ke pelbagai halaman tanpa perlu log masuk semula. **Sesi** adalah berguna dalam situasi seperti laman web e-dagang, di mana maklumat seperti item dalam troli pengguna perlu disimpan sementara pengguna meneroka laman web. Untuk memulakan sesi, anda perlu memanggil fungsi `session_start()` di bahagian atas setiap halaman yang menggunakan sesi.

Sebagai contoh, selepas pengguna log masuk dengan berjaya, anda boleh menyimpan maklumat mereka dalam sesi:

Fail .php

```
<?php
session_start();

$_SESSION['user_id'] = $user_id;

$_SESSION['user_name'] = $user_name;
```

```
// Sekarang pengguna log masuk, anda boleh rujuk maklumat mereka di mana-mana halaman
```

```
echo "Selamat datang, " . $_SESSION['user_name'];
```

```
?>
```

Sesi ini akan kekal aktif sehingga pengguna log keluar atau sesi tamat. Anda boleh menggunakan maklumat yang disimpan dalam sesi untuk menentukan hak akses pengguna, mengurus kandungan peribadi, dan memastikan bahawa hanya pengguna yang sah boleh mengakses halaman tertentu. Selain itu, jika pengguna log keluar, anda boleh menggunakan `session_destroy()` untuk menamatkan sesi dan menghapuskan semua maklumat yang disimpan dalamnya.

Logik backend yang lebih kompleks juga boleh melibatkan operasi seperti pengurusan kuki (cookies). Kuki digunakan untuk menyimpan maklumat di pelayar pengguna dalam tempoh yang lebih lama berbanding sesi. Kuki boleh digunakan untuk menyimpan maklumat seperti pilihan pengguna atau status log masuk, supaya pengguna tidak perlu log masuk semula apabila mereka kembali ke laman web pada masa hadapan. Kuki dan sesi adalah dua komponen penting dalam menangani proses logik backend, memberikan pengalaman pengguna yang lancar dan selamat.

Berikut adalah contoh kod yang menggabungkan penggunaan sesi dan pengesahan input pengguna untuk log masuk:

Fail .php

```
<?php
```

```
session_start();
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

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

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

```

// Menyemak kesahihan input pengguna
if (empty($username) || empty($password)) {
    echo "Sila masukkan nama pengguna dan kata laluan.";
} else {
    // Sambungkan ke pangkalan data dan semak maklumat log
masuk
    $conn = new mysqli("localhost", "root", "", "db_users");

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // Persediaan query untuk mencari pengguna
    $sql = "SELECT * FROM users WHERE username = '$username'
AND password = '$password'";

    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // Pengguna berjaya log masuk, simpan maklumat dalam
sesi
        $row = $result->fetch_assoc();
        $_SESSION['user_id'] = $row['id'];
        $_SESSION['username'] = $row['username'];
    }
}

```

```

                echo "Log masuk berjaya. Selamat datang, " .
$_SESSION['username'] . "!";

        } else {

                echo "Nama pengguna atau kata laluan tidak sah.";

        }

        $conn->close();

    }

}

?>

```

Dalam contoh di atas, pengguna memasukkan nama pengguna dan kata laluan dalam borang. PHP kemudian memproses input tersebut, menyemak dengan pangkalan data untuk mengesahkan maklumat log masuk, dan menyimpan maklumat pengguna dalam sesi jika log masuk berjaya. Sesi ini boleh digunakan untuk mengekalkan status log masuk pengguna di seluruh halaman web.

4.5 Laman Login dan Logout dengan PHP

Sistem **Login dan Logout** adalah asas untuk membina laman web yang memerlukan autentikasi pengguna. Dalam aplikasi web, pengguna log masuk menggunakan nama pengguna dan kata laluan yang sah. Proses ini melibatkan **PHP** untuk memproses input, **HTML** untuk membina borang, dan **CSS** untuk pengayaan. Setelah pengguna log masuk, mereka boleh mengakses kandungan peribadi, dan untuk keluar, mereka perlu menggunakan fungsi logout untuk menghapus sesi yang aktif.

Pembangunan Laman Login

Laman **Login** adalah borang yang membenarkan pengguna memasukkan **nama pengguna** dan **kata laluan**. PHP kemudian memproses input ini dengan menyemak sama ada nama pengguna

dan kata laluan yang dimasukkan sepadan dengan rekod dalam pangkalan data. Jika sepadan, sesi akan dimulakan untuk mengekalkan status log masuk pengguna.

HTML dan CSS (Borang Login):

Fail .html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width,
initial-scale=1.0">

  <title>Login</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      background-color: #f0f0f0;

      display: flex;

      justify-content: center;

      align-items: center;

      height: 100vh;

    }

    form {

      background-color: white;
```

```
padding: 20px;

border-radius: 8px;

box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
}
```

```
input[type="text"], input[type="password"] {

    width: 100%;

    padding: 10px;

    margin: 10px 0;

    border: 1px solid #ccc;

    border-radius: 4px;
}
```

```
input[type="submit"] {

    background-color: #4CAF50;

    color: white;

    padding: 10px;

    border: none;

    border-radius: 4px;

    cursor: pointer;
}
```

```
        input[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <form action="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>

        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Proses Autentikasi dengan PHP

PHP digunakan untuk memproses input yang diterima dari borang login. Nama pengguna dan kata laluan disemak terhadap pangkalan data. Jika pengguna berjaya log masuk, sesi akan dimulakan menggunakan `$_SESSION`.

PHP (Proses Login):

Fail.php

```
<?php
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Sambungan ke pangkalan data
    $conn = new mysqli("localhost", "root", "", "db_users");
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT * FROM users WHERE username = '$username' AND
password = '$password'";

    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        $_SESSION['username'] = $username;

        header("Location: welcome.php"); // Menghantar pengguna
ke halaman selamat datang
    } else {
```



```
        echo "Nama pengguna atau kata laluan tidak sah.";
    }

    $conn->close();
}
?>
```

Laman Logout

Setelah pengguna selesai menggunakan laman web, mereka perlu log keluar. Proses ini melibatkan penghapusan semua sesi aktif menggunakan `session_destroy()`.

PHP (Laman Logout):

Fail .php

```
<?php
session_start();

session_destroy(); // Memusnahkan semua sesi

header("Location: login.php"); // Mengembalikan pengguna ke laman
login

exit();

?>
```

Dengan gabungan PHP, HTML, dan CSS, anda boleh membina sistem login dan logout yang berfungsi dengan baik. Sistem ini memastikan hanya pengguna yang sah boleh mengakses kandungan peribadi dan sesi diuruskan dengan selamat.

4.6 Laman Borang Pendaftaran dengan PHP

Borang pendaftaran membolehkan pengguna memasukkan maklumat seperti nama, e-mel, dan kata laluan, yang kemudiannya disimpan ke dalam pangkalan data untuk kegunaan masa depan. **HTML** digunakan untuk membina borang pendaftaran, **CSS** untuk penggayaan, dan **PHP** untuk memproses data yang dihantar oleh pengguna dan menyimpan data tersebut dalam pangkalan data.

Struktur Borang Pendaftaran

Borang pendaftaran terdiri daripada beberapa medan input yang menerima data pengguna seperti nama, e-mel, dan kata laluan. **PHP** kemudian memproses data ini dan menyimpannya dalam pangkalan data menggunakan query SQL.

HTML dan CSS (Borang Pendaftaran):

Fail .html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Borang Pendaftaran</title>

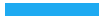
<style>

    form {

        width: 300px;

        margin: 0 auto;

        padding: 20px;
```



```
background-color: #f9f9f9;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}
```

```
input[type="text"], input[type="email"],
input[type="password"] {

width: 100%;

padding: 10px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}
```

```
input[type="submit"] {

width: 100%;

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;
```

```
        border: none;

        border-radius: 4px;

        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>

<h2>Borang Pendaftaran</h2>

<form action="register.php" method="post">
    <label for="name">Nama:</label>

    <input type="text" id="name" name="name" required>

    <label for="email">Email:</label>

    <input type="email" id="email" name="email" required>

    <label for="password">Kata Laluan:</label>
```

```
<input type="password" id="password" name="password"
required>

<input type="submit" value="Daftar">

</form>

</body>

</html>
```

Proses Pendaftaran dengan PHP

Setelah pengguna menghantar borang, PHP memproses data tersebut dan menyimpannya ke dalam pangkalan data. Untuk keselamatan, kata laluan disulitkan menggunakan `password_hash()` sebelum disimpan.

PHP (Proses Pendaftaran):

Fail .php

```
<?php

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    $name = $_POST['name'];

    $email = $_POST['email'];

    $password = password_hash($_POST['password'],
PASSWORD_DEFAULT);

    $conn = new mysqli("localhost", "root", "", "db_users");
```

```

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO users (name, email, password) VALUES
('$name', '$email', '$password)";

if ($conn->query($sql) === TRUE) {
    echo "Pendaftaran berjaya!";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
}
?>

```

Dengan gabungan **PHP, HTML, dan CSS**, anda boleh membina borang pendaftaran yang membolehkan pengguna mendaftar dan menyimpan maklumat mereka secara selamat dalam pangkalan data. Kata laluan disulitkan untuk memastikan keselamatan data pengguna.

4.7 Paparan Senarai Maklumat Pendaftaran dengan PHP

Selepas pendaftaran, maklumat pengguna boleh dipaparkan dalam bentuk jadual yang teratur. PHP digunakan untuk mendapatkan data daripada pangkalan data, **HTML** untuk membina

struktur jadual, dan **CSS** untuk pengayaan. Setiap baris dalam jadual memaparkan maklumat pengguna yang telah mendaftar, seperti nama dan e-mel.

Paparan Senarai Pengguna

Anda boleh menggunakan **PHP** untuk mendapatkan maklumat pengguna daripada pangkalan data dan memaparkannya dalam bentuk **jadual**. Ini amat berguna dalam aplikasi yang memerlukan pengurusan pengguna, seperti sistem pendaftaran acara atau pengurusan akaun.

HTML dan CSS untuk Paparan Senarai Pendaftaran

Berikut adalah contoh bagaimana data pendaftaran pengguna boleh dipaparkan dalam bentuk jadual. **CSS** digunakan untuk menjadikan jadual kelihatan kemas dan mudah dibaca dengan warna latar alternatif bagi setiap baris.

HTML dan CSS (Paparan Senarai Pendaftaran):

Fail .html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Senarai Pendaftaran</title>

<style>

    table {

        width: 70%;

        margin: 20px auto;

        border-collapse: collapse;
```



```
    }

    table, th, td {
        border: 1px solid black;
    }

    th, td {
        padding: 12px;
        text-align: left;
    }

    th {
        background-color: #4CAF50;
        color: white;
    }

    tr:nth-child(even) {
        background-color: #f2f2f2;
    }
</style>
</head>
<body>
```

```
<h2>Senarai Pendaftaran Pengguna</h2>
```

```
<table>
```

```
  <tr>
```

```
    <th>Nama</th>
```

```
    <th>Email</th>
```

```
    <th>Tindakan</th>
```

```
</tr>
```

```
<?php
```

```
    // PHP untuk mendapatkan dan memaparkan maklumat dari
    pangkalan data
```

```
    $conn = new mysqli("localhost", "root", "", "db_users");
```

```
    if ($conn->connect_error) {
```

```
        die("Connection failed: " . $conn->connect_error);
```

```
    }
```

```
    $sql = "SELECT id, name, email FROM users";
```

```
    $result = $conn->query($sql);
```

```
    if ($result->num_rows > 0) {
```

```

while ($row = $result->fetch_assoc()) {
    echo "<tr>
        <td>" . $row['name'] . "</td>
        <td>" . $row['email'] . "</td>
        <td><a href='delete.php?id=" . $row['id'] .
''>Padam</a></td>
        </tr>";
    }
} else {
    echo "<tr><td colspan='3'>Tiada pendaftaran.</td></tr>";
}

$conn->close();
?>
</table>

</body>
</html>

```

PHP untuk Memaparkan Data dari Pangkalan Data

Dalam contoh di atas, **PHP** digunakan untuk mendapatkan maklumat pendaftaran dari pangkalan data dan memaparkannya dalam jadual. Untuk setiap pengguna yang telah mendaftar, satu baris akan dipaparkan dengan butang **Padam** yang membolehkan pentadbir memadam pengguna tersebut.

PHP (Fungsi Padam Pengguna):

Fail .php

```
<?php

// Memadam pengguna berdasarkan ID

if (isset($_GET['id'])) {

    $id = $_GET['id'];

    $conn = new mysqli("localhost", "root", "", "db_users");

    if ($conn->connect_error) {

        die("Connection failed: " . $conn->connect_error);

    }

    $sql = "DELETE FROM users WHERE id = $id";

    if ($conn->query($sql) === TRUE) {

        echo "Pengguna berjaya dipadam!";

    } else {

        echo "Error: " . $conn->error;

    }

    $conn->close();

}
```

```
        // Kembali ke halaman senarai pendaftaran selepas pengguna  
dipadam  
  
        header("Location: list.php");  
  
        exit();  
  
    }  
  
?>
```

Dengan gabungan **PHP, HTML, dan CSS**, anda boleh membina sistem yang memaparkan senarai pengguna yang telah mendaftar dalam bentuk jadual. Penggunaan PHP membolehkan interaksi dinamik dengan pangkalan data, seperti memadam pengguna. **CSS** digunakan untuk memastikan paparan jadual lebih menarik dan mudah dibaca oleh pentadbir atau pengguna.

Bab 5: Pengurusan Pangkalan Data dengan PhpMyAdmin

“Ilmu adalah sahabat terbaik yang dapat membawa seseorang keluar dari kegelapan kejahilan menuju cahaya kebenaran.”

Al-Razi (Rhazes)

5.1 Pengenalan kepada Pangkalan Data MySQL

Pangkalan data **MySQL** adalah sistem pengurusan pangkalan data berasaskan SQL (Structured Query Language) yang digunakan secara meluas untuk aplikasi web. Ia menyimpan data dalam bentuk **jadual**, yang terdiri daripada baris dan lajur. MySQL membolehkan pengguna menyimpan, mengurus, dan mengakses data secara efisien. Dalam pembangunan web, MySQL sering digunakan untuk menyimpan maklumat pengguna, data produk, catatan blog, dan sebagainya.

MySQL berfungsi dengan pelbagai bahasa pengaturcaraan seperti **PHP**, Python, dan Java. Kelebihan utamanya termasuk keupayaan untuk mengendalikan sejumlah besar data, integrasi yang lancar dengan pelbagai aplikasi, dan kebolehan untuk menjalankan operasi yang kompleks seperti **JOIN**, **GROUP BY**, dan subquery. MySQL juga menyokong keselamatan data dengan kawalan akses pengguna dan enkripsi.

PhpMyAdmin adalah alat berbasis web yang memudahkan pengurusan pangkalan data MySQL. Ia menyediakan antara muka grafik (GUI) yang mesra pengguna untuk mencipta dan mengurus jadual, menjalankan query SQL, dan memantau pangkalan data. Ini menjadikan pengurusan pangkalan data lebih mudah untuk pembangun yang mungkin tidak selesa menggunakan baris arahan.

MySQL menggunakan model **relasi** untuk menyusun data, di mana data disusun dalam jadual yang berhubungan antara satu sama lain. Setiap jadual mempunyai **primary key** yang unik dan **foreign key** yang boleh digunakan untuk menghubungkan data antara jadual yang berbeza. Sebagai contoh, anda boleh mempunyai jadual `users` dengan primary key `user_id` dan jadual

orders dengan foreign key `user_id` yang menghubungkannya ke jadual `users`. Contoh SQL asas untuk mencipta pangkalan data:

Fail .sql

```
CREATE DATABASE db_users;
```

5.2 Mencipta dan Mengurus Jadual dalam PhpMyAdmin

Dalam **PhpMyAdmin**, anda boleh mencipta jadual baru dalam pangkalan data dengan mudah melalui antara muka grafik. Setelah pangkalan data dicipta, anda boleh mula mencipta jadual yang akan menyimpan data dalam baris dan lajur. Setiap lajur ditakrifkan dengan jenis data seperti **INT** untuk integer, **VARCHAR** untuk teks, dan **DATE** untuk tarikh. Anda juga boleh menetapkan **primary key** untuk setiap jadual bagi memastikan setiap rekod adalah unik.

Untuk mencipta jadual, anda perlu menentukan nama jadual dan bilangan medan yang anda perlukan. Setiap medan juga perlu ditentukan dengan jenis data dan sama ada ia memerlukan nilai lalai atau bukan (**NOT NULL**). Sebagai contoh, jadual pengguna boleh mengandungi medan seperti `user_id` (primary key), `nama`, `email`, dan `password`.

PhpMyAdmin juga membenarkan anda menambah, mengubah suai, atau memadam rekod dengan menggunakan antara muka tanpa perlu menulis SQL secara manual. Ini amat membantu untuk pengguna yang baru belajar SQL dan masih tidak mahir dengan syntax SQL. Selain itu, anda boleh menggunakan PhpMyAdmin untuk mengeksport dan mengimport pangkalan data, memudahkan proses penyedaran dan pemulihan data.

Satu ciri penting dalam pengurusan jadual adalah **indexing**, di mana anda boleh menetapkan indeks pada medan tertentu untuk mempercepatkan carian data. Sebagai contoh, jika anda kerap mencari rekod berdasarkan e-mel pengguna, anda boleh menetapkan medan `email` sebagai indeks untuk mempercepatkan operasi tersebut. Contoh SQL untuk mencipta jadual:

Fail .sql

```
CREATE TABLE users (  
    user_id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(50) NOT NULL,
```

```
password VARCHAR(255) NOT NULL,  
reg_date TIMESTAMP  
);
```

5.3 Menghubungkan Pangkalan Data dengan PHP

Menghubungkan **PHP** dengan pangkalan data MySQL adalah salah satu langkah penting dalam pembangunan aplikasi web dinamik. PHP menyediakan fungsi terbina dalam seperti `mysqli_connect()` dan **PDO** (PHP Data Objects) untuk mengurus sambungan ke pangkalan data. Setelah sambungan berjaya, PHP boleh digunakan untuk menjalankan query SQL, mendapatkan data dari pangkalan data, dan memprosesnya untuk dipaparkan di laman web.

Untuk memulakan, anda perlu menentukan butiran sambungan seperti **hostname**, **nama pengguna** MySQL, **kata laluan**, dan **nama pangkalan data**. Dalam kebanyakan kes, hostname adalah `localhost`, melainkan anda menggunakan pelayan pangkalan data yang dihoskan di tempat lain. Selepas menyambungkan PHP ke pangkalan data, anda boleh mula menjalankan operasi seperti **Create**, **Read**, **Update**, dan **Delete** (CRUD) terhadap jadual yang telah anda cipta.

Apabila menggunakan sambungan **MySQLi** atau **PDO**, adalah penting untuk menggunakan teknik **prepared statements** untuk melindungi aplikasi anda daripada serangan **SQL injection**. Prepared statements membenarkan PHP untuk memisahkan logik query daripada data input, memastikan input yang dimasukkan oleh pengguna tidak boleh digunakan untuk merosakkan query SQL. Contoh kod untuk sambungan pangkalan data menggunakan **MySQLi**:

Fail .php

```
<?php  
  
$servername = "localhost";  
  
$username = "root";  
  
$password = "";  
  
$dbname = "db_users";  
  
  
  
// Membuat sambungan
```

```

$conn = new mysqli($servername, $username, $password, $dbname);

// Semak sambungan
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

echo "Sambungan berjaya!";

?>

```

5.4 CRUD (Create, Read, Update, Delete) Operasi Asas

Operasi **CRUD** adalah asas kepada pengurusan data dalam pangkalan data. **Create** bermaksud mencipta rekod baru, **Read** untuk mendapatkan data dari pangkalan data, **Update** untuk mengemas kini rekod sedia ada, dan **Delete** untuk memadam rekod. Menggunakan **PHP** dan **MySQL**, anda boleh melaksanakan operasi CRUD ini untuk mengurus data pengguna, produk, atau apa sahaja jenis data yang disimpan dalam pangkalan data.

Create (Cipta Rekod)

Untuk mencipta rekod baru, anda menggunakan pernyataan **INSERT INTO** dalam SQL. Sebagai contoh, untuk menambah pengguna baru ke dalam jadual `users`, anda perlu menentukan medan yang akan diisi seperti `name`, `email`, dan `password` seperti yang diberikan di bawah:

Fail .php

```

$sql = "INSERT INTO users (name, email, password) VALUES ('Ali',
'ali@example.com', 'password123')";

if ($conn->query($sql) === TRUE) {
    echo "Rekod baru berjaya ditambah!";
}

```


Read (Baca Rekod)

Untuk mendapatkan data daripada pangkalan data, anda menggunakan pernyataan **SELECT**. Contohnya, jika anda ingin mendapatkan semua pengguna dari jadual `users`, anda boleh menulis query seperti berikut:

Fail .php

```
$sql = "SELECT id, name, email FROM users";

$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "ID: " . $row["id"]. " - Nama: " . $row["name"]. " -
Email: " . $row["email"]. "<br>";
    }
} else {
    echo "Tiada hasil.";
}
```

Update (Kemaskini Rekod)

Untuk mengemas kini rekod sedia ada, anda menggunakan pernyataan **UPDATE**. Contohnya, jika anda ingin menukar alamat e-mel pengguna dengan `id = 1`, anda boleh menulis query seperti berikut:

Fail .php

```
$sql = "UPDATE users SET email='ali_baru@example.com' WHERE
id=1";

if ($conn->query($sql) === TRUE) {
    echo "Rekod berjaya dikemaskini!"; }
}
```

Delete (Padam Rekod)

Untuk memadam rekod, anda menggunakan pernyataan **DELETE**. Contohnya, jika anda ingin memadam pengguna dengan `id = 1`, anda boleh menulis query seperti berikut:

Fail .php

```
$sql = "DELETE FROM users WHERE id=1";  
  
if ($conn->query($sql) === TRUE) {  
    echo "Rekod berjaya dipadam!";  
}
```

Dengan menggunakan operasi CRUD ini, anda boleh mengurus semua aspek pengurusan data dalam pangkalan data dengan mudah.

5.5 Penggunaan AI untuk Automasi Operasi Pangkalan Data

Kecerdasan Buatan (**AI**) kini memainkan peranan penting dalam memudahkan dan mempercepatkan operasi pangkalan data. AI boleh digunakan untuk **automasi** tugas-tugas rutin seperti penciptaan query, pengoptimuman indeks, dan bahkan pembersihan data. Alat seperti **AI-driven SQL query generators** membolehkan pembangun menulis query SQL yang kompleks dengan lebih pantas melalui cadangan automatik yang diberikan oleh AI.

Selain itu, AI boleh digunakan untuk **pemantauan prestasi** pangkalan data, di mana ia boleh mengesan corak penggunaan dan memberikan cadangan untuk pengoptimuman. Sebagai contoh **AI** boleh digunakan untuk **pemantauan prestasi** pangkalan data dengan mengesan corak penggunaan, memberikan cadangan untuk pengoptimuman query, dan membantu pembangun mengenal pasti kelemahan dalam reka bentuk pangkalan data. Alat ini boleh menganalisis penggunaan sumber seperti memori, pemprosesan, dan ruang storan untuk memberi cadangan yang tepat tentang cara mengoptimumkan pangkalan data anda. Sebagai contoh, AI boleh mencadangkan indeks tambahan untuk mempercepatkan query tertentu atau membantu menentukan jadual yang mungkin perlu dioptimumkan kerana beban kerja yang tinggi.

Selain itu, **pembersihan data automatik** dengan bantuan AI membolehkan anda mengenal pasti rekod yang duplikasi, data yang tidak lengkap, atau data yang tidak konsisten dengan lebih mudah. AI boleh mengenal pasti pola yang sukar untuk dikesan oleh manusia dan memastikan data dalam pangkalan data lebih konsisten dan bersih. Ini amat berguna dalam aplikasi yang berurusan dengan jumlah data yang besar, seperti e-dagang atau analisis data.

Dalam pengurusan **backup dan pemulihan data**, AI boleh membantu automasi proses tersebut dengan memastikan backup dilakukan secara berkala dan memberikan cadangan untuk jadual pemulihan berdasarkan pola penggunaan sistem. Ini memastikan bahawa data sentiasa selamat dan boleh dipulihkan dengan cepat sekiranya berlaku kegagalan sistem.

Alat AI juga boleh digunakan untuk **automasi pengurusan keselamatan**. Contohnya, AI boleh memantau akses pangkalan data dan mengenal pasti sebarang aktiviti yang mencurigakan, seperti percubaan akses tanpa kebenaran atau query yang tidak biasa yang mungkin menunjukkan potensi ancaman keselamatan. Dengan analisis tingkah laku yang dilakukan secara automatik oleh AI, sistem boleh dioptimumkan untuk mengurangkan risiko keselamatan dengan lebih cepat daripada kaedah tradisional.

Berikut adalah contoh bagaimana AI boleh digunakan untuk membantu automasi operasi pangkalan data, seperti **cadangan query** menggunakan alat AI:

Fail .sql

```
-- Alat AI menjana query SQL yang sesuai berdasarkan deskripsi  
  
SELECT name, email  
  
FROM users  
  
WHERE registered_date >= '2023-01-01'  
  
ORDER BY registered_date DESC;
```

Dalam contoh di atas, AI boleh menjana query berdasarkan arahan semula jadi yang diberikan oleh pembangun, seperti “Dapatkan senarai pengguna yang berdaftar selepas Januari 2023.” Ini mengurangkan masa yang diperlukan untuk menulis query secara manual dan memudahkan tugas pembangun yang mungkin tidak begitu arif tentang syntax SQL yang kompleks.

Bab 6: Gabungan Komponen dan Pembangunan Projek

“Mencari ilmu adalah tanggungjawab setiap manusia, dan jalan untuk mencapainya adalah melalui penyelidikan dan eksperimen.”

Al-Biruni

6.1 Gabungan Frontend dan Backend

Gabungan **frontend** dan **backend** dalam pembangunan aplikasi web melibatkan penyepaduan antara komponen visual yang dilihat oleh pengguna (frontend) dan logik yang dijalankan di pelayan (backend). **Frontend** terdiri daripada teknologi seperti **HTML**, **CSS**, dan **JavaScript**, yang bertanggungjawab untuk mencipta antara muka pengguna dan interaksi. **Backend** pula menggunakan **PHP**, **MySQL**, dan teknologi server-side lain untuk memproses data, menyimpan maklumat, dan mengendalikan logik aplikasi.

Dalam pembangunan aplikasi yang dinamik, **frontend** menghantar permintaan (request) ke **backend** melalui borang atau interaksi pengguna, yang kemudian diproses oleh server. Sebagai contoh, apabila pengguna mengisi borang pendaftaran, **frontend** menghantar maklumat ke **backend**, yang memproses data tersebut dengan PHP dan menyimpannya ke pangkalan data MySQL. Setelah data diproses, **backend** memberikan maklum balas (response) yang akan dipaparkan kepada pengguna di **frontend**.

Frontend bertanggungjawab dalam memberikan pengalaman pengguna yang mesra dan mudah difahami, manakala backend memastikan bahawa logik aplikasi dijalankan dengan betul. Komunikasi antara frontend dan backend boleh dilakukan menggunakan **HTTP requests**, di mana **GET** dan **POST** adalah dua kaedah yang paling biasa digunakan. **GET** digunakan untuk mendapatkan data dari server, manakala **POST** digunakan untuk menghantar data ke server.

Sebagai contoh, apabila pengguna log masuk ke aplikasi, frontend akan menghantar permintaan **POST** dengan nama pengguna dan kata laluan melalui borang login. Backend akan menyemak maklumat tersebut dalam pangkalan data, dan jika berjaya, ia akan memulakan sesi pengguna

dan menghantar respons ke frontend untuk memaparkan laman peribadi pengguna. Berikut adalah contoh gabungan frontend dan backend dalam laman log masuk:

Fail .php

```
// PHP (Backend) - Proses Login
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Sambungan ke pangkalan data
    $conn = new mysqli("localhost", "root", "", "db_users");

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT * FROM users WHERE username = '$username' AND
password = '$password'";

    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        session_start();
        $_SESSION['username'] = $username;
        header("Location: dashboard.php");
    } else {
```

```
        echo "Log masuk gagal.";
    }

    $conn->close();
}
```

6.2 Pembangunan Sistem Login Menggunakan PHP dan MySQL

Sistem login adalah salah satu ciri paling asas dan penting dalam pembangunan aplikasi web yang memerlukan **autentikasi pengguna**. Dengan PHP dan MySQL, sistem login boleh dibina dengan menyimpan butiran pengguna seperti **nama pengguna** dan **kata laluan** dalam pangkalan data MySQL, dan PHP akan digunakan untuk memproses data login serta memulakan sesi pengguna yang berjaya log masuk.

Proses pembangunan sistem login bermula dengan membina **borang login** di frontend menggunakan HTML. Apabila pengguna menghantar borang, PHP akan memproses input yang diterima dengan mengakses pangkalan data untuk menyemak nama pengguna dan kata laluan. Jika nama pengguna dan kata laluan sah, PHP akan memulakan sesi pengguna menggunakan `$_SESSION`, yang membolehkan pengguna kekal log masuk semasa mereka menavigasi laman web.

Sistem login juga perlu menyertakan keselamatan asas seperti **penyulitan kata laluan** menggunakan `password_hash()` semasa menyimpan kata laluan dalam pangkalan data, serta menggunakan **prepared statements** untuk mengelakkan serangan **SQL Injection**. Ini memastikan bahawa data pengguna dilindungi dan serangan luar dapat dikurangkan.

Selain log masuk, sistem ini perlu merangkumi fungsi **log keluar**, di mana sesi pengguna akan dimusnahkan apabila mereka keluar dari aplikasi. Proses ini dilakukan dengan fungsi `session_destroy()`, yang memastikan bahawa maklumat sesi tidak dapat diakses lagi selepas pengguna keluar.

Contoh Kod (Sistem Login Menggunakan PHP dan MySQL):

Fail .php

```
// Sambungan ke pangkalan data

$conn = new mysqli("localhost", "root", "", "db_users");
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Semak nama pengguna dan kata laluan dalam pangkalan data
    $stmt = $conn->prepare("SELECT * FROM users WHERE username = ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $user = $result->fetch_assoc();

        // Semak kata laluan
        if (password_verify($password, $user['password'])) {
            session_start();
            $_SESSION['username'] = $username;
            header("Location: dashboard.php");
        } else {
            echo "Kata laluan salah.";
        }
    } else {
```

```
        echo "Nama pengguna tidak wujud.";
    }
}
```

6.3 Menambah Fungsi AI dalam Projek Aplikasi Web

Kecerdasan Buatan (AI) kini semakin banyak digunakan dalam pembangunan aplikasi web untuk menambah nilai dan memperbaiki pengalaman pengguna. Dalam konteks aplikasi web, AI boleh digunakan untuk pelbagai fungsi seperti **pencarian pintar**, **cadangan produk**, **pengelasan data**, dan **chatbots**. Salah satu cara mudah untuk menambah fungsi AI dalam projek web adalah dengan menggunakan **API AI**, seperti **OpenAI** atau **Google AI**, yang memberikan akses kepada model pembelajaran mesin yang canggih.

Sebagai contoh, dalam laman web e-dagang, AI boleh digunakan untuk menyediakan cadangan produk berdasarkan sejarah pembelian pengguna atau tingkah laku pelayaran mereka. Dengan menggunakan **algoritma pembelajaran mesin**, sistem AI dapat mengenali pola dan memberikan cadangan yang relevan kepada pengguna. Ini meningkatkan pengalaman pengguna dan boleh meningkatkan jualan dalam aplikasi e-dagang.

AI juga boleh digunakan untuk **chatbots**, di mana ia memberikan sokongan pelanggan yang lebih responsif dan berkesan. Chatbot yang dikuasakan oleh AI boleh memahami bahasa semula jadi pengguna dan memberikan jawapan yang relevan kepada soalan atau masalah mereka. Ini membantu mengurangkan beban sokongan manusia dan menyediakan perkhidmatan yang lebih pantas kepada pelanggan.

Integrasi AI ke dalam aplikasi web melibatkan penggunaan API AI yang berkomunikasi dengan aplikasi backend, memproses data, dan mengembalikan hasil yang relevan kepada frontend. Sebagai contoh, untuk menambah chatbot ke laman web, frontend akan menghantar input pengguna ke backend, di mana model AI memproses input tersebut dan memberikan respons yang relevan untuk dipaparkan di frontend.

Contoh Kod (Menggunakan API AI untuk Pencarian Pintar):

Fail .php

```
// Menggunakan API OpenAI untuk pencarian pintar
$api_key = 'API_KEY_ANDA';
$query = 'Cari produk terbaik untuk telefon bimbit';

$url =
'https://api.openai.com/v1/engines/text-davinci-003/completions';
$data = array(
    'prompt' => $query,
    'max_tokens' => 100
);
$options = array(
    'http' => array(
        'header' => "Content-Type:
application/json\r\nAuthorization: Bearer $api_key\r\n",
        'method' => 'POST',
        'content' => json_encode($data)
    )
);
$content = stream_context_create($options);
$result = file_get_contents($url, false, $content);
$response = json_decode($result, true);

echo $response['choices'][0]['text'];
```

6.4 Ujian dan Penambahbaikan Projek

Ujian adalah proses penting dalam pembangunan aplikasi web untuk memastikan bahawa semua fungsi beroperasi dengan betul dan tiada pepijat yang mungkin menjejaskan pengalaman pengguna. Terdapat pelbagai jenis ujian yang perlu dilakukan dalam projek aplikasi web, termasuk **ujian fungsi**, **ujian integrasi**, **ujian kebolehgunaan**, dan **ujian keselamatan**. Setiap jenis ujian ini memainkan peranan penting dalam mengesahkan bahawa aplikasi berfungsi seperti yang diharapkan.

Ujian fungsi memastikan setiap komponen aplikasi berjalan dengan betul mengikut reka bentuk asal. Sebagai contoh, dalam sistem login, ujian perlu dilakukan untuk memeriksa sama ada pengguna boleh log masuk dengan nama pengguna dan kata laluan yang sah, serta apakah berlaku ralat apabila input yang tidak sah dimasukkan.

Ujian integrasi pula memastikan bahawa komponen frontend dan backend berfungsi dengan lancar bersama-sama. Dalam konteks gabungan frontend dan backend, ujian perlu memastikan bahawa frontend menghantar input dengan betul ke backend, dan backend memberikan respons yang sesuai kepada frontend.

Selain itu, **ujian keselamatan** perlu dilakukan untuk memastikan bahawa aplikasi selamat daripada ancaman seperti serangan **SQL Injection**, **Cross-Site Scripting (XSS)**, dan percubaan akses tanpa kebenaran. Ini termasuk menguji input pengguna, melindungi data sensitif seperti kata laluan, dan memastikan **akses yang sah** kepada maklumat peribadi pengguna. Sebagai contoh, dalam sistem login, ujian keselamatan perlu memastikan bahawa kata laluan pengguna disimpan dengan betul dalam bentuk yang telah disulitkan, dan percubaan log masuk yang mencurigakan seperti serangan brute force dapat dikesan dan dihalang.

Selepas ujian dijalankan, penambahbaikan dilakukan berdasarkan hasil ujian tersebut. **Penambahbaikan** ini boleh melibatkan pembetulan pepijat, pengoptimuman prestasi aplikasi, atau penambahbaikan pengalaman pengguna. Misalnya, jika ujian kebolehgunaan menunjukkan bahawa proses pendaftaran terlalu rumit, pembangun boleh mempermudah borang pendaftaran atau menambah panduan pengguna untuk membantu mereka mengisi borang dengan betul.

Selain itu, **ujian automasi** juga semakin popular dalam projek aplikasi web yang kompleks. Alat seperti **Selenium** dan **JUnit** membolehkan pembangun menjalankan ujian automatik secara berkala, terutamanya apabila terdapat perubahan pada kod. Ujian automatik memastikan bahawa tiada fungsi penting yang terjejas akibat perubahan pada kod dan mengurangkan risiko pengenalan pepijat baru. Contoh Kod (Ujian untuk Log Masuk) seperti di bawah:

Fail.php

```
// Ujian untuk memastikan sistem login berfungsi

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    $username = $_POST['username'];

    $password = $_POST['password'];

    if (empty($username) || empty($password)) {

        echo "Nama pengguna dan kata laluan diperlukan.";

    } else {

        $conn = new mysqli("localhost", "root", "", "db_users");

        $stmt = $conn->prepare("SELECT * FROM users WHERE
username = ?");

        $stmt->bind_param("s", $username);

        $stmt->execute();

        $result = $stmt->get_result();

        if ($result->num_rows > 0) {

            $user = $result->fetch_assoc();

            if (password_verify($password, $user['password'])) {

                echo "Log masuk berjaya!";

            }

        }

    }

}
```

```
        } else {  
            echo "Kata laluan salah.";  
        }  
    } else {  
        echo "Nama pengguna tidak wujud.";  
    }  
}  
}
```

6.5 Memasang Aplikasi Web ke Pelayan Tempatan dengan XAMPP

Memasang aplikasi web ke **pelayan tempatan** (localhost) adalah langkah penting dalam pembangunan sebelum aplikasi dipindahkan ke pelayan produksi. **XAMPP** adalah salah satu persekitaran pembangunan pelayan tempatan yang paling popular, yang menyediakan **Apache** sebagai pelayan web, **MySQL** sebagai sistem pengurusan pangkalan data, dan **PHP** sebagai bahasa skrip untuk backend. Dengan XAMPP, anda boleh menguji aplikasi web sepenuhnya sebelum menyebarkannya ke pelayan langsung.

Langkah pertama dalam memasang aplikasi web ke XAMPP ialah memuat turun dan memasang XAMPP pada komputer tempatan. Setelah dipasang, fail aplikasi anda perlu diletakkan dalam direktori `htdocs`, yang merupakan root direktori untuk semua aplikasi web yang dijalankan melalui XAMPP. Anda boleh mengakses aplikasi anda dengan mengunjungi `http://localhost/nama-aplikasi/` di pelayar web.

Seterusnya, anda perlu menyediakan **pangkalan data MySQL** menggunakan PhpMyAdmin, yang disertakan dalam XAMPP. Anda boleh membuat pangkalan data baru dan mencipta jadual yang diperlukan untuk aplikasi anda. Sambungan PHP ke pangkalan data juga perlu dikemas kini dengan menggunakan butiran pelayan tempatan (seperti hostname `localhost`, nama pengguna `root`, dan tanpa kata laluan secara lalai).

Apabila aplikasi telah berfungsi dengan baik di pelayan tempatan, anda boleh mula melakukan pengujian akhir untuk memastikan bahawa semuanya berjalan lancar. XAMPP menyediakan

persekitaran yang serupa dengan pelayan produksi, membolehkan pembangun mengesan dan menyelesaikan isu yang mungkin timbul sebelum aplikasi dipindahkan ke pelayan langsung. Contoh Kod (Sambungan Pelayan Tempatan Menggunakan XAMPP) seperti di bawah:

Fail .php

```
// Sambungan ke pangkalan data di pelayan tempatan

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "db_users";

// Membuat sambungan ke MySQL

$conn = new mysqli($servername, $username, $password, $dbname);

// Semak sambungan

if ($conn->connect_error) {

    die("Sambungan gagal: " . $conn->connect_error);

}

echo "Sambungan berjaya!";
```

Setelah aplikasi diuji dan berfungsi dengan baik di XAMPP, anda boleh mula mempertimbangkan langkah-langkah untuk memindahkan aplikasi tersebut ke pelayan produksi. Ini termasuk menyandakan pangkalan data, memindahkan fail aplikasi, dan mengemas kini tetapan pelayan untuk pelayan produksi.

Penulis



Ts. Izwan Suhadak Ishak adalah seorang pensyarah UNISEL dalam bidang Sains Komputer dan seorang Professional Technologist yang diiktiraf oleh Malaysia Board of Technologists (MBOT). Beliau memiliki pengalaman luas dalam pengajaran dan pembangunan modul pendidikan, terutamanya dalam pengaturcaraan dan teknologi kecerdasan buatan (AI). Izwan aktif dalam mempromosikan pendidikan teknologi kepada pelajar-pelajar sekolah menengah dan universiti melalui bengkel, buku panduan, dan latihan khusus.

Beliau juga merupakan jurulatih *Modul Keibubapaan Digital* dan *Bina Website Guna CANVA*. Selain itu, beliau juga memainkan peranan penting dalam pelaksanaan projek Eko Wellness yang menggabungkan teknologi dengan alam semula jadi untuk kesejahteraan mental dan fizikal masyarakat.



Ts. Roziyani Setik adalah seorang teknologis dan pensyarah di Jabatan Pengkomputeran UNISEL yang memiliki pengalaman dalam bidang pengkomputeran, khususnya dalam teknologi web dan kecerdasan buatan. Beliau berpengalaman dalam pembangunan aplikasi web, perkhidmatan teknologi pendidikan, dan pengajaran kepada pelajar-pelajar sekolah dan institusi pengajian tinggi. Sebagai seorang yang komited kepada inovasi dalam pendidikan teknologi, Roziyani telah terlibat dalam Beliau juga merupakan tenaga pengajar *Modul CANVA dan AI* dan pelbagai projek yang bertujuan untuk memupuk kreativiti dan pemikiran kritis dalam kalangan pelajar melalui

penggunaan teknologi terkini. Beliau juga merupakan Penasihat Persatuan Mahasiswa Teknologi Maklumat di UNISEL Kampus Bestari Jaya.

Kedua-dua penulis ini bergabung untuk menghasilkan buku *Syntax Champions: Empowering Code With AI* dengan tujuan untuk memperkasa pelajar dengan kemahiran pengaturcaraan sambil memperkenalkan peranan kecerdasan buatan dalam pembangunan perisian.

