

PEMROGRAMAN WEB 101: MEMAHAMI DASAR-DASAR UNTUK MENGEMBANGKAN SITUS WEB



PENULIS:

**Ade Maulana,
Rahmat Taufik R.L Bau,
Hermila A.,
Zen Munawar,
Rony Setiawan,
Sitti Aisa,
Wirawan Istiono,
Andi Irfan,
Yudhistira Adhitya Pratama,
Emha Diambang Ramadhany,
Suwito Pomalingo,
Angga Aditiya Permana**

PEMROGRAMAN WEB 101: MEMAHAMI DASAR-DASAR UNTUK MENGEMBANGKAN SITUS WEB

**Ade Maulana
Rahmat Taufik R.L Bau
Hermila A.
Zen Munawar
Rony Setiawan
Sitti Aisa
Wirawan Istiono
Andi Irfan
Yudhistira Adhitya Pratama
Emha Diambang Ramadhany
Suwito Pomalingo
Angga Aditiya Permana**



GET PRESS INDONESIA

PEMROGRAMAN WEB 101: MEMAHAMI DASAR-DASAR UNTUK MENGEMBANGKAN SITUS WEB

Penulis :

Ade Maulana
Rahmat Taufik R.L Bau
Hermila A.
Zen Munawar
Rony Setiawan
Sitti Aisa
Wirawan Istiono
Andi Irfan
Yudhistira Adhitya Pratama
Emha Diambang Ramadhany
Suwito Pomalingo
Angga Aditiya Permana

ISBN : 978-623-198-534-7

Editor : Diana Purnama Sari, S.E., M.E

Penyunting : Ari Yanto, M.Pd

Desain Sampul dan Tata Letak : Atyka Trianisa, S.Pd

Penerbit : GET PRESS INDONESIA

Anggota IKAPI No. 033/SBA/2022

Redaksi :

Jln. Palarik Air Pacah No 26 Kel. Air Pacah
Kec. Koto Tangah Kota Padang Sumatera Barat
Website : www.getpress.co.id
Email : adm.getpress@gmail.com

Cetakan pertama, 28 Juli 2023

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk
dan dengan cara apapun tanpa izin tertulis dari penerbit.

KATA PENGANTAR

Segala Puji dan syukur atas kehadiran Allah SWT dalam segala kesempatan. Sholawat beriring salam dan doa kita sampaikan kepada Nabi Muhammad SAW. Alhamdulillah atas Rahmat dan Karunia-Nya penulis telah menyelesaikan Buku Pemrograman Web 101: Memahami Dasar-Dasar Untuk Mengembangkan Situs Web ini.

Buku Ini Membahas Pengantar Pemrograman Web, Dasar-dasar HTML, Dasar-dasar CSS, Dasar-dasar JavaScript, Membuat Halaman Responsif, Pengenalan jQuery, Interaksi dengan Server melalui AJAX, Dasar-dasar Basis Data, Pengenalan PHP, Pengelolaan Formulir, Keamanan dan Perlindungan terhadap Serangan, Tren dan Perkembangan Terkini dalam Pemrograman Web.

Penulis ucapkan terima kasih kepada semua pihak yang telah mendukung dalam penyelesaian buku ini. Terutama pihak yang telah membantu terbitnya buku ini dan telah mempercayakan mendorong, dan menginisiasi terbitnya buku ini. Semoga buku ini dapat bermanfaat bagi masyarakat Indonesia.

Padang, 28 Juli 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
BAB 1 PENGANTAR PEMROGRAMAN WEB.....	1
1.1 Sejarah dan Evolusi Web	1
1.2 Cara Kerja Web.....	5
1.3 Bahasa Pemrograman.....	6
1.4 Arsitektur Web	7
1.4.1 Arsitektur <i>Client-Server</i>	9
1.4.2 Arsitektur Single Page Application	10
1.4.3 Arsitektur <i>Microservice</i>	11
1.4.4 Arsitektur <i>Serverless</i>	12
1.4.5 Progressive Web Application	13
1.5 Struktur Dasar Halaman Web	14
1.6 Pengembangan Back-end dan Front-end.....	16
1.7 Web UI/UX Design	17
1.8 Situs Web dan Aplikasi Berbasis Web.....	18
DAFTAR PUSTAKA	20
BAB 2 DASAR-DASAR HTML	21
2.1 Pendahuluan	21
2.2 Pengenalan HTML.....	23
2.3 Pengenalan Tag HTML	24
2.3.1 Klasifikasi Tag HTML Dasar	25
2.3.2 Ketentuan Penggunaan Tag HTML.....	28
2.4 Pengenalan Atribut dasar HTML	29
DAFTAR PUSTAKA	32
BAB 3 DASAR-DASAR CSS	33
3.1 Pendahuluan	33
3.2 Fungsi CSS	35
3.3 Dasar Penggunaan CSS.....	36
3.3.1 Selektor	37

3.3.2 Property	38
3.3.3 Nilai.....	38
3.3.4 Kelas dan ID	38
3.3.5 Penurunan.....	40
3.3.6 Posisi dan Tata Letak	40
3.3.7 Responsif	41
3.3.8 Selektor Pseduo-Class dan Psedue-element....	41
3.3.9 Animasi dan Transisi	42
3.3.10 Modularitas dan Pemeliharaan.	43
3.3.11 Kompatibilitas Browser	43
3.4 Cara Menulis CSS.....	44
3.4.1 CSS Style Internal	44
3.4.2 CSS Style External.....	45
3.4.4 CSS Style Inline	47
DAFTAR PUSTAKA.....	49
BAB 4 DASAR-DASAR JAVASCRIPT	51
4.1 Pendahuluan.....	51
4.2 Apa itu JavaScript	53
4.2.1 Menjalankan JavaScript	55
4.2.2 Cara Menggunakan JavaScript di HTML	56
4.2.3 Tipe Data dalam JavaScript	57
4.2.4 Variabel dalam JavaScript.....	59
4.3 Fungsi dalam JavaScript.....	61
DAFTAR PUSTAKA.....	64
BAB 5 MEMBUAT HALAMAN RESPONSIF	65
5.1 Pendahuluan.....	65
5.2 Pengertian Halaman Responsif	66
5.3 HTML 5 dan CSS3.....	67
5.3.1 Meta tag viewport	68
5.3.2 <i>Layout</i> Fleksibel	68
5.3.3 <i>Image</i> Fleksibel	70
5.3.4 Media Query	73
DAFTAR PUSTAKA.....	79
BAB 6 PENGENALAN JQUERY	81

6.1 Pendahuluan	81
6.2 Defenisi JQuery	81
6.2.1 Ketangguhan jquery	82
6.2.2 Keistimewaan jquery	82
6.2.4 Cara Memperoleh Library jquery.....	83
6.2 Cara Kerja JQuery	83
6.3 Menggunakan Selector	85
6.3.1 Basic Selector.....	85
6.3.2 Attribute Selector.....	87
6.3.3 Filter Selector	88
6.3.4 Form Selector	88
6.3.5 Hierarchy Selector	88
6.4 Memahami Effect.....	88
6.5 Pemahaman Fungsi Event.....	90
DAFTAR PUSTAKA	92
BAB 7 INTERAKSI DENGAN SERVER MELALUI AJAX ..	93
7.1 Apa itu Ajax.....	93
7.2 Mengapa Mempelajari AJAX?.....	94
7.3 Teknologi AJAX.....	96
7.4 Web Terkenal dengan Ajax.....	96
7.4.1 Dukungan Peramban.....	97
7.4.2 Langkah Operasi AJAX.....	97
7.4.3 AJAX - XMLHttpRequest	98
7.5 Ajax dan teknologi sisi server.....	99
7.6 Dunia AJAX.....	101
7.7 Mengubah Aplikasi HTML menjadi Aplikasi Ajax..	102
7.8 Properti XMLHttpRequest	104
7.9 Membangun Kembali Koneksi.....	105
7.10 Respons Server Web: Penanganan	109
7.11 Mengontrol Permintaan XMLHttpRequest Saat Ini.....	113
DAFTAR PUSTAKA	114
BAB 8 DASAR-DASAR BASIS DATA	115
8.1 Pendahuluan	115
8.2 Definisi Basis Data	115

8.3 Sejarah Basis Data	116
8.4 Fungsi Basis Data	117
8.5 Komponen Basis Data	119
8.6 Jenis Basis Data.....	122
8.7 Konsep Dasar Basis Data	124
8.8 Operasi Dasar Basis Data.....	126
8.9 Desain Basis Data.....	128
DAFTAR PUSTAKA.....	131
BAB 9 PENGENALAN PHP	133
9.1 Pendahuluan.....	133
9.2 Instalasi	137
9.3 Dasar PHP	138
9.3.1 Struktur Dasar	138
9.3.2 Tipe Data.....	140
9.3.3 Variabel	141
9.3.4 Operator.....	142
9.4 Struktur Kontrol.....	144
9.4.1 Kondisi.....	145
9.4.2 Perulangan	147
9.5 Fungsi.....	149
9.5.1 Deklarasi Fungsi	150
9.5.2 Pemanggilan Fungsi	150
DAFTAR PUSTAKA.....	152
BAB 10 PENGELOLAAN FORMULIR.....	155
10.1 Definisi	155
10.2 Jenis-Jenis Formulir.....	155
10.3 Cara Kerja Formulir	157
10.4 Metode Pengiriman Data Formulir	159
10.5 Membuat Formulir dengan HTML.....	164
10.5.1 Elemen <i>Form</i>	164
10.5.2 Elemen Input	165
10.6 Pengelolaan Form Dengan <i>Database</i>	166
10.7 Studi Kasus Pembuatan Formulir	167
DAFTAR PUSTAKA.....	171

BAB 11 KEAMANAN DAN PERLINDUNGAN

TERHADAP SERANGAN.....173

11.1 Pengantar Keamanan Web.....	173
11.1.1 Pentingnya Keamanan Dalam Pengembangan Web	173
11.1.2 Ancaman Umum Terhadap Situs Web Dan Aplikasi	174
11.1.3 Konsep Dasar Keamanan Web	175
11.2 Serangan Web Umum.....	176
11.2.1 <i>Cross-Site Scripting</i> (XSS)	176
11.2.2 <i>Cross-Site Request Forgery</i> (CSRF)	177
11.2.3 Injection Attacks (SQL, Command, dll.)	178
11.2.4 Session Hijacking dan Session Fixation	179
11.2.5 Brute-Force Attacks	180
11.2.6 <i>Man-in-the-Middle</i> (MitM) Attacks.....	180
11.2.7 Denial of Service Attacks	181
11.3 Perlindungan Terhadap Serangan XSS.....	182
11.3.1 Menyaring Input Pengguna	182
11.3.2 Penggunaan Fungsi Sanitasi Data	183
11.3.3 Menggunakan Header HTTP Yang Tepat (<i>Content Security Policy</i>).....	184
11.3.4 Prinsip-Prinsip Pengkodean Yang Aman	187
11.4 Melindungi Aplikasi dari Serangan CSRF	188
11.4.1 Penggunaan Token CSRF.....	188
11.4.2 Menggunakan Header HTTP yang tepat (Referer, Origin)	191
11.4.3 Verifikasi Permintaan Sisi Server	192
11.5 Keamanan Basis Data dan Serangan Injection	193
11.5.1 Menggunakan Prepared Statements atau ORM.....	193
11.5.2 Validasi Data Pengguna.....	194
11.5.3 Mengenkripsi Data Sensitif.....	194
11.5.4 Konfigurasi Yang Aman Untuk Basis Data	195
11.6 Keamanan Sesi dan Otentikasi	196

11.6.1 Penggunaan Token Sesi Yang Aman.....	196
11.6.2 Melakukan Otentikasi Yang Kuat.....	197
11.6.3 Menggunakan Metode Hashing Yang Aman..	198
11.6.4 Manajemen Token Akses	199
11.7 Praktek Pengembangan Aplikasi yang Aman	200
11.7.1 Menggunakan Framework Yang Aman	200
11.7.2 Menerapkan Pembaruan Keamanan Secara Teratur.....	201
11.7.3 Memeriksa Kerentanan Terkait Keamanan ..	202
11.7.4 Mengelola Akses Dan Izin Dengan Bijak.....	203
11.8 Melakukan Pengujian Keamanan.....	203
11.8.1 Metodologi Pengujian Keamanan.....	203
11.8.2 Tools dan Teknik Pengujian Yang Umum	205
11.8.3 Melakukan Pengujian Penetrasi.....	206
11.8.4 Pemantauan Keamanan Dan Perbaikan Kerentanan	207
DAFTAR PUSTAKA.....	208
BAB 12 TREND DAN PERKEMBANGAN TERKINI	
DALAM PEMROGRAMAN WEB	211
12.1 Pendahuluan	211
12.2 Sejarah Pemrograman Website.....	214
12.3 Kebutuhan Pasar Terhadap Pengembangan Website	216
12.4 Tujuh Frameworks Populer	219
12.5 Contoh Penggunaan Framework Populer dalam Studi Kasus Pembuatan E-Commerce.....	225
12.6 Bahasa Pemrograman untuk Web Based.....	229
12.7 Tips Menjadi Programmer Website	236
DAFTAR PUSTAKA.....	239
BIODATA PENULIS	

DAFTAR GAMBAR

Gambar 1.1.	Halaman Web Pertama	1
Gambar 1.2.	Halaman Web Facebook Tahun 2004 yang telah dengan CSS dan Javascript (Kiri) dan Halaman Web Facebook Tahun 2020 yang telah dilengkapi dengan Responsive Design (Kanan)	3
Gambar 1.3.	Perkembangan Situs Web Dari Waktu ke Waktu.....	4
Gambar 1.4.	Proses Interaksi Antara Klien dan Server Saat Pengguna Mengakses Sebuah Situs Web.....	5
Gambar 1.5.	Empat Lapisan pada Aplikasi berbasis Web.....	8
Gambar 1.6.	Arsitektur <i>Client-server</i>	10
Gambar 1.7.	Arsitektur <i>Single Page Application</i>	11
Gambar 1.8.	Arsitektur <i>Microservice</i>	12
Gambar 1.9.	Arsitektur <i>Serverless</i>	13
Gambar 1.10.	<i>Progresive Web Application</i> (PWA)	14
Gambar 1.11.	Tiga Pilar Halaman Web	14
Gambar 2.1.	Posisi HTML dalam bagan Bahasa Markup/Pemrograman yang populer saat ini	22
Gambar 2.2.	Penjelasan tag dalam HTML	25
Gambar 2.3.	Tampilan penggunaan tag teks.....	27
Gambar 2.4.	Tampilan penggunaan tag daftar	28
Gambar 2.5.	Penggunaan atribut dalam tag anchor.....	29
Gambar 2.6.	Tampilan penggunaan atribut style.....	30
Gambar 3.1.	Alur Kerja Desain Web.....	33
Gambar 3.2.	Contoh Koding CSS	34
Gambar 3.3.	Anatomi CSS	37
Gambar 3.4.	Contoh penggunaan selektor.....	37

Gambar 3.5. Contoh Penggunaan nilai pada style variasi warna.....	38
Gambar 3.6. Contoh Penggunaan kelas dan id	39
Gambar 3.7. Contoh Sintaks kelas dan id css	39
Gambar 3.8. Contoh Sintaks penurunan css	40
Gambar 3.9. Contoh Sintaks penurunan css	41
Gambar 3.10. Contoh Sintaks Kode CSS Internal Style...	44
Gambar 3.12. Hasil Contoh Sintaks Kode CSS Eksternal Style	46
Gambar 3.13. Hasil Contoh Sintaks Kode CSS Eksternal Style	47
Gambar 3.14. Contoh Sintaks Kode CSS Inline Style	47
Gambar 3.15. Hasil Contoh Sintaks Kode CSS Inline Style.....	48
Gambar 4.1. Brendan Eich, Pencipta JavaScript.....	52
Gambar 4.2. Cara Kerja JavaScript	53
Gambar 5.1. Contoh tampilan halaman responsif.....	67
Gambar 5.2. Contoh rancangan <i>fluid layout</i>	69
Gambar 5.3. Contoh tampilan <i>image Responsive</i>	71
Gambar 5.4. Contoh tampilan <i>image Responsive</i>	72
Gambar 5.5. Contoh media query bernilai <i>false</i>	75
Gambar 5.6. Contoh media query bernilai <i>true</i>	75
Gambar 6.1. <i>Script</i> jQuery.....	84
Gambar 6.2. Hasil Program	85
Gambar 6.3. <i>Syntax</i> Untuk Selector <i>Class</i>	85
Gambar 6.4. Hasil Program Dari Selector <i>Class</i>	87
Gambar 6.6. Hasil Program	87
Gambar 6.7. Hasil program dari event click	91
Gambar 7.1. Permintaan klien untuk halaman PHP	99
Gambar 7.2. HTTP, HTML, ASP.NET, dan JavaScript beraksi	100
Gambar 7.3. Contoh panggilan AJAX	101
Gambar 7.4. Menjalankan aplikasi AJAX Joke of the Day.....	112

Gambar 9.1. Kompilasi penggunaan Server-side Scripting.....	135
Gambar 9.2. Konsep Kerja PHP	137
Gambar 9.3. Operator Aritmatika	142
Gambar 9.4. Operator Penugasan	143
Gambar 9.5. Operator Penugasan	143
Gambar 9.6. Operator Logika.....	144
Gambar 10.1. Cara Kerja Pengelolaan Formulir.....	158
Gambar 10.2. Method GET Pada Form.....	160
Gambar 10.3. Method POST Pada Form	160
Gambar 10.4. Struktur Komponen Pengiriman Data Form	161
Gambar 10.5. Tampilan form.html	163
Gambar 10.6. Tampilan tampilkan.php	164
Gambar 10.7. Tampilan input.html	166
Gambar 10.8. Alur Pemrosesan Form Dengan Basis Data	167
Gambar 12.1. Ilustrasi Tren Pemrograman Web.....	211
Gambar 12.2. Sejarah Perkembangan Web.....	214
Gambar 12.3. Ilustrasi Kebutuhan Pasar	216
Gambar 12.4. Framework Populer	219
Gambar 12.5. Ilustrasi E-Commerce	225
Gambar 12.6. Contoh Script HTML.....	229
Gambar 12.7. Contoh Script CSS.....	230
Gambar 12.8. Contoh Script JavaScript.....	231
Gambar 12.9. Contoh Script PHP.....	232
Gambar 12.10. Contoh Script Python	233
Gambar 12.11. Contoh Script Ruby	234
Gambar 12.12. Contoh Script Java.....	235
Gambar 12.13. Sistem Organisasi Kehidupan makhluk Hidup	236

DAFTAR TABEL

Tabel 5.1. Satuan ukur relatif	70
Tabel 5.2. Resolusi dan ukuran monitor dan smartphone	77
Tabel 6.1. Perintah Dalam Selector	85
Tabel 7.1. Atribut Objek XMLHTTPREQUEST	104
Tabel 7.2. Parameter metode OPEN() objek XMLHTTPREQUEST	106
Tabel 7.3. XMLHTTPREQUEST Nilai atribut status objek	108
Tabel 10.1. Jenis-Jenis Kontrol Formulir	155

BAB 1

PENGANTAR PEMROGRAMAN WEB

Oleh Ade Maulana

1.1 Sejarah dan Evolusi Web

Web dimulai pada tahun 1990-an dengan pengenalan HTML (*Hypertext Markup Language*) sebagai bahasa markah utama untuk membangun halaman-halaman web. Pada tahun 1993, World Wide Web mulai populer. Awalnya hanya ada lima puluh server Web di seluruh dunia, tetapi pada bulan Oktober jumlahnya meningkat menjadi lima ratus. Penyebabnya adalah karena Gopher, protokol yang digunakan sebagai alternatif utama untuk mengakses informasi di Internet pada saat itu. (Isaacson, 2014) Pada saat itu, web masih sangat sederhana dengan tampilan teks dan tautan antara halaman-halaman (Gambar 1.1).

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypertext](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), [November's W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)
Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)
on the browser you are using

[Software Products](#)
A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Xrdb](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Librery](#))

[Technical](#)
Details of protocols, formats, program internals etc

[Bibliography](#)
Paper documentation on W3 and references.

[People](#)
A list of some people involved in the project.

[History](#)
A summary of the history of the project.

[How can I help?](#)
If you would like to support the web.

[Getting code](#)
Getting the code by [anonymous FTP](#), etc.

Gambar 1.1. Halaman Web Pertama
(Sumber : Shontell, 2011)

Namun, seiring berjalannya waktu, web mengalami evolusi yang cepat. Pengenalan CSS (*Cascading Style Sheets*) memungkinkan pengembang web untuk mengatur tampilan dan gaya halaman secara terpisah dari struktur kontennya. Ini membawa perubahan besar dalam desain dan estetika web. Kemudian, JavaScript menjadi bahasa pemrograman klien yang populer untuk menambahkan interaktivitas dan fungsionalitas dinamis pada halaman web. Ini membuka kesempatan bagi pengembang web untuk membuat efek-efek animasi, validasi form, dan berbagai fitur interaktif lainnya.

Perkembangan teknologi web juga melahirkan pendekatan dan framework pengembangan web yang baru. Pengenalan AJAX (*Asynchronous JavaScript and XML*) memungkinkan komunikasi asinkron antara browser dan server, membawa pengalaman pengguna yang lebih responsif dan interaktif.

Selanjutnya, munculnya HTML5 membawa fitur-fitur baru seperti audio dan video yang terintegrasi, API geolokasi, penyimpanan lokal, dan banyak lagi. Ini membuka peluang baru bagi pengembang web untuk menciptakan pengalaman web yang lebih kaya dan lebih serupa dengan aplikasi.

Tren terbaru dalam pemrograman web termasuk pengembangan responsif untuk menyesuaikan tampilan halaman dengan berbagai perangkat dan layar yang berbeda. Gambar 1.2 memvisualisasikan evolusi Halaman Web Facebook dari tahun 2004 hingga 2020 dengan penggunaan CSS, JavaScript, dan Responsive Design. Pada tahun 2004, halaman web Facebook menampilkan desain sederhana dan berfokus pada fungsi. Namun, seiring perkembangan teknologi khususnya web pada tahun 2020, halaman tersebut telah mengadopsi Responsive Design, memungkinkan tampilan yang adaptif pada berbagai perangkat dengan penggunaan CSS dan JavaScript yang lebih canggih. Perubahan ini mencerminkan

pentingnya pengalaman pengguna yang dinamis dan interaktif serta mengikuti perkembangan teknologi dalam pengembangan web modern.



Gambar 1.2. Halaman Web Facebook Tahun 2004 yang telah dengan CSS dan Javascript (Kiri) dan Halaman Web Facebook Tahun 2020 yang telah dilengkapi dengan Responsive Design (Kanan) (Sumber : Web Design Museum)

Evolusi web (Gambar 1.3) telah mengubah lanskap digital dengan luar biasa sejak awalnya diperkenalkan. Dimulai dari Web 1 yang menampilkan halaman statis, kemudian muncul Web 2 yang menghadirkan interaktivitas dan partisipasi pengguna melalui jejaring sosial, blog, dan layanan berbasis *cloud*. Perkembangan ini memungkinkan pengguna untuk berbagi konten, terlibat dalam komunitas *online*, dan menciptakan pengalaman pengguna yang lebih dinamis. Selanjutnya, munculnya Web 3 membawa konsep semantik yang memperkaya pemahaman mesin terhadap data web, memungkinkan personalisasi yang lebih baik dan integrasi antara aplikasi. Saat ini, dunia digital sedang mencari perubahan dengan munculnya Web 4 yang bertujuan untuk menghadirkan desentralisasi melalui teknologi blockchain dan kontrak cerdas.



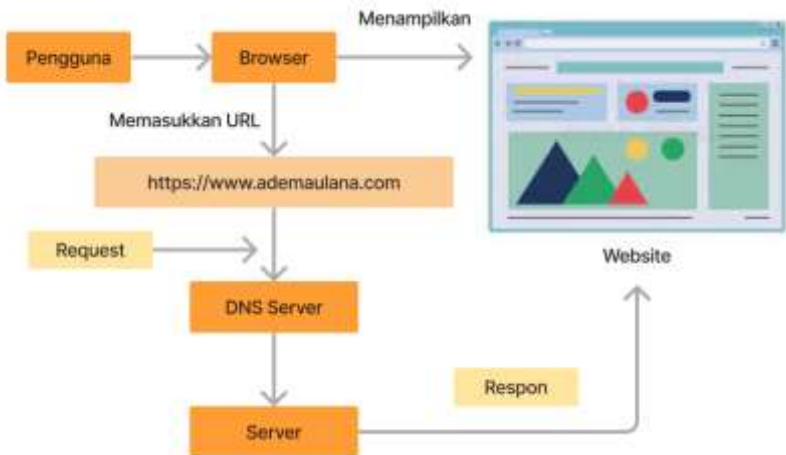
Gambar 1.3. Perkembangan Situs Web Dari Waktu ke Waktu
Sumber : (Madurai, 2018)

Perubahan dalam pengembangan web tidak hanya terbatas pada antarmuka pengguna dan arsitektur *front-end*, tetapi juga melibatkan perubahan dalam pengembangan *back-end*. *Back-end* merupakan bagian dari sistem web yang bertanggung jawab untuk memproses logika bisnis, mengelola basis data, dan berkomunikasi dengan server. Perkembangan dalam pengembangan *back-end* melibatkan penggunaan bahasa pemrograman dan kerangka kerja yang lebih canggih dan efisien, seperti Node.js, Django, Ruby on Rails, atau Laravel, yang memungkinkan pengembang untuk membangun aplikasi web yang lebih kuat dan *scalable*. Selain itu, terdapat juga perubahan dalam penggunaan teknologi basis data, seperti adopsi basis data NoSQL seperti MongoDB dan Redis, yang memberikan fleksibilitas dan kinerja yang lebih baik untuk aplikasi web yang kompleks. Perubahan dalam pengembangan *back-end* bertujuan untuk meningkatkan kehandalan, kecepatan, dan skalabilitas aplikasi web, serta memastikan

integrasi yang baik antara *front-end* dan *back-end* untuk memberikan pengalaman pengguna yang optimal.

1.2 Cara Kerja Web

Saat seorang pengguna mengakses sebuah situs web, peran klien dan server saling berinteraksi (Gambar 1.4). Klien, yang biasanya adalah browser web yang dijalankan di perangkat pengguna, membuat permintaan (*request*) kepada server untuk mengambil halaman web atau sumber daya lainnya. Server, di sisi lain, bertugas menyediakan respons (*response*) atas permintaan tersebut.



Gambar 1.4. Proses Interaksi Antara Klien dan Server Saat Pengguna Mengakses Sebuah Situs Web.

Protokol HTTP (*Hypertext Transfer Protocol*) digunakan sebagai dasar komunikasi antara klien dan server dalam web. Klien membuat permintaan dengan menggunakan metode HTTP seperti GET (untuk mengambil data), POST (untuk mengirimkan data), atau metode lainnya. Permintaan ini

mengandung informasi seperti URL, header, dan mungkin juga data yang dikirimkan.

Server menerima permintaan dan memprosesnya, dengan melibatkan pembacaan informasi dari permintaan, pengambilan sumber daya yang diminta, dan persiapan respons. Respons ini berisi kode status HTTP (misalnya, 200 OK atau 404 Not Found), header yang mengandung informasi tambahan, dan konten yang diminta seperti HTML, gambar, atau data lainnya.

Selain HTTP, ada juga protokol HTTPS (*HTTP Secure*) yang menggunakan lapisan enkripsi SSL/TLS untuk mengamankan komunikasi antara klien dan server (Muttaqin, et al, Pengantar Internet, 2023). Ini digunakan untuk menjaga keamanan dan privasi data yang ditransmisikan melalui web, terutama ketika terdapat pertukaran informasi sensitif seperti kata sandi atau data pribadi.

Pemahaman tentang cara kerja web, termasuk peran klien dan server, serta protokol HTTP dan HTTPS, penting bagi pengembang web. Ini memungkinkan pengembang web untuk memahami bagaimana komunikasi terjadi di web, bagaimana data dikirimkan dan diterima, serta memastikan keamanan dan keandalan dalam pengembangan situs web.

1.3 Bahasa Pemrograman

Dalam mengembangkan sebuah situs web, bahasa pemrograman umum diperkenalkan yang digunakan dalam pengembangan situs web. Bahasa-bahasa ini meliputi HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheets*), dan JavaScript.

HTML adalah bahasa markah yang digunakan untuk membangun struktur halaman web. Dengan menggunakan elemen-elemen HTML seperti tag, atribut, dan nilai-nilai, pengembang web dapat menentukan bagaimana elemen-

elemen halaman, seperti teks, gambar, tautan, dan formulir, terstruktur dan diatur.

CSS adalah bahasa gaya yang digunakan untuk mengatur tampilan dan penataan elemen pada halaman web. Dengan menggunakan aturan CSS, pengembang web dapat mengubah warna, ukuran, jenis font, tata letak, dan efek visual lainnya pada elemen-elemen halaman web. Ini memungkinkan pengembang untuk membuat halaman web yang menarik dan seragam secara visual.

JavaScript adalah bahasa pemrograman yang lebih canggih yang digunakan untuk memberikan interaktivitas dan fungsionalitas dinamis pada halaman web. Dengan menggunakan JavaScript, pengembang dapat membuat efek animasi, validasi formulir, memanipulasi dan memperbarui elemen halaman secara dinamis, serta berkomunikasi dengan server melalui permintaan asinkron.

Selain ketiga bahasa tersebut, ada juga bahasa pemrograman web lainnya seperti PHP, Python, Ruby, dan banyak lagi yang umum digunakan untuk pengembangan *back-end* atau pengembangan fitur dinamis yang berjalan di sisi server.

Memahami dan menguasai bahasa-bahasa pemrograman web ini merupakan fondasi penting dalam pengembangan situs web. Dengan menggabungkan HTML, CSS, dan JavaScript secara efektif, pengembang web dapat membuat halaman web yang menarik, interaktif, dan fungsional.

1.4 Arsitektur Web

Arsitektur aplikasi web menjelaskan bagaimana komponen-komponen aplikasi saling terhubung dan berkomunikasi. Arsitektur web juga mencakup hubungan antara klien dan server yang menentukan koneksi serta pengaturan komunikasi antara keduanya. Meskipun berbagai

ukuran dan kompleksitas aplikasi web mengikuti struktur arsitektur yang sama, setiap aplikasi memiliki detail yang unik. Proses permintaan-respons dan komponen yang terlibat dalamnya, serta siklus permintaan-respons sederhana, merupakan bagian penting dalam memahami arsitektur aplikasi web.

Sebuah kerangka kerja aplikasi berbasis web umumnya terdiri dari lapisan presentasi, bisnis, persistensi, dan database. Aplikasi kecil biasanya memiliki lapisan gabungan bisnis dan persistensi, sedangkan aplikasi yang lebih besar dapat memiliki empat hingga enam lapisan yang berfungsi untuk memisahkan dan mengatur komponen aplikasi secara terstruktur.



Gambar 1.5. Empat Lapisan pada Aplikasi berbasis Web
Sumber : (InterviewBit, 2022)

Gambar 1.5 menampilkan gambar dari empat buah lapisan yang ada pada aplikasi berbasis web, antara lain :

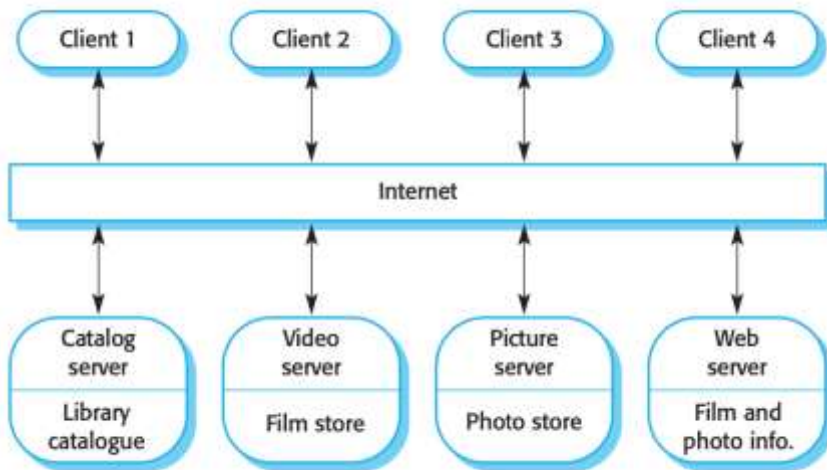
1. Lapisan presentasi terdiri dari HTML, CSS, JavaScript, dan kerangka kerjanya, yang menyediakan template HTML untuk lapisan presentasi dan memungkinkan interaksi antara antarmuka dan browser.
2. Logika bisnis dan aturan berada di lapisan bisnis. Permintaan dari browser diproses di sini, diikuti oleh

- eksekusi instruksi bisnis yang ditentukan dalam permintaan tersebut, lalu diteruskan ke lapisan presentasi.
3. Lapisan persistensi data merupakan bagian dari lapisan persistensi dan juga dikenal sebagai Lapisan Akses Data. Ini terhubung ke lapisan bisnis dan, selain pengambilan data, juga mengelola penyimpanan data. Data diambil dari server database oleh lapisan persistensi data.
 4. Logika bisnis terhubung ke sisi klien melalui lapisan database, yang melindungi integritas data dengan memisahkannya dari bisnis.

Dalam pengembangan web, terdapat berbagai macam arsitektur aplikasi yang umum digunakan. Setiap arsitektur memiliki pendekatan dan prinsip yang berbeda dalam membangun dan mengorganisir aplikasi web. Pemilihan arsitektur yang tepat sangat penting untuk mencapai tujuan yang diinginkan, seperti skalabilitas, keterpisahan tugas, dan kemudahan pemeliharaan. Dengan pemahaman yang baik tentang arsitektur aplikasi web, pengembang dapat membuat keputusan yang tepat dalam merancang dan membangun aplikasi yang efisien dan handal.

1.4.1 Arsitektur *Client-Server*

Arsitektur *client-server* adalah model arsitektur yang umum digunakan di web. Pada model arsitektur ini, aplikasi terdiri dari dua bagian utama: klien dan server. Klien berinteraksi dengan pengguna dan menangani tampilan serta logika pengguna. Server, di sisi lain, mengelola logika bisnis dan menyimpan dan memproses data. Komunikasi antara klien dan server terjadi melalui protokol HTTP.

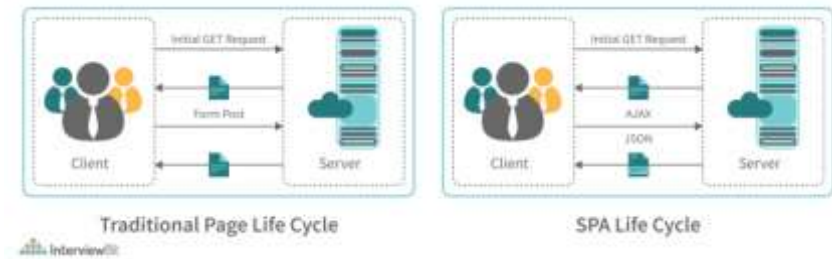


Gambar 1.6. Arsitektur *Client-server*

Sumber : (Sommerville, 2021)

1.4.2 Arsitektur Single Page Application

SPA (*Single Page Applications*) bertujuan untuk mengatasi kendala umum dalam membangun aplikasi yang mulus, dengan tujuan memberikan pengalaman pengguna yang mudah digunakan dan intuitif. Sebagai alternatif dari memuat halaman baru, SPA memuat satu halaman web dan secara dinamis memperbarui kontennya untuk mencerminkan perubahan, menghilangkan kebutuhan akan pembaruan halaman. Logika pada *front-end* sejalan dengan *back-end*, dan framework JavaScript di sisi klien digunakan untuk mengembangkan SPA. Dengan pendekatan ini, SPA menawarkan pengalaman pengguna yang lebih lancar dan interaktif dibandingkan dengan aplikasi web tradisional.



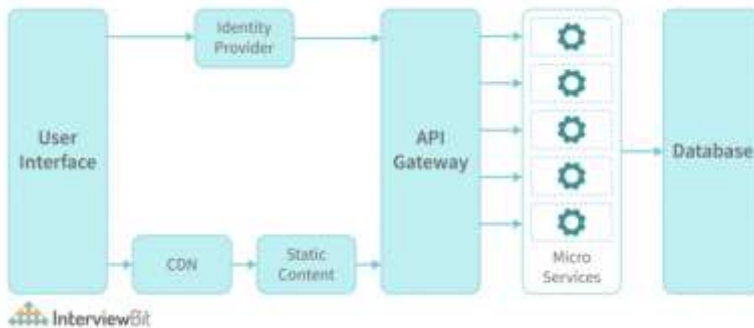
Gambar 1.7. Arsitektur *Single Page Application*
Sumber : (InterviewBit, 2022)

1.4.3 Arsitektur *Microservice*

Arsitektur monolitik kini telah digantikan oleh arsitektur mikro layanan atau lebih dikenal dengan *microservice* yang memanfaatkan sejumlah layanan untuk bekerja secara independen dan menyelesaikan masalah yang kompleks. Komunikasi antar layanan dalam arsitektur ini menggunakan API, sehingga hubungan antar layanan menjadi longgar.

Penerapan aplikasi web menjadi lebih mudah berkat penggunaan arsitektur mikro layanan yang tidak memiliki komponen yang saling terikat. Dalam arsitektur *microservices*, kesulitan dalam penerapan komponen layanan secara monolitik dapat dihindari. Pendekatan ini juga mengeliminasi kebutuhan untuk menerapkan banyak komponen layanan dan mengatasi kesulitan dalam penerapan aplikasi monolitik.

Beberapa perusahaan teknologi terkemuka seperti Amazon, Netflix, SoundCloud, Comcast, dan eBay merupakan contoh dari penggunaan arsitektur mikro layanan dalam pengembangan aplikasi mereka. Gambaran umum mengenai penerapan dari arsitektur *microservice* dapat dilihat pada Gambar 1.8.



Gambar 1.8. Arsitektur *Microservice*

Sumber : (InterviewBit, 2022)

1.4.4 Arsitektur *Serverless*

Penyedia layanan cloud seperti Amazon dan Microsoft mengelola arsitektur *serverless*, sehingga tidak diperlukan intervensi manual pada server. Tidak peduli seberapa kompleks kode Anda, penyedia layanan cloud dapat menangani semuanya untuk Anda - Anda tidak perlu mendeploynya secara manual pada server Anda. Arsitektur *serverless* adalah pola desain di mana aplikasi dibangun dan dijalankan tanpa intervensi manusia pada server yang dikelola oleh penyedia layanan cloud pihak ketiga seperti Amazon dan Microsoft. Hal ini memungkinkan pengembang fokus pada kualitas produk dan kesulitan membuatnya sangat *scalable* dan *reliable*. *Backend-as-a-Service* (BaaS) dan *Function-as-a-Service* (FaaS) adalah dua jenis layanan *serverless*.

Pengembang yang menggunakan BaaS untuk menghilangkan operasi *back-end*, sehingga dapat fokus pada membangun aspek *front-end* aplikasi mereka tanpa khawatir tentang mengoperasikan *back-end*. Contohnya adalah Amazon Amplify yang populer. Di sisi lain, FaaS adalah model yang didorong oleh peristiwa (*event-driven*) yang memungkinkan pengembang fokus pada penulisan kode dan *event-driven*.

Tugas-tugas yang tersisa akan ditangani oleh penyedia layanan FaaS seperti Amazon Lambda dan Microsoft Azure.



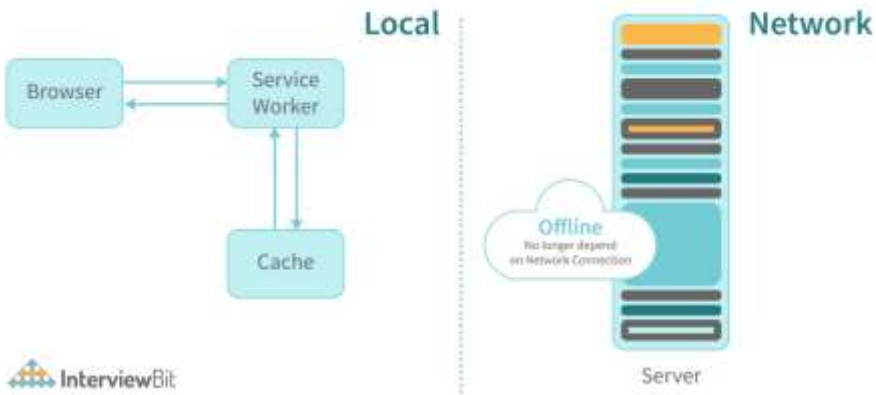
Gambar 1.9. Arsitektur *Serverless*

Sumber : (InterviewBit, 2022)

1.4.5 Progressive Web Application

Pada 2015, Google memperkenalkan tahun *Progressive Web Apps* (PWA) sebagai solusi untuk mengembangkan aplikasi yang memberikan fungsionalitas mirip aplikasi native, dengan keandalan yang ditingkatkan, dan proses instalasi yang mudah. PWA dirancang untuk berfungsi pada semua browser dan perangkat, secara otomatis menyesuaikan diri dengan berbagai ukuran layar, termasuk tablet dan desktop. Salah satu keunggulan utama PWA adalah kemampuannya untuk bekerja dalam mode offline dan tetap berperforma baik bahkan dengan koneksi internet yang buruk.

Perusahaan-perusahaan terkemuka seperti Uber, Aliexpress, Alibaba, Pinterest, dan Starbucks telah mengadopsi pendekatan PWA dalam pengembangan produk mereka, menggunakan situs web PWA untuk memberikan pengalaman pengguna yang kaya dan menarik bagi pelanggan mereka.



Gambar 1.10. *Progressive Web Application (PWA)*
Sumber : (InterviewBit, 2022)

1.5 Struktur Dasar Halaman Web

Struktur Dasar Halaman Web merujuk pada komponen-komponen esensial yang membentuk kerangka dasar sebuah halaman web. Dalam pengembangan sebuah halaman web, ada tiga pilar teknologi yakni HTML, CSS, dan Javascript (Gambar 1.11)



Gambar 1.11. Tiga Pilar Halaman Web

Pilar pertama, *Structure and Content*, berkaitan dengan bagaimana informasi dan data disusun dan diorganisir di dalam sebuah situs web. Ini melibatkan penggunaan bahasa markup seperti HTML (*Hypertext Markup Language*) untuk membangun struktur dasar halaman web dan menentukan elemen-elemen konten seperti teks, gambar, video, dan tautan. Elemen-elemen ini memberikan kerangka dasar untuk menampilkan konten, memasukkan gambar, membuat hyperlink, dan memformat teks di halaman web (Duckett, 2014). Struktur dan konten yang baik memastikan informasi dapat diakses dengan mudah dan dipahami oleh pengguna dan mesin pencari.

Pilar kedua, *Presentation*, berkaitan dengan cara tampilan visual situs web ditentukan dan didefinisikan. Ini melibatkan penggunaan bahasa markup seperti CSS (*Cascading Style Sheets*) untuk mengatur tata letak, warna, jenis huruf, dan elemen desain lainnya. *Presentation* memungkinkan pengembang untuk menciptakan pengalaman visual yang menarik dan konsisten di seluruh situs web. Hal ini juga memungkinkan responsivitas, yang memungkinkan situs web untuk menyesuaikan tampilan dengan baik pada berbagai perangkat dan ukuran layar.

Pilar ketiga, *Behavior*, melibatkan aspek interaktivitas dan responsifitas dari sebuah situs web. Ini melibatkan penggunaan bahasa pemrograman seperti JavaScript untuk mengendalikan perilaku dan interaksi situs web dengan pengguna. Dengan *Behavior*, pengembang dapat membuat efek animasi, validasi formulir, interaksi dengan pengguna, pemrosesan data, dan masih banyak lagi. Perilaku yang baik memastikan situs web responsif dan memberikan pengalaman pengguna yang interaktif dan dinamis.

Dengan ketiga pilar ini bekerja bersama-sama, pengembang dapat membangun situs web yang baik struktur dan kontennya, menarik secara visual dalam presentasinya, serta interaktif dan responsif dalam perilakunya. Dengan

memahami pilar pada halaman web, akan dapat membangun halaman web yang terstruktur dengan baik dan terorganisir. Hal ini penting untuk menghasilkan pengalaman pengguna yang baik dan memudahkan pengembangan lebih lanjut. Dengan menggunakan HTML, CSS, dan JavaScript secara efektif, pengembang dapat menciptakan halaman web yang menarik, fungsional, dan responsif bagi pengguna.

1.6 Pengembangan Back-end dan Front-end

Dalam pengembangan aplikasi berbasis web dikenal dua komponen utama yakni *front-end* dan *back-end*. Pengembangan front-end berkaitan dengan mengatur tampilan dan interaksi di sisi klien, sementara pengembangan back-end berkaitan dengan pengelolaan server dan logika di sisi server.

Pengembangan *front-end* melibatkan pembuatan antarmuka pengguna yang menarik dan responsif. Pengembang front-end menggunakan bahasa pemrograman seperti HTML, CSS, dan JavaScript untuk membangun struktur halaman, mengatur tata letak, dan memberikan fungsionalitas interaktif. Mereka bertanggung jawab untuk menciptakan pengalaman pengguna yang baik dengan memperhatikan desain visual, penggunaan yang mudah, dan kinerja yang optimal di berbagai perangkat dan browser.

Di sisi lain, pengembangan *back-end* berkaitan dengan pengelolaan server dan logika di sisi server. Pengembang *back-end* menggunakan bahasa pemrograman seperti PHP, Python, Ruby, atau JavaScript (dalam konteks server-side, seperti Node.js) untuk mengelola basis data, memproses permintaan pengguna, dan menghasilkan respons yang tepat. Mereka bertanggung jawab atas aspek keamanan, keandalan, dan skala aplikasi web, serta mengoptimalkan kinerja server dan pengelolaan data.

Kerja sama antara pengembangan *front-end* dan *back-end* sangat penting dalam membangun situs web yang lengkap. Pengembang *front-end* menyediakan tampilan dan interaksi yang menarik bagi pengguna, sedangkan pengembang *back-end* mengelola logika dan data di balik layar. Mereka harus bekerja secara sinergis untuk menciptakan pengalaman pengguna yang mulus dan aplikasi web yang berfungsi dengan baik.

Memahami perbedaan antara pengembangan *front-end* dan *back-end* merupakan langkah penting dalam menjadi pengembang web yang kompeten. Ini membantu mengarahkan fokus dan pemahaman dalam mempelajari teknologi dan alat yang relevan dengan setiap bidang pengembangan, serta memungkinkan kolaborasi efektif dalam tim pengembangan web.

1.7 Web UI/UX Design

Web UI/UX Design adalah disiplin yang berfokus pada perancangan antarmuka pengguna (UI) dan pengalaman pengguna (UX) dalam konteks situs web. UI Design (Desain Antarmuka Pengguna) melibatkan pengaturan tampilan visual dan elemen-elemen grafis dalam situs web. Ini mencakup pemilihan warna, jenis font, ikon, gambar, dan elemen-elemen lain yang menciptakan identitas visual situs. Tujuannya adalah membuat antarmuka yang menarik, konsisten, dan intuitif sehingga pengguna dapat dengan mudah berinteraksi dengan situs web.

Sementara itu, UX Design (Desain Pengalaman Pengguna) berfokus pada bagaimana pengguna berinteraksi dengan situs web dan mencapai tujuan mereka dengan cara yang efisien dan memuaskan. Proses UX Design melibatkan pemahaman mendalam tentang pengguna dan tujuan mereka, serta perancangan alur navigasi, tata letak halaman, dan

elemen interaktif yang memastikan pengalaman yang positif dan memuaskan.

Web UI/UX Design berupaya menciptakan antarmuka pengguna yang menarik secara visual, intuitif, dan mudah digunakan. Dalam bab ini, pembaca akan mempelajari prinsip-prinsip desain UI/UX, seperti *hierarchy* (hierarki), *affordance* (kesesuaian), *usability* (kegunaan), dan *readability* (keterbacaan). Pembaca juga akan diperkenalkan dengan alat dan teknik desain yang umum digunakan dalam industri, seperti Adobe XD, Sketch, Figma, dan prototyping.

Memahami Web UI/UX Design merupakan hal yang penting bagi pengembang web, karena desain yang baik dapat meningkatkan kualitas dan kepuasan pengguna dalam berinteraksi dengan situs web. Dengan menerapkan prinsip-prinsip desain UI/UX, pengembang dapat menciptakan antarmuka yang menarik, mudah digunakan, dan menghadirkan pengalaman yang memuaskan bagi pengguna situs web.

1.8 Situs Web dan Aplikasi Berbasis Web

Situs web dan aplikasi berbasis web adalah dua konsep yang berbeda dalam dunia pengembangan web. Sebuah situs web adalah kumpulan halaman-halaman yang terhubung dan diakses melalui Internet. Situs web biasanya digunakan untuk menyajikan informasi secara online, seperti konten berita, informasi perusahaan, blog, dan sebagainya. Situs web umumnya berfokus pada penyampaian konten kepada pengguna dengan tampilan yang responsif dan navigasi yang mudah. Mereka cenderung lebih statis dalam sifatnya, dengan interaksi terbatas dan tergantung pada pengguna untuk mengambil tindakan lebih lanjut (Muttaqin, *et al*, 2023).

Di sisi lain, aplikasi berbasis web adalah aplikasi yang diakses melalui browser web. Mereka memiliki fitur dan

fungsionalitas yang lebih kompleks dibandingkan dengan situs web. Aplikasi berbasis web dapat memberikan pengalaman interaktif yang mirip dengan aplikasi desktop atau mobile, termasuk kemampuan untuk menginput data, memproses informasi, dan menghasilkan respons yang dinamis. Aplikasi berbasis web biasanya melibatkan pengembangan front-end yang kompleks dengan menggunakan teknologi seperti JavaScript, HTML, dan CSS, serta pengembangan back-end untuk mengelola logika bisnis dan pemrosesan data.

Perbedaan utama antara situs web dan aplikasi berbasis web terletak pada kompleksitas dan tingkat interaktivitas yang disediakan. Situs web cenderung lebih fokus pada menyampaikan informasi secara online, sementara aplikasi berbasis web dirancang untuk memberikan fungsionalitas yang lebih kaya dan pengalaman pengguna yang lebih interaktif.

Namun, perbatasan antara situs web dan aplikasi berbasis web dapat menjadi kabur karena perkembangan teknologi web yang terus berlanjut. Beberapa situs web dapat memiliki fitur dan fungsionalitas yang lebih mirip dengan aplikasi berbasis web, dengan interaktivitas yang lebih tinggi dan kemampuan offline. Sebaliknya, aplikasi berbasis web dapat dirancang dengan tampilan dan navigasi yang sederhana sehingga terlihat seperti situs web tradisional.

Dalam pengembangan web, penting untuk memahami perbedaan dan kemiripan antara situs web dan aplikasi berbasis web agar dapat memilih pendekatan yang sesuai sesuai dengan kebutuhan proyek dan tujuan pengembangan.

DAFTAR PUSTAKA

- Duckett, J. 2014. *HTML & CSS : Design and Build Websites*. Wiley.
- InterviewBit. 2022, June 17. *Web Application Architecture – Detailed Explanation*. Retrieved from InterviewBit: <https://www.interviewbit.com/blog/web-application-architecture/>
- Isaacson, W. 2014. *The Innovators: How a Group of Hackers, Geniuses, and Geeks Created the Digital Revolution*. Simon & Schuster.
- Madurai, V. 2018, Februari 17. *Web Evolution from 1.0 to 3.0*. Retrieved from Medium: <https://medium.com/@vivekmadurai/web-evolution-from-1-0-to-3-0-e84f2c06739>
- Museum, W. D. (n.d.). *Facebook*. Retrieved from Web Design Museum: <https://www.webdesignmuseum.org/timeline/facebook-2020>
- Muttaqin, et al. 2023. *Pengantar Internet*. Medan: Yayasan Kita Menulis.
- Muttaqin, et al. 2023. *Pengantar Teknologi Digital*. Medan: Yayasan Kita Menulis.
- Shontell, A. 2011, Juni 30. *FLASHBACK: This Is What The First-Ever Website Looked Like*. Retrieved from Business Insider: <https://www.businessinsider.com/flashback-this-is-what-the-first-website-ever-looked-like-2011-6#:~:text=The%20first%20web%20page%20went,%2FWWW%2FTheProject.html>.
- Sommerville, I. 2021. *Software Engineering, 10th edition*. United Kingdom: Pearson.

BAB 2

DASAR-DASAR HTML

Oleh Rahmat Taufik R.L Bau

2.1 Pendahuluan

Sebelum HTML ditemukan, pengembangan situs web sangat berbeda dengan saat ini. Pada era sebelum HTML, situs web umumnya dibangun dengan cara yang lebih manual dan kompleks. Pengembang web harus membuat semua tampilan halaman web dari awal, termasuk mengkodekan setiap elemen secara manual menggunakan bahasa pemrograman seperti C, Perl, atau Shell Script.

Metode ini sangat rumit dan membutuhkan waktu yang lama untuk membuat halaman web yang sederhana sekalipun. Bahkan ketika para pengembang web berhasil membuat situs web, mereka mungkin tidak dapat memastikan bahwa tampilan dan fungsionalitas yang dihasilkan akan konsisten di semua perangkat dan browser yang berbeda.

Namun, dengan ditemukannya HTML oleh Sir Tim Berners-Lee, segala hal yang terkait dengan pengembangan situs web menjadi lebih mudah dan terstandarisasi. HTML memungkinkan pengembang web untuk membuat struktur situs web dan menambahkan konten dengan mudah menggunakan tag-tag HTML yang telah didefinisikan.

Dengan menggunakan HTML, pengembang web dapat dengan cepat dan mudah membuat halaman web dengan tampilan dan fungsi yang konsisten di semua perangkat dan browser yang berbeda. Seiring berkembangnya waktu, HTML terus ditingkatkan dan dikembangkan, serta menjadi dasar dari teknologi web modern yang kita kenal saat ini.

Signifikansi mempelajari HTML di era teknologi saat ini sangatlah penting. Dalam dunia yang semakin tergantung pada teknologi, pemahaman tentang HTML menjadi kunci penting bagi siapa saja yang ingin menjadi pengembang web, designer, atau bahkan pemilik bisnis online. HTML membantu pengembang web dalam menghasilkan situs web yang menarik dan interaktif dengan menggunakan tampilan dan fungsi yang terstandarisasi. Selain itu, HTML juga menjadi dasar penting untuk mempelajari bahasa pemrograman lainnya seperti CSS dan JavaScript.



Gambar 2.1. Posisi HTML dalam bagan Bahasa Markup/Pemrograman yang populer saat ini
(Sumber : badoystudio)

Dalam era digital saat ini, situs web dan aplikasi web menjadi hal yang sangat penting bagi perusahaan dan organisasi untuk memperluas jangkauan bisnis mereka dan menjangkau lebih banyak pelanggan. Dalam konteks ini, HTML menjadi kunci penting dalam pengembangan situs web yang

responsif dan mudah diakses oleh berbagai jenis perangkat, seperti desktop, tablet, dan smartphone.

Dengan demikian, pemahaman yang baik tentang HTML adalah penting bagi siapa saja yang ingin terlibat dalam pengembangan web dan menjadi bagian dari revolusi teknologi informasi. Mengetahui dan memahami dasar-dasar HTML akan membantu pengembang web untuk menghasilkan situs web yang menarik, interaktif, dan mudah digunakan oleh pengguna.

2.2 Pengenalan HTML

HTML, atau *HyperText Markup Language*, adalah bahasa *markup* standar yang digunakan untuk membuat halaman web. Sejak penemuan pertamanya pada tahun 1991, HTML telah menjadi dasar bagi pengembangan web *modern* dan menjadi salah satu teknologi terpenting dalam menghubungkan orang dan informasi di seluruh dunia.

Pada dasarnya, HTML berfungsi untuk memberikan struktur dan konten pada sebuah halaman web. Dalam bahasa *markup*, konten diberikan dalam bentuk elemen yang ditandai dengan tag. Setiap tag memiliki fungsinya sendiri dan dapat mengandung atribut yang memberikan informasi tambahan tentang elemen tersebut.

Selain itu, HTML juga memungkinkan pengguna untuk membuat tautan atau *hyperlink* antara halaman web dan membuat formulir yang dapat digunakan untuk mengumpulkan informasi dari pengguna. Selain itu, HTML juga dapat digunakan untuk menyematkan media seperti gambar, video, dan audio ke dalam halaman web.

Dalam pengembangan web modern, HTML biasanya digunakan bersama dengan CSS dan JavaScript. CSS digunakan untuk mengubah tampilan halaman web, seperti mengatur ukuran dan warna teks, menentukan layout halaman, dan lain-lain. Sedangkan JavaScript digunakan untuk menambahkan

interaktivitas pada halaman web, seperti menampilkan pesan pop-up atau membuat efek animasi.

Hypertext Markup Language merupakan standar bahasa yang di gunakan untuk menampilkan dokument web, yang bisa anda lakukan dengan HTML yaitu :

- a) Mengontrol tampilan dari *webpage* dan kontennya.
- b) Mempublikasikan dokumen secara *online* sehingga bisa di akses dari seluruh dunia.
- c) Membuat *online form* yang bisa di gunakan untuk menangani pendaftaran, transaksi secara online.
- d) Menambahkan objek-objek seperti *image, audio, video* dan juga java applet dalam dokumen HTML.

Penting untuk dipahami bahwa HTML terus berkembang dan mengalami perubahan seiring berjalannya waktu. Versi terbaru dari HTML adalah HTML5, yang menawarkan banyak fitur baru dan perbaikan dibandingkan dengan versi sebelumnya.

2.3 Pengenalan Tag HTML

Tag HTML adalah elemen dasar dari bahasa markup HTML yang digunakan untuk mengelompokkan dan menandai berbagai bagian atau elemen pada halaman web. Tag HTML digunakan untuk memberikan instruksi pada browser tentang bagaimana menampilkan konten pada halaman web, seperti teks, gambar, video, tabel, dan lain sebagainya.

Setiap tag HTML memiliki sintaks dan aturan penulisan yang khusus, dan memiliki fungsinya masing-masing dalam mengatur tampilan dan struktur halaman web. Penggunaan tag HTML yang tepat dan konsisten sangat penting untuk menghasilkan halaman web yang terstruktur, teratur, dan mudah dimengerti oleh browser dan pengguna.

Ketentuan mendasar mengenai tag HTML, selalu dimulai dengan karakter "<" dan diakhiri dengan karakter ">", contohnya :

```
<p>Belajar dasar HTML</p>
```

Dalam contoh di atas, tag "<p>" digunakan untuk menandai awal dan tag "</p>" digunakan untuk menandai akhir teks yang akan ditampilkan sebagai paragraf di halaman web.



Gambar 2.2. Penjelasan tag dalam HTML

2.3.1 Klasifikasi Tag HTML Dasar

Dalam penggunaannya, sebaiknya tag HTML digunakan secara konsisten dan sesuai dengan standar, untuk menghasilkan halaman web yang terstruktur, teratur, dan mudah dimengerti oleh browser dan pengguna. Sesuai penggunaannya, tag-tag dalam HTML dapat diklasifikasikan sebagai berikut :

1. Tag Struktur

Tag struktur HTML adalah tag yang digunakan untuk mengatur struktur atau susunan konten pada halaman web. Tag ini tidak menampilkan konten pada halaman web secara langsung, tetapi mengatur bagaimana konten ditampilkan dan diatur secara hierarkis. Beberapa tag struktur yang umum digunakan dalam HTML dasar adalah sebagai berikut:

```
<html>  
  <head>  
    <title>Dasar HTML</title>
```

```
</head>
<body>
  <p>Belajar dasar HTML</p>
</body>
</html>
```

Tag `<html>` merupakan tag pembuka untuk seluruh elemen pada halaman web, digunakan untuk mendeklarasikan bahwa *script* yang sedang dibuat menggunakan bahasa *markup* HTML. Tag `<head>` digunakan untuk menentukan informasi meta atau informasi halaman, seperti judul halaman yang menggunakan tag `<title>`. Tag `<body>` digunakan untuk menentukan konten utama pada halaman web, contohnya menampilkan tulisan Belajar dasar HTML yang diapit oleh tag paragraph, `<p>`.

2. Tag Teks

Tag teks HTML adalah tag yang digunakan untuk menampilkan dan mengelola teks pada halaman web. Tag ini memungkinkan kita untuk memberikan format, ukuran, jenis huruf, warna, dan tampilan lainnya pada teks, sehingga kita dapat membuat teks yang lebih menarik dan mudah dibaca.

```
<h1>Belajar dasar HTML</h1>
<h2>Klasifikasi Tag dalam HTML</h2>
<p><u>Tag teks</u></p><br>
<p><strong><em>Ini adalah contoh</em></strong></p>
```

Tag `<h1>` dan `<h2>` adalah tag heading, digunakan untuk menampilkan judul konten dengan tingkat kepentingan yang berbeda. "`<h1>`" adalah judul utama, sampai pada "`<h6>`" adalah judul terkecil. Tag `<p>` digunakan untuk membuat sebuah paragraf, tag `<u>` digunakan untuk menggaris-bawahi huruf, tag `<strong>` digunakan untuk menebalkan huruf, sedangkan tag `<em>` digunakan untuk membuat teks menjadi miring.

Belajar dasar HTML

Klasifikasi Tag dalam HTML

Tag teks

Ini adalah contoh

Gambar 2.3. Tampilan penggunaan tag teks

3. Tag Tautan dan Gambar

```
<a href="https://www.google.com">  
    
</a>
```

Diatas adalah contoh dari tag tautan `<a>` dan tag gambar `<img>`. Tag tautan (*link*) dan tag gambar (*image*) adalah dua tag penting dalam HTML yang sering digunakan untuk membuat tautan ke halaman web lain atau menampilkan gambar di halaman web.

4. Tag Daftar

Penjabaran pada klasifikasi tag HTML ini menggunakan fitur numbering untuk membuat daftar pada Microsoft Word. Apakah hal ini memungkinkan untuk dilakukan dengan menggunakan HTML? Ya, pada HTML kita bisa membuat daftar dengan menggunakan tag `<ol>` dan `<ul>`.

Tag `<ol>` atau *ordered list* digunakan untuk membuat sebuah daftar dengan menggunakan angka dan huruf yang berurutan, sedangkan `<ul>` atau *unordered list* digunakan untuk membuat daftar dengan menggunakan simbol (seperti fitur *bullets* pada Ms. Word).

Contoh Ordered List `<ol>`

```
<h1>Bahasa pemrograman faforitku:</h1>  
  <ol>  
    <li>Python</li>  
    <li>Java</li>  
    <li>PHP</li>
```

```
</ol>
```

Contoh Unordered List

```
<h1>Bahasa daerah di Indonesia</h1>
```

```
<ul>
```

```
<li>Bahasa Gorontalo</li>
```

```
<li>Bahasa Jawa</li>
```

```
<li>Bahasa Minang</li>
```

```
</ul>
```

Tag yang berada didalam tag dan berfungsi untuk membuat deretan list yang akan ditampilkan.

Bahasa pemrograman faforitku:

1. Python
2. Java
3. PHP

Bahasa daerah di Indonesia

- Bahasa Gorontalo
- Bahasa Jawa
- Bahasa Minang

Gambar 2.4. Tampilan penggunaan tag daftar

2.3.2 Ketentuan Penggunaan Tag HTML

Penggunaan tag dalam HTML memiliki beberapa ketentuan dan persyaratan yang harus diperhatikan. Beberapa di antaranya adalah:

1. Tag harus ditulis dengan benar: Setiap tag harus ditulis dengan benar, termasuk tag pembuka dan penutupnya. Tag yang ditulis dengan salah dapat mengakibatkan kesalahan dalam tampilan halaman web atau bahkan kesalahan dalam pemrosesan kode oleh browser.
2. Tag harus sesuai dengan fungsinya: Setiap tag harus digunakan sesuai dengan fungsinya. Misalnya, tag "<p>" digunakan untuk menandai paragraf, sedangkan tag "<h1>" digunakan untuk menandai judul halaman.

3. Tag dapat bersarang: Beberapa tag dapat bersarang satu sama lain, tetapi harus diletakkan secara benar dan sesuai dengan hierarki elemen pada halaman web. Contohnya adalah penggunaan tag teks dan tag daftar yang sudah disebutkan sebelumnya.
4. Untuk menentukan penggunaan mana yang lebih cocok antara tag dan sebenarnya tidak ada ketentuan tertentu yang diberlakukan, hanya saja disarankan untuk menggunakan tag jika daftar yang ingin dibuat mementingkan pengurutan seperti mana yang paling penting dan yang kurang penting, mana yang paling banyak dan yang sedikit, dan sebagainya.

2.4 Pengenalan Atribut dasar HTML

Beberapa tag dapat memiliki atribut yang harus diisi dengan nilai yang sesuai. Misalnya, tag memiliki atribut "src" yang harus diisi dengan lokasi gambar yang akan ditampilkan dan tag <a> yang memiliki atribut "href" yang digunakan untuk menjelaskan referensi halaman yang akan dituju oleh tautan.



Gambar 2.5. Penggunaan atribut dalam tag anchor

Untuk konten website yang memiliki dimensi (tinggi dan lebar) seperti gambar, video, garis, ataupun tabel, atribut "width" dan "height" dapat digunakan untuk menentukan seberapa besar dimensi yang akan digunakan dalam menampung konten-konten tersebut. Contohnya :

```
<hr width="70%" height="100px">
```

Atribut “width” digunakan untuk mengatur lebar elemen dan “height” ” digunakan untuk mengatur tinggi elemen. Nilai atribut ini dapat dinyatakan dalam piksel (px), persen (%), atau satuan lainnya tergantung pada konteks penggunaannya.

Atribut selanjutnya yang perlu diketahui dalam Bahasa HTML dasar adalah atribut “*style*”. Atribut “*style*” digunakan dalam tag HTML untuk menentukan gaya atau tampilan visual dari elemen tersebut. Dengan menggunakan atribut “*style*”, Anda dapat mengatur properti “*style*” seperti warna, ukuran, jenis huruf, tata letak, dan lainnya.

```
<p style="color: red; font-size: 18px;">Ini adalah paragraf dengan gaya khusus.</p>
```

Di dalam atribut “*style*”, properti dan nilai dipisahkan oleh tanda titik dua (:), dan setiap deklarasi properti-nilai dipisahkan oleh tanda titik koma (;). Anda dapat menyertakan beberapa deklarasi properti dalam atribut “*style*” untuk mengatur gaya yang lebih kompleks.

```
<div style="background-color: #f2f2f2; padding: 10px; text-align: center;">Isi div ini akan ditampilkan dengan gaya khusus.</div>
```

Ini adalah paragraf dengan gaya khusus.

Isi div ini akan ditampilkan dengan gaya khusus.

Gambar 2.6. Tampilan penggunaan atribut style

Perlu dicatat bahwa penggunaan atribut “*style*” cocok untuk mengatur gaya elemen secara langsung pada elemen tertentu. Namun, untuk pengaturan gaya yang lebih kompleks atau untuk mengaplikasikan gaya ke beberapa elemen, disarankan untuk menggunakan CSS terpisah dengan menggunakan selektor dan aturan CSS yang relevan.

Selain atribut "*style*", Anda juga dapat menggunakan tag `<style>` untuk menanamkan gaya CSS langsung di dalam dokumen HTML atau menyisipkan gaya melalui file eksternal dengan menggunakan tag `<link>` dan atribut "*href*".

DAFTAR PUSTAKA

- Setiawan, D. 2017. *Buku sakti pemrograman web: html, css, php, mysql & javascript*. Anak Hebat Indonesia.
- Enterprise, J. 2016. *pengenalan HTML dan CSS*. Elex Media Komputindo.
- Enterprise, J. 2018. *HTML, PHP, dan MySQL untuk Pemula*. Elex Media Komputindo.
- Nasution, F. P., Batubara, R. O., & Maulana, M. I. 2022. Dasar Pengenalan HTML pada Desain Web. *PUBLIDIMAS (Publikasi Pengabdian Masyarakat)*, 2(1), 86-91.
- www.w3schools.com

BAB 3

DASAR-DASAR CSS

Oleh Hermila A.

3.1 Pendahuluan

CSS merupakan singkatan dari Cascading Style Sheets yang memiliki fungsi sebagai bahasa sederhana yang umumnya dipakai dalam menambahkan *satyling* atau gaya seperti warna, font, ukuran pada halaman website yang dirancang. Ibaratnya adalah css ini bertindak sebagai kullit atau penutup dari HTML yang merupakan kerangka pemrograman web. Yang perlu digaris bawahi bahwa CSS merupakan bahasa pemrograman yang tak dapat menajalankan fungsi atau rumus logika yang umumnya digunakan dalam bahasa pemrogramn (1).



Gambar 3.1. Alur Kerja Desain Web
(Sumber: petanikode.com, 2022)

CSS pertama kali dibuat dan dikembangkan oleh *World Wide Consortium (W3C)* pada tahun 1996. Mulanya HTML tidak

dilengkapi dengan tags yang berfungsi untuk memberi style format halaman. Hanya dapat menulis markup untuk situs. Tag seperti , kemudian diperkenalkan HTML versi 3.2 yang kala itu menyebabkan berbagai masalah bagi depeloper aplikasi. Dimana website memiliki berbagai font, gaya dan background color, hal ini memerlukan proses yang sangat panjang dan sulit untuk menulis kembali kode sehingga W3C kemudian membuat CSS (2).

```
26 .screen-reader-text:hover,  
27 .screen-reader-text:active,  
28 .screen-reader-text:focus {  
29     background-color: #f1f3f4;  
30     border-radius: 3px;  
31     box-shadow: 0 0 2px 2px rgba(0, 0, 0, 0.6);  
32     clip: auto !important;  
33     color: #21759b;  
34     display: block;  
35     font-size: 14px;  
36     font-size: 0.875rem;  
37     font-weight: bold;  
38     height: auto;  
39     left: 5px;  
40     line-height: normal;  
41     padding: 15px 23px 14px;  
42     text-decoration: none;  
43     top: 5px;  
44     width: auto;  
45     z-index: 100000; /* Above WP toolbar. */  
46 }  
47
```

Gambar 3.2. Contoh Koding CSS
(Sumber: dicoding.com, 2022)

Berkembangnya teknologi juga mempengaruhi dunia pemrograman, CSS kini tidak lagi hanya digunakan untuk mendesain halaman web tetapi dapat pula digunakan untuk mendesain tampilan android. CSS dan HTML memiliki hubungan yang erat. Karena HTML merupakan fondasi situs dan CSS merupakan tata letak, style atau gaya dari website.

3.2 Fungsi CSS

CSS memiliki fungsi dan kegunaan yang sangat penting dalam membangun sebuah web. Berikut beberapa diantaranya:

1. Mengatur Tampilan (*Style/Gaya*)
CSS memiliki fungsi utama mengatur tampilan elemen-elemen pada halaman web. Menggunakan CSS dapat mengubah warna teks, jenis huruf, ukuran, latar belakang dan bentuk elemen. Kemudian dapat mengatur posisi, tata letak dan ruang antara elemen-elemen web.
2. Pemisah Konten dan Presentasi
CSS memungkinkan untuk memisahkan konten HTML dan CSS. Hal ini menjadi penting dalam mendesain dan mengembangkan web. Dengan memisahkan konten dan CSS, dapat memberi perubahan gaya tampilan secara konsisten diseluruh situs web hanya dengan mengubah file CSS, tanpa harus mengubah struktur konten
3. Efisiensi dan Konsistensi
Menggunakan CSS dapat mengatur gaya secara efisien dan konsisten diseluruh situs web. Mendefinisikan gaya sekali dan menerapkannya ke banyak elemen yang sama. Jika ingin merubah hanya perlu mengubah satu aturan CSS saja, dan perubahan tersebut akan diterapkan secara otomatis ke semua elemen yang menggunakan aturan tersebut.
4. Responsif dan Penyesuaian Layar
CSS memungkinkan membuat tampilan responsif yang menyesuaikan diri dengan perangkat yang digunakan oleh pengguna. Menggunakan media query dan teknik lainnya CSS dapat mengubah tata letak, ukuran elemen, dan tampilan halaman secara dinamis berdasarkan ukuran layar atau perangkat yang digunakan.
5. Efek
CSS memungkinkan untuk membuat animasi dan efek transisi yang menarik. Kemudian dapat mengubah

properti gaya seiring waktu, seperti perubahan posisi, perubahan warna, atau perubahan elemen. Hal ini juga memungkinkan memberikan interaksi dan pengalaman visual yang lebih menarik bagi pengguna.

6. Kompatibilitas

CSS didukung oleh hampir semua browser modern. Ini memungkinkan developer membuat tampilan yang konsisten diberbagai peramban web berbeda. Melalui pemanfaatan fitur-fitur CSS yang dioptimalkan untuk browser tertentu, dapat meningkatkan pengalaman pengguna dan menghindari masalah tampilan yang tidak konsisten di antara browser-browser.

3.3 Dasar Penggunaan CSS

CSS umumnya menggunakan bahasa inggris sederhana berbasis syntax yang dilengkapi dengan sekumpulan rule yang mengaturnya. File CSS disimpan dengan memberi ekstensi ".css" kemudian dihubungkan kedalam file HTML atau PHP yang telah di design. Hal yang perlu dipahami pada CSS dasar adalah konsep styling yang mengalir dari atas ke bawah. Jadi jika terdapat selektor dan properti yang sama, maka properti terakhir yang akan di-apply menjadi styling, berikut contohnya:

```
h1 {  
  Color: yellow;  
}  
h1 {  
  color: blue;  
}
```

Selanjutnya hal dasar dalam CSS yang perlu dipahami adalah tentang anatomi CSS. Ketika membuat kode CSS beberapa format yang harus diikuti.



Gambar 3.3. Anatomi CSS
Sumber: dicoding.com, 2022

Untuk lebih jelas berikut beberapa bagian penting dalam dasar CSS yang harus dipahami:

3.3.1 Selektor

CSS menggunakan selektor untuk menentukan elemen mana yang akan diberikan gaya. Misalnya, Anda dapat menggunakan selektor tag (seperti `<h1>`), selektor kelas (seperti `'kelas-saya'`), selektor ID (seperti `'#id-saya'`), atau selektor atribut (seperti `[type="text"]`). Melalui elemen-elemen selektor memungkinkan untuk memilih elemen yang tepat untuk diberi gaya.

```
<style>
  #mySelector { color: #454545; }
</style>
<div id='mySelector'>Halo dunia</div>
```

Gambar 3.4. Contoh penggunaan selektor

(Sumber : makinrajin.com, 2018)

3.3.2 Property

Properti CSS digunakan untuk mengatur gaya elemen yang dipilih. Contohnya, Anda dapat menggunakan properti **color** untuk mengubah warna teks, **font-size** untuk mengatur ukuran font, **background-color** untuk mengubah warna latar belakang, dan lain sebagainya. Berikut contoh property font size :

```
<p class="kecil">Ini adalah font kecil</p>
<p class="medium">Ini adalah font medium</p>
<p class="besar">Ini adalah font besar</p>
<p class="sangat-besar">Ini adalah font sangat besar</p>
```

3.3.3 Nilai

Nilai: Setiap properti memiliki nilai yang mengontrol bagaimana gaya akan diterapkan. Misalnya, untuk properti **'color'**, nilai bisa berupa nama warna seperti **'red'**, kode warna seperti **'#FF0000'**, atau nilai lain seperti **'rgb(255, 0, 0)'**.

```
<style>
  #mySelector { color: #454545; }
</style>
<div id='mySelector'>Halo dunia</div>
```

Gambar 3.5. Contoh Penggunaan nilai pada style variasi warna
(Sumber : makinrajin.com, 2018)

3.3.4 Kelas dan ID

Dalam CSS, Anda dapat menggunakan kelas dan ID untuk memberikan gaya kepada elemen tertentu. Kelas digunakan untuk mengelompokkan beberapa elemen yang

memiliki karakteristik serupa, sedangkan ID digunakan untuk mengidentifikasi elemen unik pada halaman.

```
<!DOCTYPE html>
<html>
<head>
  <title>Mengenal Class dan Id pada HTML | www.malasngoding.com</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>Mengenal Class dan Id pada HTML<br/>www.malasngoding.com</h1>
  <!-- contoh penggunaan class -->
  <div class="kotak">kotak 1</div>
  <div class="kotak">kotak 2</div>
  <div class="kotak">kotak 3</div>

  <!-- contoh penggunaan id -->
  <div id="kotak">Kotak 4</div>
</body>
</html>
```

Gambar 3.6. Contoh Penggunaan kelas dan id
(Sumber : makinrajin.com, 2018)

Gambar 3.4. Contoh Sintaks pemanggilan kelas dan id
css pada html
Sumber : malasngoding.com, 2021

```
h1{
    color: orange;
    font-family: sans-serif;
    text-align: center;
}

.kotak{
    padding: 50px;
    width: 100px;
    color: #fff;
    margin: 10px;
    background: orange;
}

#kotak{
    width: 400px;
    color: #fff;
    background: blue;
    padding: 50px;
}
```

Gambar 3.7. Contoh Sintaks kelas dan id css
Sumber : malasngoding.com, 2021

3.3.5 Penurunan

CSS menggunakan konsep penurunan (*cascading*) yang berarti bahwa gaya dapat diwariskan dari elemen induk ke elemen anak. Ini memungkinkan Anda mengatur gaya secara efisien dengan mendefinisikan gaya umum untuk elemen induk dan menerapkan gaya tambahan pada elemen anak. Berikut contoh sintaks penurunan menggunakan property inherit.

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Inheritance dalam CSS</title>
  <style type="text/css">
    div.satu {
      border: 1px solid black;
      color: blue;
    }

    div.dua {
      border: 1px solid black;
    }
    p.dua {
      border: inherit;
    }
  </style>
</head>

<body>
  <div class="satu">
    <p class="satu">
      Kalimat ini tanpa <em class="satu">nilai property inherit</em> CSS
    </p>
  </div>
  <br />
  <div class="dua">
    <p class="dua">
      Kalimat ini menggunakan <em class="dua">nilai property inherit</em> CSS
    </p>
  </div>
</body>
</html>
```

Gambar 3.8. Contoh Sintaks penurunan css

Sumber : malasngoding.com, 2021

3.3.6 Posisi dan Tata Letak

CSS memiliki kemampuan untuk mengatur tata letak elemen pada halaman web. Anda dapat menggunakan properti seperti **position**, **display**, dan **float** untuk mengatur posisi elemen, dan properti seperti **margin**, **padding**, dan **width** untuk mengatur tata letak elemen.

```
#relative {
    background-color: yellow;
    position: relative;
    top: 20px;
    left: 100px;
    width: 30%;
    height: 80px;
}
```

Gambar 3.9. Contoh Sintaks penurunan css
Sumber : penulis, 2023

3.3.7 Responsif

Dengan CSS, Anda dapat membuat tampilan responsif yang menyesuaikan diri dengan perangkat yang digunakan oleh pengguna. Ini dapat dilakukan dengan menggunakan media query dan teknik lainnya untuk mengubah tampilan halaman berdasarkan ukuran layar atau perangkat yang digunakan.

3.3.8 Selektor Pseduo-Class dan Psedue-element

CSS mendukung selektor pseudo-class dan pseudo-element yang memungkinkan Anda meng gaya elemen berdasarkan keadaan atau posisi mereka dalam struktur dokumen. Misalnya, Anda dapat menggunakan `:hover` untuk mengubah gaya saat pengguna mengarahkan kursor ke atas elemen, atau `:before` untuk menambahkan konten sebelum elemen tertentu.

Pseudo class selector merupakan jenis selektor CSS yang diikuti oleh titik dua. Berikut contoh penggunaan :

```
a:hover {
    /* Yep, hover adalah sebuah pseudo class */
}
```

Pseudo-element berfungsi menyeleksi ‘potongan’ tag atau mengakses sebuah elemen yang sebelumnya tidak ada.

Contoh pseudo-element dapat digunakan untuk menyeleksi huruf pertama pada sebuah paragraf.

```
p::first-letter { color:red; }
```

3.3.9 Animasi dan Transisi

Belajar tentang animasi dan efek transisi menggunakan properti seperti animation, transition, dan keyframes akan memungkinkan Anda untuk membuat pergerakan dan perubahan yang halus pada elemen. Berikut contoh sintaks transisi.

Index.html

```
<!DOCTYPE html>
<html>
<head>
<title>Tutorial          CSS3          Transition          |
www.malasngoding.com</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<h1>Tutorial CSS3 Transition | www.malasngoding.com</h1>
<div class="kotak">ini kotak</div>
<div class="kotak2">ini kotak 2</div>
</body>
</html>
```

Style.css

```
body{ background: #35A9DB;
      font-family: roboto;
      text-align: center;
}
h1{
    color: #fff;
}
div{
    margin: 10px;
```

```

}
.kotak{      background: #fff;
              width: 100px;
              height: 100px;
              transition: width 1s;
              -webkit-transition: width 1s;
}
.kotak:hover{
              width: 300px;
}
.kotak2{      background: #fff;
              width: 100px;
              height: 100px;
              transition: width 1s, height 2s;
              -webkit-transition: width 1s, height 2s;
}
.kotak2:hover{
              width: 300px;
              height: 300px;
}

```

3.3.10 Modularitas dan Pemeliharaan.

Memahami prinsip-prinsip desain CSS yang baik seperti modularitas, reusable styles, dan pemeliharaan yang efisien akan membantu Anda membuat kode CSS yang terorganisir, mudah diubah, dan dapat dikelola dengan baik.

3.3.11 Kompatibilitas Browser

Mengetahui perbedaan implementasi CSS di berbagai browser dan cara menangani masalah kompatibilitas akan memastikan tampilan yang konsisten di seluruh platform.

3.4 Cara Menulis CSS

Dalam menulis CSS umumnya ada tiga cara yaitu CSS Style Internal, External, dan Inline.

3.4.1 CSS Style Internal

CSS style internal adalah gaya CSS yang didefinisikan secara langsung di dalam dokumen HTML menggunakan elemen `<style>`. Gaya ini berlaku hanya untuk halaman HTML yang spesifik. Untuk mempelajari CSS style internal, Anda perlu memahami sintaks CSS untuk memilih elemen HTML dan mendefinisikan properti gaya. Anda juga perlu belajar tentang konsep selector, deklarasi, dan aturan CSS.

A screenshot of a Notepad window titled 'Ccu-dasar - Notepad'. The window shows HTML code with internal CSS styles. The code includes a head section with a title 'Contoh Internal CSS' and a style block for paragraphs (p) and headings (h2). The body section contains a heading (h2) and a paragraph (p). The code is as follows:

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Internal CSS</title>
  <!-- penulisan internal css dalam tag head -->
  <style type="text/css">
    p{
      font-family: serif;
      line-height: 1.75em;
      font-size: 18px;
    }
    h2 {
      font-family: sans;
      color: orange;
    }
  </style>
</head>

<body>
  <!-- penulisan internal css dalam tag body -->
  <style type="text/css">
    h2 {
      font-family: sans;
      color: #333;
    }
  </style>
  <h2>Ini judul artikel</h2>
  <p>Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja.
  Percobaan untuk mendemonokan <i>internal css</i>. Seperti namanya,
  <i>inline CSS</i> adalah kode CSS yang ditulis langsung dalam file HTML.</p>
</body>
</html>
```

Gambar 3.10. Contoh Sintaks Kode CSS Internal Style
(Sumber : petanikode.com, 2017)

Ini judul artikel

Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja.

Percobaan untuk mendemonikan *internal css*. Seperti namanya, *inline CSS* adalah kode CSS yang ditulis langsung dalam file HTML.

Gambar 3.11. Hasil Contoh Sintaks Kode CSS Internal Style
(Sumber : petanikode.com, 2017)

Tag **<head>** dalam penulisan css di atas menunjukkan bahwa akan lebih dibandingkan **<body>**, style yang dipakai adalah yang terakhir.

3.4.2 CSS Style External

CSS *style external* adalah gaya CSS yang didefinisikan di dalam file eksternal terpisah dengan ekstensi .css. File CSS ini kemudian disambungkan dengan dokumen HTML menggunakan elemen **<link>**. Dengan menggunakan CSS style external, Anda dapat menerapkan gaya yang sama ke beberapa halaman HTML. Untuk mempelajari CSS style external, Anda perlu memahami bagaimana menyusun file CSS terpisah dengan benar dan menggunakan elemen **<link>** untuk menghubungkannya dengan dokumen HTML. Berikut contoh penggunaan CSS Style Eksternal :

1. Membuat sebuah file terlebih dahulu dan beri nama style-ku.css kemudian ketiklah kode css berikut pada notepad.

```
p {  
    font-family: serif;  
    line-height: 1.75em;  
}  
i {  
    font-family: sans;  
    color: orange;
```

```

}
h2 {
    font-family: sans;
    color: #333;
}

```

2. Selanjutnya untuk menggunakan style pada no.1 di atas, harus dihubungkan dengan HTML menggunakan **<link>** atau menggunakan **@import**.

```
<link rel="stylesheet" type="text/css" href="style-ku.css">
```

```

<style type="text/css">
@import "style-ku.css";
</style>

```



```

*style-eksternal - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>
    <title>Contoh Eksternal CSS</title>
    <link rel="stylesheet" type="text/css" href="style-ku.css">
</head>

<body>
    <h2>Ini judul artikel</h2>
    <p>Ini adalah paragraf yang memuat isi artikel.
    Paragraf ini hanya untuk percobaan saja.
    Percobaan untuk mendemokan <i>internal css</i>. Seperti namanya,
    <i>inline CSS</i> adalah kode CSS yang ditulis langsung dalam file HTML.</p>
</body>
</html>

```

Gambar 3.12. Hasil Contoh Sintaks Kode CSS Eksternal Style

(Sumber : petanikode.com, 2017)

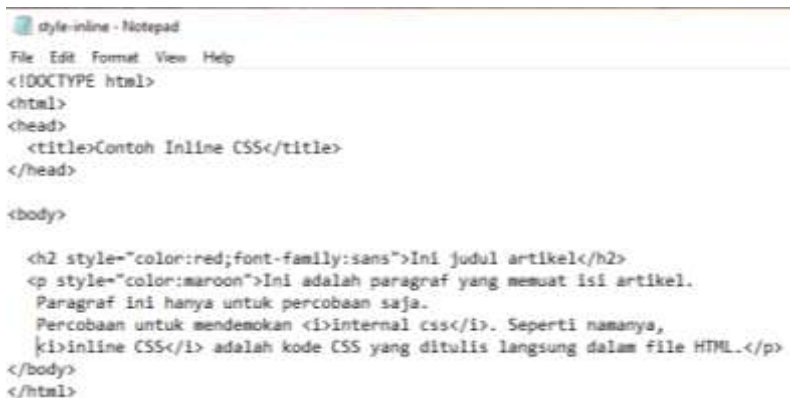
Ini judul artikel

Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja. Percobaan untuk mendemonstrasikan *internal css*. Seperti namanya, *inline CSS* adalah kode CSS yang ditulis langsung dalam file HTML.

Gambar 3.13. Hasil Contoh Sintaks Kode CSS Eksternal Style
(Sumber : petanikode.com, 2017)

3.4.4 CSS Style Inline

CSS style inline adalah gaya CSS yang didefinisikan langsung di dalam atribut elemen HTML. Setiap elemen HTML dapat memiliki atribut "style" yang berisi deklarasi gaya CSS. Gaya ini hanya berlaku untuk elemen HTML yang spesifik dan mengambil prioritas tertinggi dibandingkan dengan CSS internal dan external. Untuk mempelajari CSS style inline, Anda perlu memahami sintaks CSS dan memahami cara menyematkan properti gaya langsung ke dalam atribut elemen HTML. Kode CSS ditulis langsung tanpa menggunakan kurung kurawal { ... }. Kemudian properti tetap dipisah dengan titik koma.



```
style-inline - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Inline CSS</title>
</head>
<body>
  <h2 style="color:red;font-family:sans">Ini judul artikel</h2>
  <p style="color:maroon">Ini adalah paragraf yang memuat isi artikel.
  Paragraf ini hanya untuk percobaan saja.
  Percobaan untuk mendemonstrasikan <i>internal css</i>. Seperti namanya,
  <i>inline CSS</i> adalah kode CSS yang ditulis langsung dalam file HTML.</p>
</body>
</html>
```

Gambar 3.14. Contoh Sintaks Kode CSS Inline Style
(Sumber : petanikode.com, 2017)

Ini judul artikel

Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja. Percobaan untuk mendemonikan *internal css*. Seperti namanya, *inline CSS* adalah kode CSS yang ditulis langsung dalam file HTML.

Gambar 3.15. Hasil Contoh Sintaks Kode CSS Inline Style
(Sumber : petanikode.com, 2017)

DAFTAR PUSTAKA

- Septiyanto A. Dasar Penggunaan CSS [Internet]. Dicoding Blog. 2022 [cited 2023 May 25]. Available from: <https://www.dicoding.com/blog/dasar-penggunaan-css/>
- C A. Apa Itu CSS? Pengertian, Fungsi, dan Cara Kerjanya [Internet]. Hostinger Tutorial. 2019 [cited 2023 May 25]. Available from: <https://www.hostinger.co.id/tutorial/apa-itu-css>

BAB 4

DASAR-DASAR JAVASCRIPT

Oleh Zen Munawar

4.1 Pendahuluan

JavaScript adalah bahasa skrip sisi klien open-source dan paling populer yang didukung oleh semua browser. JavaScript terutama digunakan untuk meningkatkan interaksi halaman web dengan pengguna dengan membuatnya lebih hidup dan interaktif. Ini juga digunakan untuk pengembangan game dan pengembangan aplikasi seluler. JavaScript adalah bahasa pemrograman yang menambahkan interaktivitas ke situs web Anda. Ini terjadi dalam game, dalam perilaku respons saat tombol ditekan atau saat memasukkan data ke formulir; dengan gaya dinamis; dengan animasi, dan lain-lain. Bab ini membantu Anda memulai dengan JavaScript dan memperluas pemahaman

Anda tentang apa yang mungkin. JavaScript adalah bahasa pemrograman yang kuat yang dapat menambahkan interaktivitas ke situs web (Svekis, Putten and Percival, 2021). JavaScript ditemukan oleh Brendan Eich. JavaScript serbaguna dan ramah bagi pemula. Dengan lebih banyak pengalaman, Anda akan dapat membuat game, grafik animasi 2D dan 3D, aplikasi berbasis database yang komprehensif, dan banyak lagi. JavaScript sendiri relatif kompak, namun sangat fleksibel. Pengembang telah menulis berbagai alat di atas bahasa JavaScript inti, membuka sejumlah besar fungsi dengan sedikit usaha. Ini termasuk: Application Programming Interfaces (API) dibangun ke dalam browser web, menyediakan fungsionalitas seperti membuat HTML secara dinamis dan menyetel gaya CSS;

mengumpulkan dan memanipulasi aliran video dari webcam pengguna, atau membuat grafik 3D dan sampel audio.



Gambar 4.1. Brendan Eich, Pencipta JavaScript
Sumber : (Tyson, 2022)

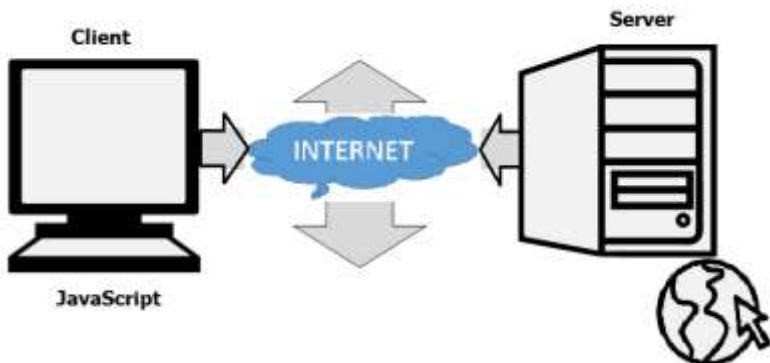
Brendan Eich sebagai pencipta JavaScript, co-founder Mozilla, dan sekarang CEO Brave bisa dilihat pada gambar 4.1 (Tyson, 2022). *Application* Terkenal sebagai pencipta bahasa pemrograman JavaScript, Brendan Eich juga merupakan pendiri dan CEO Brave Software, pencipta browser Brave, dan co-creator dari Basic Attention Token. Sebagai CTO Mozilla, dia membantu merintis browser web dari zaman kegelapan.

Programming Interfaces pihak ketiga memungkinkan pengembang menggabungkan fungsionalitas di situs dari penyedia konten lain, seperti Twitter atau Facebook. Framework dan library pihak ketiga yang dapat Anda terapkan ke HTML untuk mempercepat pekerjaan pembuatan situs dan aplikasi. Ini di luar cakupan bab ini—sebagai pengantar ringan untuk JavaScript—untuk menyajikan detail perbedaan bahasa inti JavaScript dari alat yang tercantum di atas. Anda dapat mempelajari lebih lanjut di area pembelajaran JavaScript MDN, serta di bagian lain MDN. Bagian di bawah memperkenalkan

beberapa aspek bahasa inti dan menawarkan kesempatan untuk bermain dengan beberapa fitur API browser juga.

4.2 Apa itu JavaScript

JavaScript adalah bahasa pemrograman dinamis yang digunakan untuk pengembangan web, dalam aplikasi web, untuk pengembangan game, dan banyak lagi JavaScript adalah bahasa pemrograman web dan digunakan oleh lebih banyak pengembang perangkat lunak saat ini daripada bahasa pemrograman lainnya (Flanagan, 2020). Ini memungkinkan Anda untuk menerapkan fitur dinamis pada halaman web yang tidak dapat dilakukan hanya dengan HTML dan CSS (Megida, 2021). Serangkaian penemuan yang diusulkan mencakup konstruksi dan model untuk merepresentasikan hubungan antara komponen, produk, dan infrastruktur teknologi informasi, serta metode baru untuk mengidentifikasi dan merepresentasikan pola evolusi teknologi. Bersama-sama, mereka merupakan proses baru untuk mewakili, memahami, mengevaluasi, dan meramalkan lanskap teknologi informasi, sehingga memungkinkan manajer mengambil keputusan investasi teknologi informasi yang lebih efektif.



Gambar 4.2. Cara Kerja JavaScript

Bahasa ini awalnya disebut LiveScript dan kemudian berganti nama menjadi JavaScript. Perbedaan antara bahasa skrip sisi server dan sisi klien. Bahasa skrip sisi klien : lingkungan sisi klien yang digunakan untuk menjalankan skrip biasanya adalah browser, Pemrosesan berlangsung di komputer pengguna akhir (Sastradipraja and Munawar, 2022). Banyak programmer yang menganggap JavaScript dan Java itu sama. Faktanya, JavaScript dan Java sangat tidak berhubungan. Java adalah bahasa pemrograman yang sangat kompleks sedangkan JavaScript hanyalah bahasa scripting. Sintaks JavaScript sebagian besar dipengaruhi oleh bahasa pemrograman C. Umumnya browser menggunakan JavaScript sebagai bahasa skrip untuk melakukan hal-hal dinamis di web. Setiap kali Anda melihat menu tarik-turun klik-untuk-tampilkan, konten tambahan ditambahkan ke halaman, dan mengubah warna elemen secara dinamis pada halaman, untuk menyebutkan beberapa fitur, Anda melihat efek JavaScript.

Tanpa JavaScript, semua yang Anda miliki di web hanyalah HTML dan CSS. Ini saja membatasi Anda pada beberapa implementasi halaman web. 90% (jika tidak lebih) halaman web Anda akan statis, dan Anda hanya akan memiliki perubahan dinamis seperti animasi yang disediakan CSS. Dalam JavaScript dan jQuery: Pengembangan Front-End Interaktif, panduan untuk membuat situs web Anda lebih interaktif dan antarmuka Anda lebih menarik dan intuitif (Duckett, 2014). Dalam JavaScript, Anda akan menjelajahi konsep pemrograman dasar yang tidak mengasumsikan pengetahuan sebelumnya tentang pemrograman di luar kemampuan membuat halaman web menggunakan HTML & CSS. HTML mendefinisikan struktur dokumen web Anda dan konten di dalamnya. CSS mendeklarasikan berbagai gaya untuk konten yang disediakan di dokumen web. HTML dan CSS sering disebut bahasa markup daripada bahasa pemrograman, karena pada intinya mereka menyediakan markup untuk dokumen dengan sedikit

dinamisme. JavaScript, di sisi lain, adalah bahasa pemrograman dinamis yang mendukung perhitungan Matematika, memungkinkan Anda menambahkan konten HTML ke DOM secara dinamis, membuat deklarasi gaya dinamis, mengambil konten dari situs web lain, dan banyak lagi.

4.2.1 Menjalankan JavaScript

Menjadi bahasa scripting, JavaScript tidak dapat berjalan sendiri (Osmani, 2023). Faktanya, browser bertanggung jawab untuk menjalankan kode JavaScript. Saat pengguna meminta halaman HTML dengan JavaScript di dalamnya, skrip dikirim ke browser dan terserah browser untuk mengeksekusinya. Keuntungan utama JavaScript adalah semua browser web modern mendukung JavaScript. Jadi, Anda tidak perlu khawatir apakah pengunjung situs Anda menggunakan Internet Explorer, Google Chrome, Firefox atau browser lainnya. JavaScript akan didukung. Juga, JavaScript berjalan pada sistem operasi apa pun termasuk Windows, Linux atau Mac. Dengan demikian, JavaScript mengatasi kelemahan utama VBScript (Sekarang sudah tidak digunakan lagi) yang terbatas hanya pada IE dan Windows.

Alat yang Anda Butuhkan, untuk memulainya, Anda memerlukan editor teks untuk menulis kode dan browser untuk menampilkan halaman web yang Anda kembangkan. Anda dapat menggunakan editor teks pilihan Anda termasuk Notepad++, Visual Studio Code, Sublime Text, Atom atau editor teks lain yang Anda sukai. Anda dapat menggunakan browser web apa pun termasuk Google Chrome, Firefox, Microsoft Edge, Internet Explorer dan lain-lain.

4.2.2 Cara Menggunakan JavaScript di HTML

Sama seperti CSS, JavaScript dapat digunakan dalam HTML dengan berbagai cara, seperti:

1. Inline JavaScript

Di sini, Anda memiliki kode JavaScript dalam tag HTML di beberapa atribut khusus berbasis JS.

Misalnya, tag HTML memiliki atribut peristiwa yang memungkinkan Anda mengeksekusi beberapa kode inline saat event trigger. Contoh kode Inline:

```
<button onclick="alert('Anda baru saja mengklik tombol')">Klik disini!</button>
```

ini adalah contoh JavaScript sebaris. Nilai onclick dapat berupa perhitungan Kecocokan, penambahan dinamis ke DOM – kode JavaScript apa pun yang valid sintaks.

2. JavaScript internal, dengan tag skrip

Sama seperti tag gaya untuk deklarasi gaya dalam laman HTML, tag skrip ada untuk JavaScript. Berikut cara penggunaannya:

```
<script>
    function(){
        alert("Saya di dalam tag skrip ")
    }
</script>
```

3. JavaScript eksternal

Anda mungkin ingin memiliki kode JavaScript di file yang berbeda. JavaScript eksternal memungkinkan ini. Untuk kasus penggunaan seperti itu, begini caranya:

```
<!-- index.html -->
<script src="./script.js"></script>

// script.js
alert("Saya ada di dalam file eksternal");
```

Atribut src dari tag skrip memungkinkan Anda menerapkan sumber untuk kode JavaScript. Referensi itu

penting karena memberi tahu browser untuk juga mengambil konten script.js. script.js bisa berada di direktori yang sama dengan index.html, atau bisa didapatkan dari website lain. Untuk yang terakhir, Anda harus memberikan URL lengkap (<https://.../script.js>).

Perhatikan ekstensi .js? Itu adalah ekstensi untuk file JavaScript, sama seperti HTML yang memiliki .html. Sekarang kita telah melihat cara untuk menerapkan JavaScript ke HTML kita, mari kita lihat beberapa fitur JavaScript.

4.2.3 Tipe Data dalam JavaScript

Dalam JavaScript, data harus dari satu jenis atau lainnya (Freeman and Robson, 2014). JavaScript perlu mengetahui hal ini agar mengetahui cara menggunakannya dengan data lain atau cara mengoperasikan data tersebut.

Berikut adalah tipe data dasar yang didukung JavaScript:

1. **Number** (misalnya, 6, 7, 8.9): di mana Anda dapat menerapkan operasi aritmatika (seperti penjumlahan) dan banyak lagi
2. **String** (seperti "javascript", 'kalimat panjang', paragraf pendek): apa pun yang ditemukan di antara tanda kutip tunggal ('...'), tanda kutip ganda ("...") dan backticks (...). Tidak ada perbedaan antara tanda kutip tunggal dan ganda, tetapi tanda kutip balik memiliki lebih banyak fitur, seperti: interpolasi variabel dalam string, seperti: Nama saya \${name}. nama di sini adalah variabel, di injeksi ke dalam string. String multi baris. Dengan tanda kutip normal, Anda perlu menambahkan karakter escape seperti `\n` untuk baris baru, tetapi backticks memungkinkan Anda melanjutkan string di baris lain, seperti:

```
let str = `Saya adalah string multibaris`;
```

3. **Boolean** (hanya bisa dua nilai: benar atau salah): lebih seperti ya (benar) atau tidak (salah)

4. **Array** (misalnya, [1, 2, "halo", salah]): sekelompok data (yang dapat bertipe apa pun, termasuk larik) yang dipisahkan dengan koma. Pengindeksan dimulai dari 0. Mengakses konten grup semacam itu bisa seperti ini: `array[0]`, yang untuk contoh ini akan mengembalikan 1, karena ini adalah item pertama.
5. **Object** (misalnya {nama: 'javascript', usia: 5}): juga merupakan kumpulan data tetapi dalam bentuk pasangan kunci:nilai. Kuncinya harus berupa string, dan nilainya bisa berupa tipe apa pun termasuk objek lain. Mengakses konten grup dilakukan dengan kunci, misalnya `obj.age` atau `obj["age"]` akan mengembalikan 5.
6. **Undefined** (satu-satunya data yang didukung tipe ini adalah tidak terdefinisi): Data ini dapat diberikan ke variabel secara eksplisit, atau secara implisit (oleh JavaScript) jika variabel telah dideklarasikan tetapi tidak diberi nilai. Nanti di artikel ini, kita akan melihat deklarasi variabel dan penetapan nilai.
7. **Null** (satu-satunya data yang didukung tipe ini adalah null): Null berarti tidak ada nilai. Itu memegang nilai, tetapi bukan nilai sebenarnya – lebih tepatnya, nol.
8. **Function** (misalnya, `function(){ console.log("function")}`): Fungsi adalah tipe data yang memanggil blok kode saat dipanggil.

Tipe data JavaScript bisa sedikit rumit untuk dipahami. Anda mungkin pernah mendengar bahwa array dan fungsi juga merupakan objek, dan itu benar. Memahami ini melibatkan pemahaman tentang sifat prototipe JavaScript. Namun, pada tingkat dasar, ini adalah tipe data yang perlu Anda ketahui terlebih dahulu di JavaScript.

4.2.4 Variabel dalam JavaScript

Variabel adalah wadah untuk nilai dari semua tipe data. Mereka menyimpan nilai sedemikian rupa sehingga ketika variabel digunakan, JavaScript menggunakan nilai yang mereka wakili untuk operasi itu. Variabel dapat dideklarasikan dan dapat diberi nilai. Saat Anda mendeklarasikan variabel, Anda melakukan ini:s

```
let nama;
```

Untuk yang di atas, nama sudah dideklarasikan, tetapi belum memiliki nilai. Seperti yang Anda harapkan dari bagian tipe data, JavaScript secara otomatis memberikan nilai undefined ke nama. Jadi jika Anda mencoba menggunakan nama di mana saja, undefined akan digunakan untuk operasi itu. Inilah yang dimaksud dengan menetapkan nilai ke variabel:

```
let nama;  
nama = "JavaScript";
```

Sekarang jika Anda menggunakan nama, itu akan mewakili JavaScript. Deklarasi dan penugasan dapat dilakukan pada satu baris seperti ini:

```
let nama = "JavaScript";
```

Mengapa membiarkan? Anda mungkin bertanya pada diri sendiri, dan inilah alasannya: JavaScript mendukung tiga metode deklarasi variabel, yaitu:

1. Operator var: ini sudah ada dengan JavaScript sejak awal. Anda dapat mendeklarasikan variabel dan memberikan nilai padanya yang dapat diubah nanti dalam kode. Inilah yang saya maksud:

```
var nama = "JavaScript";  
nama = "Bahasa";
```

2. Operator `let`: ini juga sangat mirip dengan `var` – ia mendeklarasikan dan menetapkan nilai ke variabel yang dapat diubah nanti dalam kode. Perbedaan utama antara operator ini adalah bahwa `var` mengerek variabel tersebut, sedangkan `let` tidak mengerek. Konsep hoisting dapat dijelaskan secara singkat dengan kode berikut:

```
function print() {  
    console.log(nama);  
    console.log(umur);  
    var nama = "JavaScript";  
    let umur = 5;  
}  
print();
```

Saat memanggil fungsi cetak (`print()`), `console.log` pertama mencetak `undefined` sementara `console.log` kedua melontarkan kesalahan bahwa "Tidak dapat mengakses usia sebelum inisialisasi".

Ini karena deklarasi variabel `nama` diangkat (diangkat) ke atas fungsi dan penugasan untuk variabel tetap pada baris yang sama sedangkan deklarasi usia dan penugasan tetap pada baris yang sama. Beginilah cara kode di atas dikompilasi:

```
function print() {  
    var nama;  
    console.log(nama);  
    console.log(umur);  
    nama = "JavaScript";  
    let umur = 5;  
}  
print();
```

Masalah hoist dapat terjadi secara tidak terduga, dan itulah mengapa Anda harus menggunakan `let` daripada `var`. Operator `const`: ini juga tidak mengangkat variabel, tetapi

melakukan satu hal lagi: memastikan bahwa variabel tidak dapat diberi nilai lain selain dari apa yang ditugaskan selama inisialisasi.

Misalnya:

```
let nama = "JavaScript"  
nama = "Bahasa" // tidak ada kesalahan  
const umur = 5  
umur = 6 // kesalahan, tidak dapat menugaskan  
kembali variabel
```

4.2.5 Komentar dalam JavaScript

Sama seperti HTML, terkadang kita mungkin ingin memberi catatan di samping kode kita yang tidak perlu dijalankan. Kita dapat melakukannya di JavaScript dengan dua cara: dengan komentar satu baris, seperti ini: `// komentar satu baris`

Atau dengan komentar multibaris, seperti ini:

```
/*  
komentar  
multi baris  
*/
```

4.3 Fungsi dalam JavaScript

Dengan fungsi, Anda dapat menyimpan blok kode yang dapat digunakan di tempat lain dalam kode Anda. Katakanlah Anda ingin mencetak "JavaScript" dan "Bahasa" di tempat berbeda dalam kode Anda.

Perhatikan contoh berikut ini :

```
console.log("JavaScript")  
console.log("Bahasa")  
  
// beberapa hal di sini
```

```
console.log("JavaScript")
console.log("Bahasa")

// lebih banyak hal di sini

console.log("JavaScript")
console.log("Bahasa")
```

Dapat ditulis dengan fungsi seperti berikut ini :

```
fungsi cetak() {
  console.log("JavaScript")
  console.log("Bahasa")
}

mencetak()

// beberapa hal di sini

mencetak()

// lebih banyak hal di sini

mencetak()
```

Dengan cara ini, maka telah menyimpan blok kode berulang dalam fungsi yang dapat digunakan di mana pun kami mau. Tapi itu belum semuanya. Katakanlah kita ingin mencari rata-rata dari tiga angka. Kode untuk ini adalah:

```
let bil1 = 5
let bil2 = 6
let bil3 = 8
let rata_rata = (bil1 + bil2 + bil3) / 3
```

Melakukan ini di luar fungsi mungkin tidak ada salahnya, tetapi jika kita harus melakukannya di banyak tempat? Maka akan memiliki fungsi seperti ini:

```
function menentukanRata_rata(n1, n2, n3) {  
  let rata = (n1 + n2 + n3) / 3  
  return rata  
}  
  
let bil1 = 5  
let bil2 = 6  
let bil3 = 8  
let ratarata = menentukanRata_rata(bil1, bil2, bil3)  
  
// nanti, di tempat lain  
let ratarata2 = menentukanRata_rata(...)  
  
// nanti, di tempat lain  
let ratarata3 = menentukanRata_rata(...)
```

Seperti yang akan Anda lihat dalam deklarasi `menentukanRata_rata`, memiliki `n1`, `n2`, `n3` di dalam tanda kurung. Ini adalah parameter, yang berfungsi sebagai tempat penampung untuk nilai yang akan diberikan saat fungsi dipanggil. Blok kode menggunakan placeholder tersebut untuk menemukan rata-rata, dan kata kunci `return` mengembalikan rata-rata dari fungsi. Hal ini membuat fungsi Anda dapat digunakan kembali sehingga nilai yang berbeda pada waktu yang berbeda dapat diteruskan ke fungsi untuk menggunakan logika yang sama.

DAFTAR PUSTAKA

- Duckett, J. 2014. *JavaScript and jQuery: Interactive Front-End Web Development*. 1st edn. Wiley.
- Flanagan, D. 2020. *JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language*. 7th edn. O'Reilly Media.
- Freeman, E. and Robson, E. 2014. *Head First JavaScript Programming: A Brain-Friendly Guide*. 1st edn. O'Reilly Media.
- Megida, D. 2021. *What is JavaScript? A Definition of the JS Programming Language*, FreeCodeCamp. Available at: <https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/> (Accessed: 2 April 2023).
- Osmani, A. 2023. *Learning JavaScript Design Patterns: A JavaScript and React Developer's Guide*. 2nd edn. O'Reilly Media.
- Sastradipraja, C. K. and Munawar, Z. 2022. *Konsep Dasar Teknologi Web*. 1st edn. Edited by H. H. Solihin. Bandung: Kaizen Media Publishing.
- Svekis, L. L., Putten, M. van and Percival, R. 2021. *JavaScript from Beginner to Professional: Learn JavaScript quickly by building fun, interactive, and dynamic web apps, games, and pages*. 1st edn. Packt Publishing.
- Tyson, M. 2022. *Brendan Eich: Don't blame cookies and JavaScript*, InfoWorld. Available at: <https://www.infoworld.com/article/3673292/brendan-eich-dont-blame-cookies-and-javascript.html> (Accessed: 2 March 2023).

BAB 5

MEMBUAT HALAMAN RESPONSIF

Oleh Rony Setiawan

5.1 Pendahuluan

Perkembangan perangkat keras untuk dapat mengakses informasi di internet sangat beragam dengan berbagai jenis ukuran seperti pada perangkat monitor komputer, laptop, tablet, smartphome dengan berbagai ukuran layar dan memiliki resolusi atau kerapatan yang berbeda-beda, misal layar laptop ukuran 13 inchi 3200 x 1800 pixel, layar tablet 11 inchi 2500 x 1600 pixel, ukuran layar ponsel 5,6 inchi memiliki resolusi 1280 x 720 pixel, layar laptop ukuran 13 inchi 3200 x 1800 dan masih banyak lagi ukuran layar dan ukuran resolusi layar (Nando Rifky, 2020).

Dengan beragamnya ukuran layar, harapan setiap pengguna dapat mengakses informasi dari website dengan menampilkan informasi yang sama dengan tampilan yang mampu menyesuaikan dengan ukuran layar. Setiap pengguna akan memiliki pengalaman yang berbeda ketika menggunakan perangkat yang berbeda ukuran layarnya, yaitu ketika menggunakan monitor komputer, tablet maupun smartphome. Pengguna smartphome di Indonesia sampai tahun 2023 sebesar 233,49 juta dan diprediksi sampai 2018 sebesar 268.82 juta (Statista, 2023), berdasarkan data diatas para mengembang/developer aplikasi berbasis web harus mampu merancang sebuah aplikasi website yang mampu menyesuaikan tampilan layarnya dengan berbagai perangkat dan berbagai ukuran layarnya, sehingga para pengguna akan

memiliki pengalaman mengakses website yang mudah dengan berbagai perangkat.

5.2 Pengertian Halaman Responsif

Halaman responsif disebut juga *responsive web*, merupakan suatu halaman web yang mampu menyesuaikan tampilan layarnya dengan jenis perangkat keras dan ukuran layar yang berbeda pada saat diakses oleh pengguna dari berbagai perangkat, dan tampilan layar akan secara otomatis halaman web berubah menyesuaikan dengan ukuran layar baik *user interface*, tata letak, ukuran huruf, dan gambar akan menyesuaikan diri dengan resolusi *device* yang sedang digunakan.

Bagi para perancang aplikasi web, wajib menerapkan halaman responsif pada setiap rancangan layar, untuk merancang halaman web yang responsif sangatlah mudah dengan menguasai pemrograman HTML 5 dan *Cascading Style Sheet 3*(CSS 3).

Ethan Marcotte adalah orang pertama yang mencetuskan *responsive web design* yang mencetuskan responsive web dalam sebuah artikel ia mengulas tiga teknik dalam merancang halaman yang reponsif yakni Flexible grid layout, flexible images, dan media queries yang ia sebut dengan *Responsive Design* (Ethan Marcotte, 2010), ia juga menggunakan pendekatan lain yakni *fluid design*, *rubber layout*, *elastic layout*, *rubber layout*, *liquid design*, *cross-device design*, *adaptive layout*, dan *flexible design*.

Pada Gambar dibawah ini merupakan salah satu tampilan web site dengan menggunakan rancangan halaman responsif yang ditampilkan dari berbagai *device* dengan berbagai ukuran layar.



Gambar 5.1. Contoh tampilan halaman responsif
(Sumber : <https://orangestar.com/portfolio/medpro-system>)

Ethan Marcotte dan para ahli memiliki argumen lainnya bahwa metodologi responsive design bukan sekedar melakukan perubahan pada layout saja yang sesuai dengan ukuran layar pada browser, tetapi perubahan.

5.3 HTML 5 dan CSS3

HTML 5 memberikan perubahan yang luar biasa dibandingkan dengan versi sebelumnya yaitu HMTL 4 dari perancangan halaman web, pemakaian elemen HTML 5 lebih mudah digunakan serta mudah untuk dipahami sebagai bahasa pemrograman dalam membuat aplikasi website. CSS3 (Cascading Style Sheet) memiliki keunggulan dalam membuat halaman yang responsif yaitu media *queries* dan fitur lainnya yang mendukung perancangan halaman responsif seperti *gradient*, *shadows*, animasi dan transformasi. HTML 5 dan CSS3 merupakan bagian yang terpenting dalam membuat halaman yang responsif

Prinsip dasar dalam membuat halaman yang responsif, yaitu menggabungkan *layout* yang fleksibel, *image*/gambar yang fleksibel dan CSS media *queries* yang semuanya tersedia pada HTML 5 dan CSS 3.

5.3.1 Meta tag viewport

Meta tag viewport merupakan suatu meta tag yang digunakan untuk mengatur tampilan web pada browser terbaca dengan baik dengan menyesuaikan ukuran layar yang berbeda dari suatu *device*.

```
<head>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
</head>
```

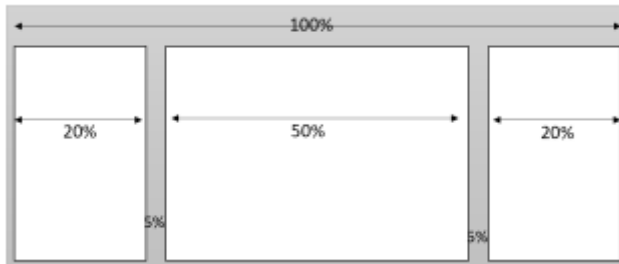
width=device-width, digunakan untuk mengatur lebar halaman akan mengikuti lebar layar perangkat yang sedang digunakan (akan bervariasi tergantung pada perangkat).

Initial-scale, digunakan untuk menentukan skala pengendalian zoom level ketika halaman web dimuat pada suatu layar perangkat.

5.3.2 Layout Fleksibel

Layout merupakan tata letak dari elemen-elemen desain terhadap suatu bidang dalam media tertentu untuk mendukung konsep atau pesan yang dibuat (Surianto Rustan, 2018)

Layout Fleksibel pada halaman web yaitu menata elemen-elemen yang akan ditampilkan dilayar browser yang dapat berubah secara otomatis ketika halaman tersebut dibuka pada ukuran layar yang berbeda atau device yang berbeda, untuk halaman responsif layout yang digunakan yaitu Fluid Layout yaitu ukuran-ukuran elemen-elemen box tidak ditulis dalam satuan pixel tetapi ditulis dengan menggunakan tanda % (persen), contoh pemakaian rancangan fluid layout pada gambar dibawah ini.



Gambar 5.2. Contoh rancangan *fluid layout*
(Sumber : <https://orangestar.com/portfolio/medpro-system>)

Penggunaan unit px tidak disarankan pada perancangan halaman web responsif karena px bersifat statis (fix), sedangkan pada halaman responsif gunakan unit relatif dengan %, maka nilai width pada elemen div akan menyesuaikan dengan ukuran layar perangkat berdasarkan *viewreport*.

```
<style>
.container {
    width:100%;
    min-height: 150px;
}
</style>
<body>
<div class='container'>
<p>Contoh lebar diatur dengan %<br> Semua elemen yang
diletakan pada div ini secara otomatis tata letaknya akan
menyesuaikan dengan viewreport</p>
</div>
```

Dengan menggunakan pengaturan width:100%, jika diterapkan pada tag <div class='container'> semua elemen

yang diletakan pada <div> tersebut elemennya akan menyesuaikan dengan lebar layar (viewreport).

Bukan hanya sekedar menentukan lebar untuk menemukan elemen, tetapi juga dipengaruhi juga dengan CSS *unit* (satuan). CSS unit terdiri dari width, margin, padding, font-size, dan yang lainnya, maka gunakan satuan yang relatif untuk menghasilkan halaman rensponsif, satuan relatif yang tercantum dalam tabel dibawah ini.

Tabel 5.1. Satuan ukur relatif

Satuan	Keterangan
em	Relatif terhadap ukuran <i>font</i> elemen (2em berarti 2 kali ukuran <i>font</i> saat ini)
ex	Relatif terhadap <i>x-height</i> dari <i>font</i> saat ini (jarang digunakan)
ch	Relatif terhadap lebar "0" (nol)
rem	Relatif terhadap ukuran font elemen root
vm	Relatif terhadap 1% dari lebar viewport
vh	Relatif terhadap 1% dari ketinggian viewport
vmin	Relatif terhadap 1% dimensi viewport yang lebih kecil
vmax	Relatif terhadap 1% dimensi viewport yang lebih besar
%	Relatif sekian % terhadap elemen induk

Sumber: <https://www.w3schools.com>

5.3.3 Image Fleksibel

Image Fleksibel merupakan pengaturan image/gambar yang ditampilkan dilayar/browser dapat menyesuaikan besar/kecilnya gambar sesuai dengan ukuran layar dari suatu

perangkat (*viewport*), sehingga halaman web menjadi responsif terhadap perubahan lebar layar.

Untuk mengatur gambar/image yang fleksibel terhadap lebar layar cukup mudah dengan mengatur dengan menggunakan satuan %.

```
<style>
.responsive {
  width: 100%;
  height: auto;
}
</style>
```

Pada contoh dibawah ini, pemakaian *image responsive* dengan pengaturan CSS lebar *image* 100% dan tinggi *auto*.



Gambar 5.3. Contoh tampilan *image Responsive*
(Sumber : https://www.w3schools.com/css/css_rwd_images.asp)

```
<style>
.responsive {
  width: 100%;
  max-width: 400px;
  height: auto;
}
</style>
```

Dari koding `max-width: 400px`, jika ukuran layar 400px keatas maka ukuran gambar tetap sesuai dengan lebar gambar asli, tetapi jika layar lebih kecil dari 400px secara otomatis gambar akan menyesuaikan ukuran 100% sesuai dengan lebar layar (*viewport*).



Gambar 5.4. Contoh tampilan *image Responsive*
(Sumber : https://www.w3schools.com/css/css_rwd_images.asp)

5.3.4 Media Query

Media Query CSS3 merupakan pengembangan dari jenis media yang ada pada CSS2, pada CSS3 lebih mengembakna kepada jenis perangkat dan kemampuan perangkat.

Media query dapat digunakan untuk memeriksa banya hal, yaitu:

1. Lebar dan tinggi *Viewreport*
2. Lebar dan tinggi perangkat
3. Orientasi (apakah tablet/ponsel dalam mode lanskap atau potret?)
4. resolusi

Aturan penulisan media query sebagai berikut:

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

Hasil query benar jika jenis media (*mediatype*), yang ditentukan sesuai dengan jenis perangkat dan semua ekspresi (*exspressions*) dalam query benar. Saat query bernilai *true* maka akan berlaku *CSS-code* dengan style sesuai dengan yang ditetapkan, jika query bernilai *false*, maka *style* akan menggunakan CSS normal.

Contoh akan memeriksa lebar layar dengan ukuran 500 pixel, jika benar lebar layar adalah 500 maka *background* layar berubah menjadi *lightgray*.

```
<!DOCTYPE html>  
<html>  
<head>  
<meta name="viewport" content="width=device-width, initial-scale=1">
```

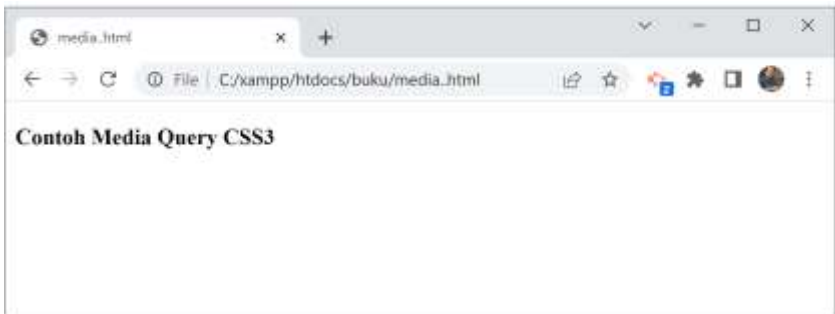
```

<style>
.smartphone {
display: none;
}
/** Media query untuk 500px **/
@media only screen and (max-width: 500px) {
body {
background-color: lightgray;

}
.smartphone {
display: block;
}
}
</style>
</head>
<body>
<h3>Contoh Media Query CSS3</h3>
<div class="smartphone">
    <h2>Ukuran layar smartphone terdeteksi 500
pixel</h2>
</div>
</body>
</html>

```

Pada gambar 5.5 ukuran layar yang digunakan normal (bukan 500 pixel), maka background berwarna putih (*default*).



Gambar 5.5. Contoh media query bernilai *false*
(Sumber : Olahan penulis)



Gambar 5.6. Contoh media query bernilai *true*
(Sumber : penulis)

Jika koding CSS yang digunakan pada media query banyak, maka koding CSS tersebut dapat disimpan pada file yang berbeda dengan file HTML (*external* CSS) pada sebuah file berakhiran .css

```
<link rel="stylesheet" type="text/css" media="screen and  
(max-width: 500px)" href="mycss.css">
```

Koding pada file HTML.

```
<!DOCTYPE html>  
<html>  
<head>  
<meta name="viewport" content="width=device-width, initial-  
scale=1">  
  
<link rel="stylesheet" type="text/css" media="screen and  
(max-width: 500px)" href="mycss.css">  
  
</head>  
<body>  
<h3>Contoh Media Query CSS3</h3>  
<div class="smartphone">  
    <h2>Ukuran layar smartphone terdeteksi 500  
pixel</h2>  
</div>  
</body>  
</html>
```

Koding pada file CSS (mycss.css)

```
<style>
    .smartphone {
        display: none;
    }

    body {
        background-color: lightgray;
    }
    .smartphone {
        display: block;
    }
</style>
```

Fitur pada CSS3 memiliki kemampuan untuk mengatur orientasi layar secara *portrait/landscape*, *min-device-width*, *device-width* untuk membuat CSS sesuai dengan kebutuhan halaman web yang responsif, berikut ukuran layar monitor dan Smartphone yang dapat diimplementasikan pada media query.

Tabel 5.2. Resolusi dan ukuran monitor dan smartphone

Resolusi	Ukuran Layar (Tinggi x Lebar)
QVGA (<i>Quarter</i> VGA)	240 x 320 piksel
HVGA (<i>Half</i> VGA)	320 x 480 piksel
FWVGA (<i>Full Wide</i> VGA)	480 x 854 piksel
qHD (<i>Quarter</i> HD)	640 x 960 piksel
HD (<i>High Definition</i>)	720 x 1280 piksel
Full HD	1920 x 1080 piksel
Full HD+	1920 x 2160 piksel

2K/QHD (<i>Quad</i> HD)	2560 x 1440 piksel
4K/UHD (<i>Ultra</i> HD)	3840 x 2160 piksel
8K/UHD	7680 x 4320 piksel

Sumber: <https://www.indonesiana.id/read/143858/macam-macam-ukuran-resolusi-layar-monitor-dan-hp>

Ukuran layar tersebut dapat diterapkan pada media query berdasarkan ukuran layar, seperti pada skrip dibawah ini.

Mendeteksi ukuran layar *portrait* dan *landscape* type HVGA (320 x 480)

```
@media only screen
    and (min-device-width : 320px)
    and (min-device-width : 480px)
{
    /* script style CSS */
}
```

Dari contoh diatas dapat diterapkan apakah hanya mendeteksi secara *portrait* atau mendeteksi layar *landscape* atau keduanya.

DAFTAR PUSTAKA

- Arkan Perdana. 2020. *Manjakan Pengunjung dengan Website Responsive Design*. <https://glints.com/id/lowongan/responsive-web-design-adalah/#.ZGSv1XZBw2w>.
- Brian Harris. 2019. *Cara Membuat Desain Web Responsive Dengan CSS*. <https://www.iltekkomputer.com/cara-membuat-desain-web-responsive-dengan-css/>
- Dimas Saputra. 2019. *Teknik Dasar Tampilan Responsif Pada Website*. <https://www.dicoding.com/blog/dasar-tampilan-responsif-pada-website/#Typography%20Units>.
- Ethan Marcotte. 2100. *Responsive Web Design*. <https://alistapart.com/article/responsive-web-design>
- Medpro System. 2023. Orange Star Design, Inc.
- Nando Rifky. 2020. *Macam-macam Ukuran Resolusi Layar Monitor*. <https://www.indonesiana.id/read/143858/macam-macam-ukuran-resolusi-layar-monitor-dan-hp>.
- Rifki Mulyawan. 2023. Memahami Pengertian Viewreport. <https://rifqimulyawan.com/blog/pengertian-viewport/>
- Statista Research Department. 2023. *Number of Smartphone users in Indonesia from 2019 to 2021 with forecasts until 2028*. <https://www.statista.com/statistics/266729/smartphone-users-in-indonesia/>
- Surianto Rustam. 2018. *Layout Dasar & Penerapannya*. Jakarta: Gramedia.
- W3schools.com. How To CSS Image Responsive. https://www.w3schools.com/howto/howto_css_image_responsive.asp
- W3school.com. Tutorial Learn Cascading Style Sheet (CSS). <https://www.w3schools.com/css/>

W3school.com. Tutorial Learn Responsive Web Design.
https://www.w3schools.com/css/css_rwd_intro.asp.

BAB 6

Pengenalan JQuery

Oleh Sitti Aisa

6.1 Pendahuluan

Bahasa pemrograman jQuery dikomandangkan pertama kali oleh seorang pria dengan nama John Resig pada tahun 2006. Saat itulah, jQuery dimantapkan menjadi salah satu *library JavaScript* yang mempunyai sifat *route platform*. Definisinya apa? Masa itu, sering terjadi tidak konsistennya antara penggunaan *JavaScript* di saluran *Firefox*, *Internet Explorer*, dan *Google Chrome*.

Di waktu lampau, saat teknologi belum bisa berkembang seperti saat ini, situasi tersebut mungkin masih dimengerti. Apalagi saat meng- *coding* bukanlah pekerjaan yang gampang.

Di sinilah belajar jQuery bisa sangat memahami para programmer. jQuery mempunyai fungsi untuk menyingkat berbagai baris *script* menjadi sebuah fungsi dapat di simpulkan, anda tidak usah lagi mengcoding ulang baris *script* agar mampu untuk mengerjakan satu tugas. jQuery mampu bertransformasi menjadi bagian dari satu *library JavaScript* yang familiar atau banyak digunakan oleh seluruh pengembang web di seluruh penjuru.

6.2 Defenisi JQuery

JQuery adalah salah satu dari berbagai macam jenis *JavaScript library*, dimana didalamnya sekumpulan fungsi dari *JavaScript* yang bisa langsung digunakan, sehingga bisa membantu mempercepat dalam membuat *script* dari *javascript*. Dengan menggunakan jQuery, kodingan javascript yang

panjang dapat disingkat menjadi beberapa baris kode saja.(Abdulloh, 2018).

Istilah JQuery mungkin belum familiar terdengar ditelinga kita, yang biasa terdengar istilah Java script kan. istilah dari JQuery pertama kali terdengar pada tahun 2006 yang diperkenalkan oleh John Resign yang di maksudkan untuk memberi kemudahan kepada programmer untuk menggunakan java script terhadap website yang akan dikembangkan. Tren di kalangan pada developer karena bersifat bisa berpindah antar platform.

Sebelum munculnya JQuery, para programmer mengalami kendala ketika membangun sebuah website yaitu saat mengimplementasikan java script pada beberapa web browser terjadi ketidakkonsisten. (Abdulloh, 2018)

6.2.1 Ketangguhan jQuery

Dalam membentuk sebuah aplikasi berbasis situs, jQuery sangat memudahkan dalam menciptakan desain halaman situs yang matang. Adapun beberapa point ketangguhan dari jQuery dalam *web* ;

1. Menyalahgunakan item pada HTML
2. Menyalahgunakan CSS
3. Pengurusan event pada item HTML
4. Menghasilkan efek dan animasi
5. Melancarkan Penggunaan AJAX
6. Meringankan *syntax JavaScript* lainnya.(Abdulloh, 2018)

6.2.2 Keistimewaan jQuery

Jika ditandingkan dengan pesaingnya antara lain Prototype, Mootools, YUI, dan library JavaScript lainnya, jQuery memiliki kelebihan tersendiri. Berikut beberapa alasan mengapa harus memilih jQuery dibandingkan yang lainnya :

1. jQuery telah dimanfaatkan oleh situs - situs yang ada di seluruh dunia

2. Kompatibel dengan semua aplikasi pencarian yang biasa dimanfaatkan seperti *mozilla firefox*, *internet explorer*, *safari*, *google chrome*, dan *Opera*.
3. Memadai digunakan untuk semua model CSS.
4. Penyimpanan, tutorial dan contoh – contoh lengkap.
5. Didukung oleh komunitas yang besar dan aktif.
6. Ketersediaan plugin yang cukup banyak jumlahnya.
7. Filenya hanya satu dan ukurannya kecil, sehingga cepat aksesnya. (Rohingun, 2015)
8. *Open Source* (gratis) dengan lisensi dari GNU General Public License.

6.2.4 Cara Memperoleh Library jQuery

Untuk memperoleh library jQuery bisa di download dari website resminya yaitu jquery.com/download. Terdapat skrip untuk mengembangkan yang lebih spesifik untuk para pengembang aplikasi dan skrip yang telah di modifikasi sehingga sizenya jauh lebih kecil yang lebih dikhususkan untuk para pengguna. Skrip yang telah diperbaharui biasanya ditambah kata ‘.min’ dibelakang namanya.

6.2 Cara Kerja JQuery

Cara penggunaan jQuery sangat mudah. Berikut ini langkah – langkah menggunakan jQuery dalam website :

1. Simpan jQuery dalam sebuah folder yang diinginkan. Biasanya pada website diletakkan pada folder script atau js.
Buat file HTML dalam folder yang sama atau dalam folder yang lain kemudian panggil jQuery di antara tag <head> dan </head>. Contoh penggunaan jQuery dapat dilihat pada kodingan dibawah ini :

```

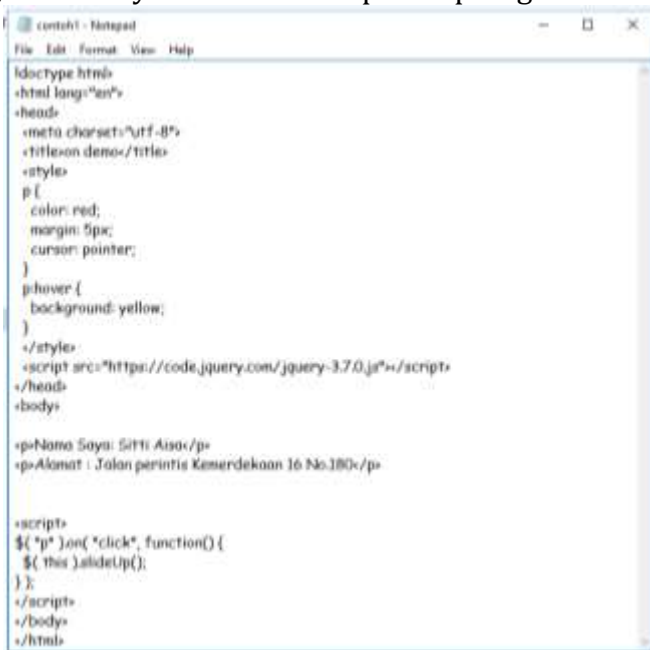
<html lang="en">
  <head>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
  </head>
  <body>
</script>

  </body>
</html>

```

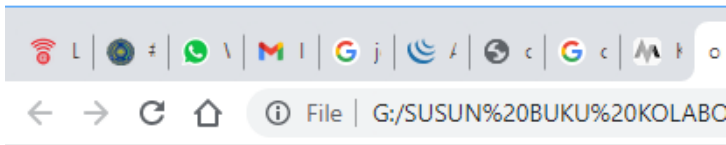
Sumber : <http://api.jquery.com/click/>

2. Selanjutnya, tambahkan perintah diantara tag <body> dan </body> misalnya untuk menampilkan paragraf.



Gambar 6.1. Script jQuery

Sumber : <http://api.jquery.com/click/>



Nama Saya: Sitti Aisa

Alamat : Jalan perintis Kemerdekaan 16 No.180

Gambar 6.2. Hasil Program

6.3 Menggunakan Selector

Selector didefinisikan sebagai salah satu *script* untuk menyeleksi suatu item HTML yang akan diubah dengan *script* jQuery. Bentuk umum penulisan dari Selector pada jQuery dapat dituliskan di dalam kurung setelah tanda dolar. Format penulisan selector adalah sebagai berikut :

`$(‘selectors’);`(Rohingun, 2015)

6.3.1 Basic Selector

Untuk jenis selector ini digunakan untuk selector dasar sesuai yang digunakan pada perintah CSS.

Tabel 6.1. Perintah Dalam Selector

Selector	Penjelasan
Selector All	Memilah semua item HTML, dan memberi tanda dengan (*).
Selector Tag	Memilah tag HTML dengan cara menuliskan nama tagnya langsung. Misalnya, memilih tag <p> maka dituliskannya menjadi (p).
Selector Class	Memilah tag HTML yang memiliki atribut class tertentu. Cara penulisannya dengan menaruh tanda titik (.) sebelum nama class. Seperti memilih tag <p class=“kuning”>, maka penulisan selector menjadi \$(‘.kuning’) atau \$(‘p.kuning’).

Selector	Penjelasan
Selector id	Memilih tag HTML yang memiliki atribut id tertentu. Cara penulisannya dengan menaruh tanda pagar(#) sebelum nama id. Misalnya, memilih tag <p id="abuabu">, maka penulisan selector menjadi. \$('#abuabu').
Multiple Selector	Memilih elemen HTML dengan beberapa selector sekaligus dimana masing – masing selector dipisahkan dengan tanda koma (,).

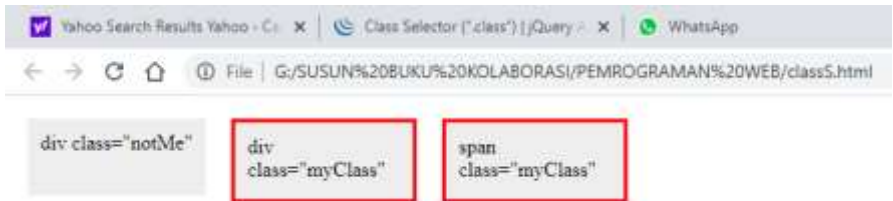
Sumber : (Abdulloh, 2018)

```

1 | <!doctype html>
2 | <html lang="en">
3 | <head>
4 |   <meta charset="utf-8">
5 |   <title>class democ</title>
6 |   <style>
7 |     div, span {
8 |       width: 120px;
9 |       height: 40px;
10 |      float: left;
11 |      padding: 10px;
12 |      margin: 10px;
13 |      background-color: #EEEEEE;
14 |    }
15 |   </style>
16 |   <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
17 | </head>
18 | <body>
19 |
20 | <div class="notMe">div class="notMe"</div>
21 | <div class="myClass">div class="myClass"</div>
22 | <span class="myClass">span class="myClass"</span>
23 |
24 | <script>
25 | $( ".myClass" ).css( "border", "1px solid red" );
26 | </script>
27 |
28 | </body>
29 | </html>

```

Gambar 6.3. *Syntax Untuk Selector Class*



Gambar 6.4. Hasil Program Dari Selector *Class*

6.3.2 Attribute Selector

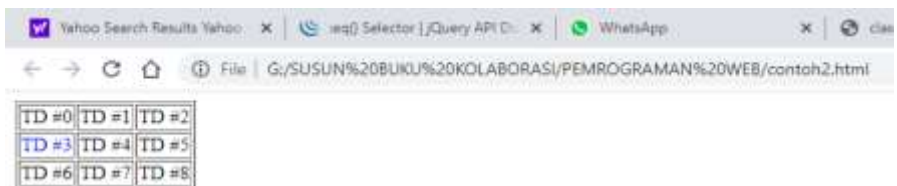
Selector ini ditulis dengan menyebutkan Nama tag diikuti Nama atribut yang diletakkan di dalam tanda kurung siku ([...]). Pada umumnya, antara atribut dan nilainya dipisahkan sama dengan (=). Salah satu contohnya dituliskan dengan perintah [:eq(index)].(Abdulloh, 2018)

```

1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>eq demo</title>
6   <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
7 </head>
8 <body>
9
10 <table border="1">
11   <tr><td>TD #0</td><td>TD #1</td><td>TD #2</td></tr>
12   <tr><td>TD #3</td><td>TD #4</td><td>TD #5</td></tr>
13   <tr><td>TD #6</td><td>TD #7</td><td>TD #8</td></tr>
14 </table>
15
16 <script>
17 $( "td:eq( 2 )" ).css( "color", "red" );
18 </script>
19
20 </body>
21 </html>

```

Gambar 6.5. *syntax* untuk perintah :eq



Gambar 6.6. Hasil Program

6.3.3 Filter Selector

Selector ini digunakan untuk memilih elemen dengan keadaan tertentu menggunakan basic selector diikuti Nama filter. Format penulisannya dengan menyebutkan basic selector diikuti Nama filter yang diawali dengan tanda titik(':').

6.3.4 Form Selector

Selector ini dikhususkan untuk memilih elemen didalam form dan penulisannya sama dengan filter selector.

6.3.5 Hierarchy Selector

Untuk selector ini digunakan untuk memilih suatu elemen pada posisi tertentu terhadap elemen lain.

6.4 Memahami Effect

Sebuah perintah effect dimaksudkan sebagai salah satu bagian method yang ada pada jQuery yang difungsikan untuk memanipulasi item HTML yang telah diseleksi oleh bagian selector misalnya, menyembunyikan, menampilkan, menganimasikan, dan sebagainya. Perintah effect dituliskan sebagai sebuah selector, yang dipisahkan dengan tanda titik (.) dan diikuti tanda kurung. Tanda kurung bisa diisi dengan pengaturan effect tersebut atau bisa dikosongkan. Cara

```
$ ('selector').variabeleffect()
```

Salah satu perintah dalam effect yang sering digunakan adalah *clearqueue* dimana digunakan untuk menghapus antrian gambar yang bergerak yang dirancang dengan menggunakan perintah *animate()*. Berikut contoh *syntax* dalam menggunakan perintah *clearqueue*. Gambar 6.7 dibawah ini memperlihatkan sebuah button start dan stop dimana ketika ditekan tombol

start maka kotak berwarna pink akan bergerak tetapi ketika tombol stop ditekan maka akan berhenti.



```
<script>
$( "#start" ).on( "click", function() {
    var myDiv = $( "div" );
    myDiv.show( "slow" );
    myDiv.animate({
        left:"+=200"
    }, 5000 );
    myDiv.queue(function() {
        var that = $( this );
        that.addClass( "newcolor" );
        that.dequeue();
    });
    myDiv.animate({
        left:"-200"
    }, 1500 );
    myDiv.queue(function() {
        var that = $( this );
        that.removeClass( "newcolor" );
        that.dequeue();
    });
    myDiv.slideUp();
});
$( "#stop" ).on( "click", function() {
    var myDiv = $( "div" );
    myDiv.clearQueue();
    myDiv.stop();
}); </script>
```

6.5 Pemahaman Fungsi Event

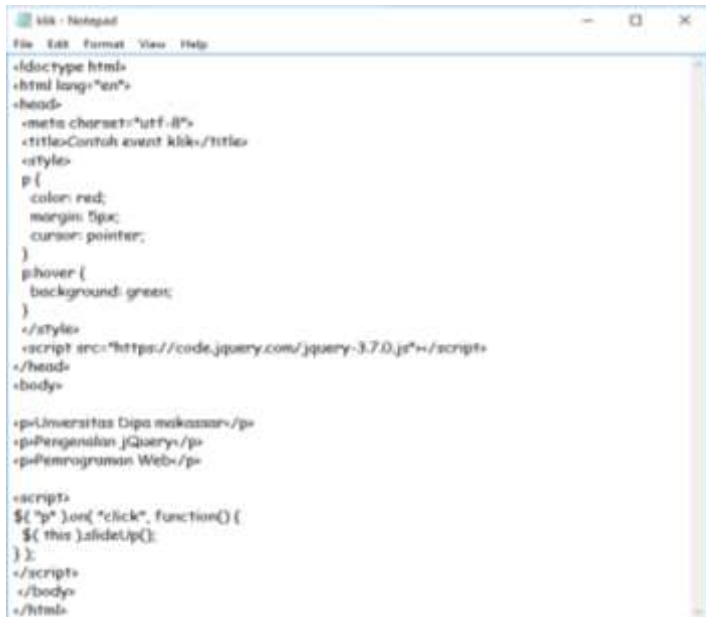
Maksud dari penggunaan event ialah *script* yang mampu memantau sebuah tindakan dari pengguna, seperti jika pengguna memencet mouse, memindahkan mouse, dan lainnya. Format penggunaan event hamper mirip dengan penggunaan effect, tetapi dalam tanda kurung yang ditulis setelah event biasanya terdapat fungsi yang akan dijalankan. Bentuk penulisan event adalah :

```
$(‘selector’).event (fungsi){  
    };
```

Adapun beberapa perintah pada event :

1. Event click (). Dimana event atau method terjadi ketika sebuah element di klik.
2. Event dblclick (). Dimana event atau method terjadi ketika sebuah element diklik dua kali.
3. Event mouseleave(). Dimana event atau method saat kursor meninggalkan element yang ada.
4. Event mouseenter(). Dimana event atau method saat kursor di simpan atau memasuki element tertentu.

Salah satu contoh penggunaan event click

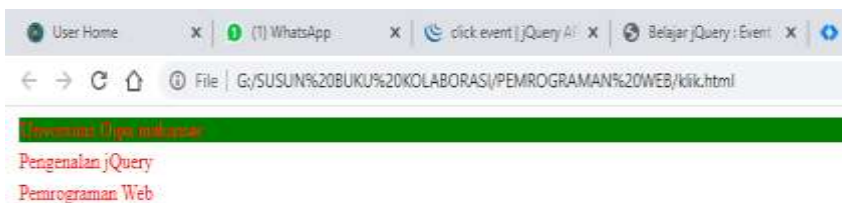


```
<code><!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Contoh event klik</title>
  <style>
    p {
      color: red;
      margin: 5px;
      cursor: pointer;
    }
    p:hover {
      background: green;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.7.0.js"></script>
</head>
<body>

  <p>Universitas Dipa makassar</p>
  <p>Pengenal jQuery</p>
  <p>Pemrograman Web</p>

  <script>
    $( "p" ).on( "click", function() {
      $( this ).slideUp();
    } );
  </script>
</body>
</html></code>
```

Hasil programnya seperti gambar 6.7 dibawah ini menunjukkan fungsi event click dimana ketika di klik maka paragraph pertama akan hilang.



Gambar 6.7. Hasil program dari event click

DAFTAR PUSTAKA

- Abdulloh, R. 2018. *7 in 1 Pemrograman Web untuk Pemula*. Elex Media Komputindo.
- Rohingun. 2015. *Smart Trik JQuery without Plugin*. Edited by Ignas. yogyakarta: Penerbit ANDI OFFSET.

BAB 7

INTERAKSI DENGAN SERVER MELALUI AJAX

Oleh Wirawan Istiono

7.1 Apa itu Ajax

Ajax adalah singkatan dari pendekatan baru yang kuat untuk membuat aplikasi web. Itu menghilangkan metode konvensional untuk membagi aplikasi web menjadi berbagai halaman yang perlu dimuat dan dimuat ulang berulang kali. Alih-alih, Ajax memungkinkan pembuatan aplikasi yang memperbarui konten halaman secara dinamis tanpa memerlukan pemuatan ulang halaman dengan menarik data dari server web di latar belakang (T.Holdener, 2008).

Pengalaman pengguna yang lebih cepat dan mirip desktop adalah hasil akhirnya. Bab ini memberikan gambaran umum tentang Ajax dan berbagai teknologi yang digunakannya untuk memberi pengembang web sumber daya yang mereka butuhkan untuk menghasilkan generasi baru aplikasi yang kuat, cepat, dan responsif. Bab ini juga akan memberikan ikhtisar tentang berbagai aplikasi Ajax, memberikan contoh jenis tugas yang dapat diselesaikan Ajax (T.Holdener, 2008). AJAX adalah metode pengembangan web yang digunakan untuk membangun aplikasi web interaktif. jika Anda terbiasa dengan XML, HTML, CSS, dan JavaScript.

Ajax adalah istilah yang digunakan untuk menggambarkan sekelompok teknologi web yang terhubung. Ia juga dikenal sebagai Asynchronous JavaScript dan XML. Dengan bantuan teknologi ini, desainer web dapat membangun aplikasi web yang dapat berkomunikasi dengan pengguna secara real

time dan secara diam-diam mengakses data dari server web. Informasi tersebut kemudian dapat dilihat di browser tanpa harus me-refresh halaman. Hasil akhirnya adalah pembuatan aplikasi web dengan antarmuka pengguna seperti desktop (Tutorialspoint, n.d.).

Aplikasi yang dibangun dengan Ajax lebih cepat dan lebih responsif daripada yang dibangun dengan teknologi web lainnya. Memodifikasi aplikasi web untuk bertukar lebih sedikit byte data dengan server web akan meningkatkan kinerja. Akibatnya, setiap kali pengguna meminta sesuatu dari aplikasi atau berinteraksi dengannya, server web tidak harus membuat dan mengirimkan seluruh halaman web. Penyegaran halaman web tidak diperlukan karena hanya sejumlah kecil data yang dipertukarkan menggantikan halaman web (Ford, 2009).

Mengirim data secara asinkron antara browser web dan server web dimungkinkan oleh Ajax. Ini menyiratkan bahwa aplikasi Ajax dapat memproses data yang diminta setiap kali dikembalikan dan mengirimkan permintaan ke server web tanpa harus menunda eksekusi aplikasi. Aplikasi Ajax mengirim permintaan ke server web menggunakan objek browser unik yang disebut objek XMLHttpRequest daripada mengirim data ke server melalui formulir. Elemen penting dari Ajax yang memungkinkan komunikasi asinkron adalah objek ini (Brinzarea-Iamandi, Darie, Hendrix, & Audra, 2009).

7.2 Mengapa Mempelajari AJAX?

JavaScript asinkron dan XML disebut sebagai AJAX. Dengan bantuan XML, HTML, CSS, dan Java Script, sebuah metode baru yang dikenal dengan AJAX memungkinkan pengembang membangun aplikasi web yang lebih efektif, cepat, dan interaktif (Brinzarea-Iamandi *et al.*, 2009).

1. Ajax menggunakan Document Object Model dan JavaScript untuk tampilan konten dinamis, bersama dengan XHTML untuk konten dan CSS untuk presentasi.
2. Aplikasi web tradisional menggunakan permintaan sinkron untuk mengirim dan menerima data ke dan dari server. Ini menandakan bahwa setelah mengirimkan formulir, Anda akan dibawa ke halaman baru yang berisi data baru dari server.
3. Saat Anda mengirimkan formulir menggunakan AJAX, JavaScript akan mengirimkan permintaan ke server, menganalisis respons, dan memodifikasi layar saat ini. Pengguna tidak akan, dalam arti yang paling ketat, menyadari bahwa apapun bahkan dikirim ke server.
4. Format apa pun, termasuk teks biasa, dapat digunakan untuk menerima data server; namun, XML sering digunakan.
5. AJAX adalah fitur browser web yang berjalan secara independen dari server web.
6. Program klien dapat terus berjalan di latar belakang, meminta data dari server saat pengguna menggunakan aplikasi.
7. interaksi pengguna yang alami dan intuitif. Pergerakan mouse saja dapat berfungsi sebagai pemicu peristiwa; mengklik tidak diperlukan.
8. bukan berbasis halaman, berbasis data.

Teknologi Rich Internet Application (RIA) yang saat ini paling praktis adalah AJAX. Industri ini mendapatkan banyak momentum, dan sejumlah perangkat serta kerangka kerja bermunculan. Namun, AJAX tidak didukung oleh semua browser dan didukung oleh JavaScript, yang sulit untuk dipelihara dan di-debug (Ford, 2009).

7.3 Teknologi AJAX

AJAX tidak dapat berfungsi sendiri. Untuk membuat halaman web interaktif, itu digabungkan dengan teknologi lain (T.Holdener, 2008).

1. JavaScript
Bahasa skrip informal.
Saat suatu peristiwa terjadi di halaman, JavaScript dipanggil.
Dasar dari operasi AJAX.
2. DOM
Akses dan manipulasi dokumen terstruktur menggunakan DOM API.
merepresentasikan bagaimana dokumen XML dan HTML disusun.
3. CSS
Objek JavaScript XMLHttpRequest JavaScript, yang terlibat dalam komunikasi asinkron dengan server, dapat mengubah gaya presentasi halaman secara terprogram berkat CSS, yang memungkinkan pemisahan yang jelas antara gaya presentasi dan konten.

7.4 Web Terkenal dengan Ajax

Berikut adalah daftar beberapa situs terkenal yang menggunakan AJAX (Brinzarea-Iamandi et al., 2009).

1. Google Map
Alih-alih mengklik tombol, pengguna dapat menyeret seluruh peta dengan menggunakan mouse.
2. Google Suggest
Google menawarkan saran saat Anda mengetik. Untuk menelusuri hasil, tekan tombol panah.

3. Gmail

Gmail adalah email web yang dirancang dengan gagasan bahwa email bisa lebih mudah digunakan, efektif, dan praktis.

4. Yahoo Mail

Sekarang menuju ke yahoo mail akan lebih menyenangkan dan lebih sederhana.

7.4.1 Dukungan Peramban

AJAX tidak kompatibel dengan setiap browser yang sedang digunakan. Browser paling populer yang mendukung AJAX tercantum di bawah ini (Tutorialspoint, n.d.).

1. Mozilla Firefox versi 1.0 dan yang lebih baru.
2. Versi Netscape 7.1 dan lebih tinggi.
3. Apple Safari versi 1.2 dan yang lebih baru.
4. Diperlukan Internet Explorer 5 dan yang lebih baru.
5. Konqueror.
6. Opera versi 7.6 dan lebih tinggi.

Pertimbangkan browser yang tidak mendukung AJAX saat membuat aplikasi Anda berikutnya. Jika browser dikatakan tidak mendukung AJAX, itu berarti browser tidak mendukung pembuatan objek Javascript, khususnya objek XMLHttpRequest.

7.4.2 Langkah Operasi AJAX

1. Ada event dari sisi klien.
2. Pembuatan objek XMLHttpRequest.
3. Objek untuk XMLHttpRequest telah disiapkan.
4. Webserver menerima permintaan asinkron dari objek XMLHttpRequest.
5. Webserver mengirimkan kembali hasil yang menyertakan dokumen XML.

6. Fungsi `callback()` dipanggil oleh objek `XMLHttpRequest`, yang kemudian memproses hasilnya.
7. DOM HTML telah dimodifikasi.

7.4.3 AJAX - XMLHttpRequest

Fondasi AJAX adalah objek `XMLHttpRequest`. Meskipun telah dapat diakses sejak Juli 2000 rilis Internet Explorer 5.5, itu tidak sepenuhnya dipahami sampai munculnya AJAX dan Web 2.0 pada tahun 2005.

JavaScript, JScript, VBScript, dan bahasa skrip browser web lainnya dapat menggunakan API `XMLHttpRequest` (XHR) untuk mentransfer dan memanipulasi data XML ke dan dari server web menggunakan HTTP, membuat saluran koneksi terpisah antara Sisi Klien halaman web dan Sisi Server.

Panggilan ke `XMLHttpRequest` akan sering mengembalikan data dari database back-end. `XMLHttpRequest` dapat digunakan untuk mengambil data dalam format selain XML, seperti JSON atau bahkan teks biasa.

Beberapa teknik dan atribut yang perlu Anda pelajari tercantum di bawah ini (Ford, 2009).

1. `abort()`
Membatalkan permintaan saat ini.
2. `getAllResponseHeaders()`
Mengembalikan set lengkap header HTTP sebagai string.
3. `getResponseHeader(namatajuk)`
mengembalikan nilai tajuk HTTP yang ditentukan.
4. `open (metode, URL)`
5. `open (metode, URL, async)`
6. `open (metode, URL, async, username)`
7. `open (metode, URL, async, username, password)`
menentukan metode permintaan, URL, dan atribut opsional tambahan.
"GET", "POST", atau "HEAD" adalah nilai yang mungkin untuk parameter metode. Dimungkinkan untuk

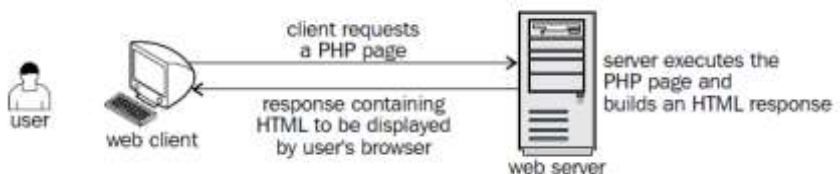
menggunakan metode HTTP tambahan, seperti perintah "PUT" dan "DELETE" (yang terutama digunakan dalam aplikasi REST).

Apakah permintaan harus diproses secara asinkron atau tidak ditentukan oleh parameter "async". "true" menunjukkan bahwa pemrosesan skrip berlanjut setelah metode send() tanpa berhenti untuk menunggu respons, dan "false" menunjukkan bahwa pemrosesan skrip berhenti untuk menunggu respons.

8. send(content)
Mengirim permintaan.
9. setRequestHeader(label, value)
Menambahkan pasangan label/nilai ke header HTTP untuk dikirim.

7.5 Ajax dan teknologi sisi server

logika sisi server, yang mencakup bahasa seperti PHP, ASP.NET, Java Server Pages (JSP), Perl, ColdFusion, Ruby on Rails, dan lain-lain. Masing-masing memiliki kelebihan dan kekurangan. Kami telah memilih PHP, bahasa scripting sumber terbuka yang menawarkan platform pengembangan yang andal dan disukai, untuk implementasi sisi server. Server menjalankan kode halaman PHP dan mengembalikan hasilnya daripada mengembalikan halaman statis. (Hasil ini tetap harus disajikan dalam HTML atau format lain yang sudah dikenal klien)

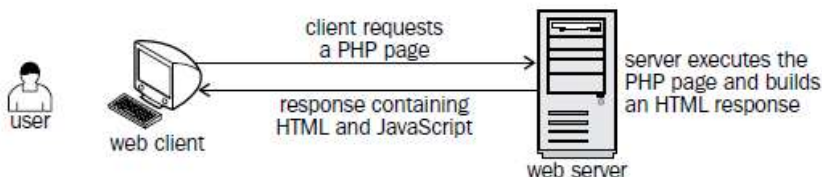


Gambar 7.1. Permintaan klien untuk halaman PHP
(Sumber: Brinzarea-Iamandiet *al.*, 2009)

Tetapi bahkan dengan PHP membangun respons berbasis database yang unik secara dinamis di server, klien masih menunjukkan dokumen web yang statis, monoton, dan (yawn) tidak terlalu cerdas.

Mengapa ada orang yang menginginkan sesuatu yang baru ketika ada begitu banyak pilihan untuk membuat fitur yang kuat di dalam peramban web? Apa yang kurang? Teknologi hadir untuk memenuhi kebutuhan pasar saat ini, seperti yang telah disebutkan di awal bab. Tanpa menggunakan Flash, applet Java, atau teknologi lain yang dianggap terlalu mencolok atau berat untuk tujuan tertentu, sebagian pasar ingin menyediakan fungsionalitas yang lebih kuat kepada klien web. Contohnya adalah validasi formulir interaktif, di mana data pengunjung harus dibandingkan dengan aturan validasi sisi server untuk memastikan kepatuhan (Brinzarea-Iamandi et al., 2009).

Dalam kasus seperti itu, pemrogram menggunakan HTML, JavaScript, dan PHP (atau teknologi sisi server lainnya) untuk membuat situs web dan aplikasi web. Gambar 7.2 menggambarkan permintaan tipikal dalam skenario ini. Ini menampilkan permintaan HTTP bersama dengan respons HTML dan JavaScript yang dihasilkan secara terprogram oleh PHP.



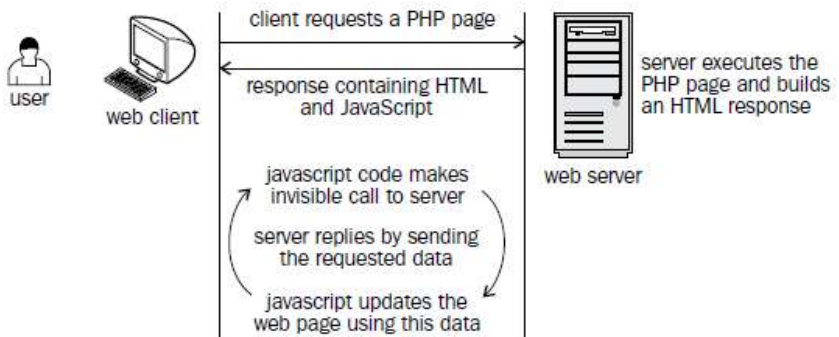
Gambar 7.2. HTTP, HTML, ASP.NET, dan JavaScript beraksi
(Sumber: Brinzarea-Iamandiet *al.*, 2009)

Masalah yang tidak diperhatikan dengan situasi ini adalah bahwa setiap kali klien meminta informasi baru dari

server, permintaan HTTP baru dibuat, yang menghentikan aktivitas pengguna. Dalam skenario modern, pemuatan ulang halaman adalah cara baru, dan AJAX menyelamatkan kita.

7.6 Dunia AJAX

JavaScript dan XML asinkron dikenal sebagai AJAX. Asinkron adalah komponen penting di sini. Sederhananya, AJAX menyediakan metode untuk memperbarui bagian tertentu dari halaman tanpa harus memuat ulang halaman lengkap dengan menggunakan JavaScript untuk melakukan panggilan server latar belakang dan mengambil data tambahan sesuai kebutuhan. Gambar 7.3 memberikan representasi visual tentang apa yang terjadi ketika pengunjung meminta halaman web berkemampuan AJAX.



Gambar 7.3. Contoh panggilan AJAX
(Sumber: Brinzarea-Iamandiet *al.*, 2009)

Dengan mengaktifkan komunikasi latar belakang antara klien dan server saat pengguna berinteraksi dengan halaman, AJAX menghilangkan ketidakseimbangan antara keduanya. Pikirkan formulir web di mana pengguna diminta untuk memasukkan informasi (seperti nama, alamat email, kata sandi, nomor kartu kredit, dll.) Yang perlu diverifikasi sebelum

melanjutkan ke langkah proses pendaftaran berikutnya. Ada tiga cara yang bisa Anda lakukan untuk melakukan ini (Tutorialspoint, n.d.):

1. Biarkan pengguna mengisi semua informasi yang diperlukan, kirimkan halaman, dan server akan menangani validasi setelah itu. Jika validasi gagal, server mengirimkan kembali formulir—yang terkadang kosong—dan meminta pengguna mengubah informasi yang salah. Dalam hal ini, terdapat penundaan (*dead time*) antara pengguna mengirimkan dan menunggu tanggapan.
2. Gunakan JavaScript untuk melakukan validasi sisi klien. Sebelum mengirimkan formulir, pengguna diperingatkan tentang data yang salah dan diberi kesempatan untuk melakukan koreksi. Metode ini hanya efektif untuk validasi yang sangat mendasar yang tidak membutuhkan data server tambahan. Metode ini juga gagal saat algoritme validasi bersifat rahasia atau hak milik dan tidak dapat ditransfer ke klien sebagai kode JavaScript.
3. Manfaatkan validasi formulir AJAX untuk memungkinkan aplikasi web memeriksa data yang dimasukkan saat pengguna mengisi formulir. Daftar kota yang diawali dengan huruf tersebut, misalnya, dapat dimuat "*on-the-fly*" oleh web browser setelah pengguna mengetikkan huruf pertama dari kota tersebut.

7.7 Mengubah Aplikasi HTML menjadi Aplikasi Ajax

Sekarang aplikasi *Joke of the Day* beroperasi dalam HTML, Anda dapat mengubahnya menjadi aplikasi Ajax. Buka aplikasi index.html di editor teks atau kode pilihan Anda untuk melakukan ini, lalu ubah seperti yang ditunjukkan di sini.

```
<HTML>
<HEAD>
```

```

<TITLE>Ajax Joke of the Day Application</TITLE>
<SCRIPT language = "javascript" type = "text/javascript">
var Request = false;
if (window.XMLHttpRequest) {
    Request = new XMLHttpRequest();
} else if (window.ActiveXObject) {
    Request = new ActiveXObject("Microsoft.XMLHTTP");
}
function retrieveJoke(url, elementID) {
    if(Request) {
        var RequestObj = document.getElementById(elementID);
        Request.open("GET", url);
        Request.onreadystatechange = function()
        {
            if (Request.readyState == 4 && Request.status == 200)
            {
                RequestObj.innerHTML = Request.responseText;
            }
        }
        Request.send(null);
    }
}
</SCRIPT>
</HEAD>
<BODY>
<H1>Where do bees go when they get married?</H1>
<FORM>
    <INPUT type = "button" value = "Fetch Answer"
    onclick = "retrieveJoke('joke.txt', 'DivTarget')">
</FORM>
<DIV id="DivTarget"> </DIV>
</BODY>
</HTML>

```

Seperti yang Anda lihat, ada banyak pernyataan baru yang ditambahkan ke aplikasi Joke of the Day untuk versi Ajax. Bab ini tidak akan menjelaskan secara mendetail tentang apa yang dilakukan oleh setiap pernyataan baru di index.html karena bagian buku tersebut belum membahas kerumitan dalam mengembangkan aplikasi Ajax. Sebagai gantinya, mari pertahankan tingkat aktivitas yang cukup tinggi (Ford, 2009).

7.8 Properti XMLHttpRequest

Anda perlu bekerja dengan berbagai properti XMLHttpRequest untuk menentukan jenis data yang diantisipasi oleh aplikasi Ajax Anda untuk menerima dari server web serta untuk memantau dan bereaksi terhadap data tersebut setelah diterima dari server web. Anda harus terbiasa dengan berbagai properti XMLHttpRequest yang tercantum dalam Tabel 7.1 untuk menggunakannya dalam aplikasi Ajax Anda.

Tabel 7.1. Atribut Objek XMLHttpRequest

Deskripsi Atribut	Deskripsi Atribut
readyState	angka antara 0 dan 4 yang mewakili status penerimaan data server web; nilai 4 menunjukkan bahwa data telah diterima.
Status	Kode status keseluruhan untuk permintaan data yang dibuat oleh objek XMLHttpRequest, dengan nilai 200 berarti semuanya berjalan lancar
responseText	menyimpan data teks yang dikembalikan oleh server web.
responseXML	Data XML yang dikembalikan oleh server web disimpan di sini.
onreadystatechange	memanggil fungsi untuk dijalankan setiap kali event readystatechange berlangsung.

Sumber: Ford, 2009

Proses multi-langkah digunakan untuk terhubung ke server web menggunakan objek XMLHttpRequest, menentukan permintaan data, lalu menangani data yang dikembalikan. Bagian-bagian berikut menjelaskan secara mendetail tentang betapa pentingnya memahami bagaimana langkah-langkah ini berfungsi (Ford, 2009).

Pembuatan objek XMLHttpRequest adalah langkah pertama dalam menggunakannya. Sayangnya, objek XMLHttpRequest adalah area lain di mana browser web berbeda. Bagaimana cara mengetahui apakah aplikasi Ajax berjalan di Firefox, Safari, Opera, atau versi Internet Explorer apa pun setelah 6 ditampilkan dalam pernyataan berikut.

```
var Request = false;
if (window.XMLHttpRequest) {
    Request = new XMLHttpRequest();
}
```

Objek XMLHttpRequest akan direferensikan menggunakan variabel Request, yang didefinisikan dalam pernyataan pertama. Pernyataan kedua menguji properti XMLHttpRequest dari objek jendela untuk melihat apakah didukung. Jika demikian, pernyataan kedua menggunakan kata kunci baru untuk membuat objek XMLHttpRequest baru dengan nama Request (Ford, 2009).

7.9 Membangun Kembali Koneksi

Setelah membuat objek XMLHttpRequest, tindakan selanjutnya adalah menggunakan metode open() untuk terhubung ke server web. Sintaks untuk metode ini adalah sebagai berikut:

```
open("method", "url" [, asyncFlag [, username [, password]]])
```

Fungsi masing-masing argumen untuk metode `open()` tercantum dalam Tabel 7.2.

Tabel 7.2. Parameter metode `OPEN()` objek `XMLHTTPREQUEST`

Nilai	Penjelasan
method	Metode HTTP (GET, POST) yang harus digunakan untuk terhubung ke server web
url	Alamat situs web tempat file akan dibuka
asyncFlag	Menyetel nilai ini ke true (default) membuat koneksi asinkron; menyetelnya ke false (default) membuat koneksi sinkron.
userName	nama pengguna yang mungkin diminta oleh server web dari Anda.
password	kata sandi yang mungkin diminta oleh server web.

Sumber: Ford, 2009

Anda harus memilih antara opsi HTML GET dan POST yang umum saat menggunakan metode `open()` untuk membuka URL di server web. Opsi POST biasanya digunakan saat Anda perlu mengunduh data dalam jumlah besar dari server web, sedangkan opsi GET, seperti yang ditunjukkan di bawah ini, adalah semua yang biasanya digunakan untuk mengambil data dalam jumlah sedang dari server web (Ford, 2009).

```
if(Request) {  
    Request.open("GET", "url");  
    //your other statement  
}
```

Objek XMLHttpRequest disiapkan untuk bekerja dengan file tertentu menggunakan opsi GET dan POST. Namun, tidak ada pilihan yang membuat aplikasi mengakses file atau membuat koneksi ke server web. Beberapa langkah lagi akan mengajari Anda cara melakukannya. Perhatikan bahwa contoh ini memverifikasi bahwa objek XMLHttpRequest yang sebenarnya telah berhasil dibuat instance-nya (dengan memastikan bahwa nilai yang ditetapkan ke variabel objek Request sama dengan true) sebelum memanggil metode open().

Ada dua opsi yang tersedia saat menentukan jalur dan nama file untuk aplikasi Ajax: jalur absolut dan jalur relatif. Jalur absolut, seperti yang ditampilkan di sini, menentukan lokasi persis file.

```
Request.open("GET", "http://www.somewebsite/Scores.txt");
```

Server web yang menghosting file Scores.txt di <http://www.somewebsite.com/Test/> telah disebutkan dalam contoh ini. Di sisi lain, jalur relatif menunjukkan lokasi file berdasarkan lokasinya di server web terkait dengan file aplikasi. Misalnya, pernyataan berikut mencari file Scores.txt yang ada di folder yang sama dengan aplikasi (Ford, 2009).

```
Request.open("GET", "Scores.txt");
```

Logika pemrograman yang diperlukan untuk menangani respons server web harus ditambahkan setelah objek XMLHttpRequest dibuat dan metode open() telah digunakan untuk menentukan file yang ingin Anda akses. Transisi objek XMLHttpRequest melalui berbagai status saat mengirim permintaan HTTP, menunggu respons dari server web, dan akhirnya menerima data dari server. Dua properti spesifik yang

diekspos oleh objek XMLHttpRequest memberikan informasi tentang status umum permintaan HTTP. Properti readyState dan status adalah yang dimaksud.

Anda dapat mengetahui kapan aplikasi selesai menerima data secara terprogram dengan memperhatikan nilai properti readyState. Setiap nilai yang tercantum dalam Tabel 7.3 dapat dilaporkan dari readyState property (Ford, 2009).

Tabel 7.3. XMLHTTPREQUEST Nilai atribut status objek

Nilai	Penjelasan
0	mewakili keadaan di mana objek XMLHttpRequest belum diinisialisasi tetapi telah dibuat.
1	mewakili status "sent", di mana metode sent() dari objek XMLHttpRequest belum dieksekusi tetapi metode open() telah dieksekusi.
2	mewakili objek XMLHttpRequest dalam status "sent", di mana ia menunggu data dikembalikan.
3	menggambarkan tahap di mana objek XMLHttpRequest berusaha mendapatkan informasi dari server web.
4	menandakan bahwa data web server telah diterima sepenuhnya oleh objek XMLHttpRequest.

Sumber: Ford, 2009

```
Request.onreadystatechange = function() {
  if (Request.readyState == 4 && Request.status == 200) {
    //your statement
  }
}
```

Fungsi anonim adalah nama yang diberikan untuk fungsi semacam ini. Anda dapat menambahkan pernyataan skrip yang dimaksudkan untuk melacak dan memantau perkembangan

permintaan data aplikasi dengan menggunakan fungsi ini, yang secara otomatis dijalankan setiap kali status permintaan berubah (Ford, 2009).

7.10 Respons Server Web: Penanganan

Anda sekarang harus menentukan jenis data yang diantisipasi aplikasi Anda terima setelah menambahkan logika pemrograman yang diperlukan untuk menangani data yang akan dikembalikan oleh server web. Salah satu properti objek XMLHttpRequest berikut menjalankan fungsi ini (Ford, 2009):

1. `ResponseText`. Data dikembalikan sebagai teks biasa.
2. `ResponseXML`. Data dikembalikan sebagai XML

Cara mengonfigurasi aplikasi Ajax Anda untuk menangani data teks yang dikembalikan oleh server web ditunjukkan dalam pernyataan berikut.

```
RequestObj.innerHTML = Request.responseText;
```

Seperti yang ditunjukkan di sini, Anda juga bisa mengonfigurasi aplikasi Anda untuk menangani data XML:

```
RequestObj.innerHTML = Request.responseXML;
```

Objek XMLHttpRequest telah dibuat, file telah ditentukan, koneksi server telah dibuat, koneksi telah diverifikasi, dan aplikasi sekarang siap untuk meminta data dari server web. Metode `send()`, yang memiliki sintaks berikut, harus digunakan untuk mengirim permintaan dan mengambil data.

```
send("string")
```

Seperti yang ditampilkan di sini, saat menggunakan mode `get`, nilai yang Anda tetapkan ke parameter string akan selalu `"null"`:

```
Request.send(null);
```

Setelah metode `send()` digunakan, prosedur pengunduhan dimulai, mengaktifkan fungsi anonim JavaScript yang Anda buat untuk mulai mengontrol prosedur pengunduhan. Mari kita lihat lagi aplikasi Ajax Joke of the Day untuk membantu menyatukan semua yang baru saja dikatakan. Berikut ini tautan ke seluruh file HTML untuk aplikasi ini:

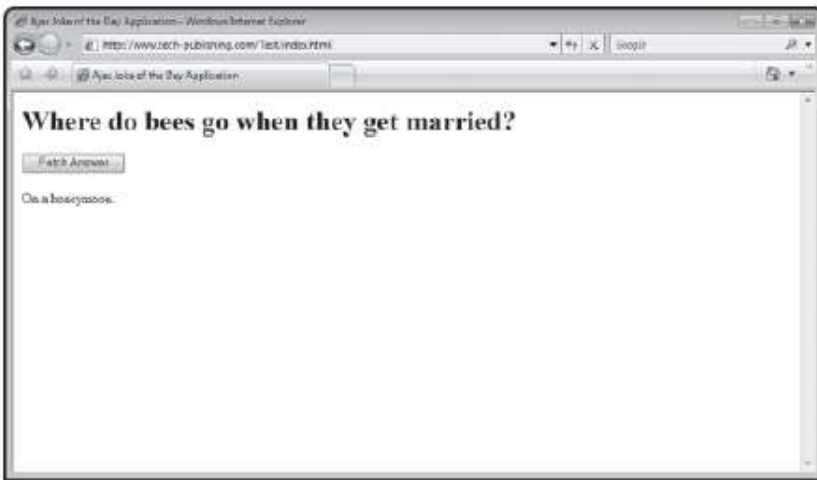
```
<HTML>
<HEAD>
<TITLE>Ajax Joke of the Day Application</TITLE>
<SCRIPT language = "javascript" type = "text/javascript">
var Request = false;
if (window.XMLHttpRequest) {
    Request = new XMLHttpRequest();
} else if (window.ActiveXObject) {
    Request = new ActiveXObject("Microsoft.XMLHTTP");
}
function retrieveJoke(url, elementID) {
    if(Request) {
        var RequestObj = document.getElementById(elementID);
        Request.open("GET", url);
        Request.onreadystatechange = function() {
            if (Request.readyState == 4 && Request.status == 200)
            {
                RequestObj.innerHTML = Request.responseText;
            }
        }
        Request.send(null);
    }
}
</SCRIPT>
</HEAD>
```

```
<BODY>
<H1>Where do bees go when they get married?</H1>
<FORM>
  <INPUT type = "button" value = "Fetch Answer"
    onclick = "retrieveJoke('joke.txt', 'DivTarget')">
</FORM>
<DIV id="DivTarget"> </DIV>
</BODY>
</HTML>
```

Seperti yang Anda lihat, objek XMLHttpRequest akan diwakili oleh variabel yang pertama kali didefinisikan dalam aplikasi ini. Untuk menentukan apakah browser web menjalankan Internet Explorer 7 atau yang lebih baru atau merupakan jenis browser yang sama sekali berbeda, blok kode pernyataan if disiapkan selanjutnya. Untuk memastikan bahwa mereka langsung berjalan saat halaman web dimuat, pernyataan kode yang membuat variabel dan memberi contoh objek XMLHttpRequest telah ditambahkan ke halaman HTML di luar fungsi apa pun.

Variabel Request pada awalnya disetel ke false sehingga nanti, JavaScript aplikasi dapat memeriksa nilainya untuk memastikan bahwa objek XMLHttpRequest telah dibuat. Kemudian, retrieveJoke() didefinisikan sebagai sebuah fungsi. Saat pengguna mengklik tombol pada formulir yang ditentukan di bagian tubuh halaman HTML, fungsi ini diaktifkan. Fungsi menerima dua argumen dari event handler onClick tombol. Argumen pertama menentukan nama file teks yang perlu diunduh dari server web, dan argumen kedua menentukan id dari sepasang tag DIV> /DIV> di bagian tubuh yang akan digunakan untuk menampilkan konten dari file teks setelah fungsi retrieveJoke() selesai memprosesnya (Ford, 2009).

Ketika fungsi `retrieveJoke()` dipanggil, pertama-tama ia memverifikasi bahwa turunan dari objek `XMLHttpRequest` telah dibuat sebelum mendeklarasikan variabel bernama `requestObj` dan memberinya nilai yang sesuai dengan tag `DIV>/DIV>` yang ditemukan di bagian tubuh dari halaman. Metode `open()` kemudian digunakan dengan opsi `GET`, dan fungsi anonim dibuat untuk menangani file teks yang diunduh server. Fungsi anonim ini menggunakan properti `innerHTML` objek `requestObj` untuk menampilkan konten file teks dalam tag `DIV>` dan `/DIV>` dari halaman HTML. Terakhir, metode `send()` digunakan untuk memulai pengunduhan dan memanggil fungsi anonim. Ilustrasi penggunaan aplikasi ini dapat dilihat pada Gambar 7.4.



Gambar 7.4. Menjalankan aplikasi AJAX Joke of the Day
(Sumber: Brinzarea-Iamandiet *al.*, 2009)

7.11 Mengontrol Permintaan XMLHttpRequest Saat Ini

Aplikasi Ajax yang telah Anda gunakan sejauh ini dalam buku ini semuanya menggunakan satu objek XMLHttpRequest. Kebutuhan untuk membuat aplikasi Ajax yang harus bereaksi terhadap berbagai tindakan pengguna sekaligus mengambil beberapa kumpulan data dari server web dapat muncul saat Anda mulai membuat aplikasi Ajax yang lebih rumit. Penggunaan objek XMLHttpRequest tunggal mencegah hal ini.

Objek XMLHttpRequest dapat didefinisikan dalam banyak contoh sebagai salah satu cara untuk menangani situasi ini. Sebagai ilustrasi, kalimat berikut mendefinisikan dua objek XMLHttpRequest.

```
var Request1 = false;
var Request2 = false;
if (window.XMLHttpRequest) {
    Request1 = new XMLHttpRequest();
    Request2 = new XMLHttpRequest();
} else if (window.ActiveXObject) {
    Request1 = new ActiveXObject("Microsoft.XMLHTTP");
    Request2 = new ActiveXObject("Microsoft.XMLHTTP");
}
```

Objek XMLHttpRequest harus dibuat sebagai instance terpisah menggunakan metode ini untuk setiap permintaan yang mungkin perlu dibuat oleh aplikasi Anda. Metode ini dapat dengan mudah menjadi sulit dikelola jika Anda membuat aplikasi yang perlu mendukung lebih dari dua atau tiga permintaan sekaligus. Untuk mengelola objek XMLHttpRequest di aplikasi Anda, sebaiknya gunakan array. Perlu diingat bahwa koleksi tipe objek apa pun yang didukung JavaScript dapat disimpan dalam array JavaScript.

DAFTAR PUSTAKA

- Brinzarea-Iamandi, Darie, B., Hendrix, C., & Audra. 2009. *AJAX and PHP: Building Modern Web Applications (2nd Edition)* (G. Singh, Ed.). Birmingham, B27 6PA, UK: PACKT publishing.
- Ford, J. L. 2009. *Ajax programming for the absolute beginner* (J. D. Technical, Ed.). Boston, MA: Course Technology PTR.
- T.Holdener, A. 2008. *AJAX The Definitive Guide* (First Edit; S. St.Laurent, Ed.). Gravenstein Highway North, Sebastopol: Oreilly.
- Tutorialspoint, T. (n.d.). AJAX Tutorial. Retrieved May 22, 2023, from [tutorialspoint.com website:](https://www.tutorialspoint.com/ajax/index.htm)
<https://www.tutorialspoint.com/ajax/index.htm>

BAB 8

DASAR-DASAR BASIS DATA

Oleh Andi Irfan

8.1 Pendahuluan

Basis data (database) adalah kumpulan data yang terorganisasi dengan baik, yang dirancang untuk menyimpan dan mengelola informasi dengan efisien. Basis data digunakan diberbagai bidang, termasuk bisnis, ilmu komputer, dan bidang lainnya (ZAENAL MUSTOFA, 2021).

Basis data memiliki peran penting dalam pengelolaan informasi karena menyediakan cara untuk mengatur dan menyimpan data dengan cara yang terstruktur. Dalam basis data, data disimpan dalam bentuk tabel yang terdiri dari baris dan kolom. Setiap baris dalam tabel mewakili entitas atau objek tertentu, sedangkan kolom mewakili atribut atau karakteristik entitas tersebut.

8.2 Definisi Basis Data

Basis data (database) adalah kumpulan terstruktur dari data yang saling terkait, yang disimpan dalam satu lokasi atau lebih. Basis data menyediakan cara untuk mengatur, menyimpan, mengelola dan mengakses data dengan efisien. Data dalam basis data dapat berupa angka, teks, gambar, suara, video atau bentuk lainnya yang dapat direpresentasikan secara digital (ZAENAL MUSTOFA, 2021).

Basis data terdiri dari entitas (objek) yang direpresentasikan oleh tabel. Setiap tabel memiliki kolom-kolom yang mewakili atribut-atribut dari entitas tersebut dan setiap baris dalam tabel mewakili satu instansi atau rekaman dari

entitas tersebut. Hubungan antara tabel-tabel tersebut ditentukan melalui kunci primer dan kunci asing.

Basis data memungkinkan pengguna untuk mengelola keamanan data dengan mengatur hak akses pengguna, melakukan enkripsi data dan menerapkan kebijakan keamanan data lainnya. Backup dan pemulihan data juga menjadi bagian penting dari basis data untuk melindungi dari kehilangan atau kerusakan. Ada berbagai jenis basis data yang tersedia, termasuk basis data relasional yang menggunakan model relasional dan SQL sebagai bahasa query, basis data berorientasi objek, basis data berbasis grafik dan banyak lagi yang lain. Pemilihan jenis basis data yang sesuai tergantung pada kebutuhan aplikasi dan sifat data yang akan disimpan (Azizah, 2022).

8.3 Sejarah Basis Data

Komponen basis data telah ada sejak zaman kuno, tapi pengembangan dan evolusinya menjadi lebih maju pada abad 20. Awal abad ke 20, pada tahun 1960 sistem basis data pertama kali dikembangkan menggunakan model hierarki. Model ini mengorganisasi data dalam hierarki dengan satu entitas utama yang berhubungan dengan beberapa entitas anak. IBM *System R* adalah salah satu sistem basis data pertama yang menggunakan model relasional.

Pada tahun 1970, Edgar F. Codd seorang ilmuwan komputer, memperkenalkan model relasional sebagai cara baru untuk mengorganisasi data. Model ini didasarkan pada konsep tabel yang terdiri dari baris dan kolom. Hubungan antara tabel-tabel tersebut ditentukan oleh kunci primer dan kunci asing. Model relasional memungkinkan akses data yang lebih fleksibel dan pemisahan antara struktur data dan operasi pengolahan data.

Pada tahun 1974, Donald D. Chamberlin dan Raymond F. Boyce memperkenalkan bahasa query yang dikenal sebagai SQL (*Structured Query Language*). SQL menjadi standar yang digunakan untuk mengakses, mengelola dan mengubah data dalam sistem basis data relasional.

Pada tahun 1980, pengembangan sistem basis data komersial yang canggih semakin berkembang. Perusahaan seperti Oracle, IBM dan Microsoft memperkenalkan produk-produk basis data mereka yang mendukung model relasional dan menyediakan fitur-fitur seperti transaksi, keamanan dan skalabilitas.

Pada tahun 1990, basis data objek relasional mulai muncul. Model ini menggabungkan fitur-fitur dari model relasional dan model objek, memungkinkan penyimpanan dan pengelolaan data yang lebih kompleks, termasuk tipe data kompleks seperti gambar, suara dan video.

Seiring dengan perkembangan internet dan aplikasi web, basis data berbasis web menjadi penting. Basis data ini digunakan untuk menyimpan dan mengelola data yang digunakan oleh situs web dan aplikasi web.

Dalam beberapa dekade terakhir, basis data terdistribusi dan basis data cloud telah menjadi populer. Basis data terdistribusi memungkinkan penyimpanan dan pengolahan data diberbagai lokasi fisik, sedangkan basis data cloud memanfaatkan infrastruktur cloud untuk menyediakan akses mudah dan skalabilitas yang tinggi (Ramadhanty, 2020).

8.4 Fungsi Basis Data

Basis data memiliki fungsi penting dalam pengelolaan dan penggunaan data (Azizah, 2022). Berikut adalah beberapa fungsi utama basis data :

1. Penyimpanan data

Fungsi utama dari basis data adalah menyimpan data dengan cara yang terstruktur dan terorganisasi. Basis data memungkinkan pengguna untuk membuat tabel, kolom dan baris yang mencerminkan struktur data yang diinginkan. Dengan menggunakan basis data, data dapat disimpan dengan efisien dan mudah diakses.

2. Pemeliharaan data yang konsisten

Basis data memungkinkan pemeliharaan data yang konsisten. Dengan menggunakan aturan integritas dan kunci referensial, basis data memastikan bahwa yang disimpan tetap akurat dan konsisten. Misalnya, aturan integritas memastikan bahwa data yang dimasukkan kedalam basis data memenuhi batasan yang telah ditetapkan, sementara kunci referensial menjaga hubungan yang konsisten antara entitas dalam tabel.

3. Pencarian dan pemrosesan data

Basis data menyediakan fungsi untuk mencari, menyortir, memfilter dan memproses data. Pengguna dapat menggunakan bahasa query seperti SQL untuk melakukan operasi tersebut. Basis data juga menggunakan indeks untuk meningkatkan kinerja pencarian data dengan membuat struktur tambahan yang mempercepat operasi pencarian.

4. Keamanan data

Fungsi keamanan basis data sangat penting untuk melindungi data dari akses yang tidak sah atau perubahan yang tidak diotorisasi. Basis data menyediakan fitur keamanan seperti hak akses pengguna yang terkontrol, enkripsi data dan mekanisme audit untuk membantu aktifitas pengguna. Hal ini membantu kerahasiaan, integritas dan ketersediaan data yang sensitif.

5. Berbagi data

Basis data memungkinkan berbagi data diantara pengguna dan aplikasi yang berbeda. Dengan menggunakan sistem basis data yang terpusat, pengguna yang berbeda dapat mengakses dan menggunakan data yang sama secara bersamaan. Ini memfasilitasi kolaborasi, koordinasi dan konsistensi data diseluruh organisasi.

6. Pemulihan data

Basis data memiliki mekanisme pemulihan data yang memungkinkan pemulihan data setelah terjadinya kegagalan sistem atau kerusakan fisik. *Backup* rutin dan pemulihan data merupakan langkah penting untuk menjaga integritas dan ketersediaan data dalam situasi darurat.

7. Pengolahan dan analisis data

Basis data menyediakan dasar untuk pengolahan dan analisis data yang lebih canggih. Dengan menggunakan basis data, pengguna dapat melakukan analisis, laporan dan pemodelan yang kompleks. Basis data juga memungkinkan pengguna untuk menggabungkan dan memanipulasi data dari berbagai sumber untuk mendapatkan pemahaman yang lebih lengkap tentang data.

Dengan memenuhi fungsi-fungsi ini, basis data membantu dalam pengelolaan data yang efisien, konsisten dan terstruktur, serta mendukung pengambilan keputusan yang lebih baik.

8.5 Komponen Basis Data

Basis data terdiri dari beberapa komponen utama yang bekerja bersama-sama untuk mengelola dan menyediakan akses terhadap data (Spada.kemdikbud.go.id, 2019). Beberapa komponen tersebut adalah :

1. Perangkat keras

Komponen perangkat keras basis data mencakup semua perangkat fisik yang digunakan dalam sistem basis data, seperti server, komputer, jaringan dan perangkat penyimpanan seperti harddisk atau solid-state drive(SSD). Perangkat keras digunakan untuk menyimpan, mengelola dan mengakses data dalam basis data.

2. Perangkat lunak

Komponen perangkat lunak mencakup semua program dan sistem operasi yang digunakan dalam sistem basis data. Termasuk sistem manajemen basis data (DBMS) yang menyediakan fungsi-fungsi penting untuk mengelola basis data seperti penyimpanan data, manajemen transaksi dan akses data. Contoh DBMS adalah Oracle, MySQL, SQL Server dan PostgreSQL.

3. Data

Komponen data mencakup informasi yang disimpan dalam basis data, seperti informasi pelanggan, transaksi penjualan atau catatan inventaris dan lain sebagainya. Data dapat berupa teks, angka, gambar, suara, video atau tipe data yang lain. Data ini diorganisir dan disimpan dalam tabel dan entitas yang sesuai dalam basis data.

4. Struktur basis data

Struktur basis data merujuk pada cara data diatur dan dihubungkan dalam basis data. Hal ini mencakup tabel, kolom, baris dan hubungan antara entitas. Struktur basis data memastikan bahwa data disimpan dengan cara yang terstruktur dan terorganisasi, memungkinkan pengelolaan data yang efisien dan akses yang cepat.

5. Bahasa Query

Bahasa query digunakan untuk berinteraksi dengan basis data dan melakukan operasi seperti mengambil, menyisipkan, memperbarui atau menghapus data. Contoh Bahasa query yang umum digunakan adalah *Structured*

Query Language (SQL), yang digunakan untuk mengelola dan memanipulasi data dalam basis data.

6. Pengguna basis data

Pengguna basis data adalah individu atau entitas yang menggunakan atau berinteraksi dengan basis data. Pengguna meliputi administrator basis data yang bertanggung jawab untuk mengelola basis data, pengembang aplikasi yang menggunakan data dari basis data atau pengguna akhir yang mengakses dan menggunakan data dalam basis data.

7. Keamanan

Komponen keamanan melibatkan langkah-langkah dan kebijakan yang diterapkan untuk melindungi integritas, kerahasiaan dan ketersediaan data dalam basis data, termasuk pengaturan hak akses pengguna, enkripsi data, mekanisme otentikasi dan pengendalian akses untuk melindungi data dari akses yang tidak sah atau perubahan yang tidak diotorisasi.

8. Indeks

Indeks adalah struktur data tambahan yang digunakan untuk meningkatkan kinerja pencarian data dalam basis data. Indeks menciptakan daftar nilai unik dari satu atau lebih kolom dalam tabel, mempercepat operasi pencarian data dengan menunjukkan lokasi data yang sesuai.

9. Backup dan pemulihan

Komponen ini melibatkan strategi dan proses untuk mencadangkan data secara teratur dan pemulihan data jika terjadi kegagalan sistem atau kehilangan data. Backup dan pemulihan yang baik penting untuk menjaga integritas dan ketersediaan data dalam situasi darurat.

Komponen-komponen ini bekerja bersama-sama untuk menyediakan pengelolaan data yang efisien, akses yang cepat,

kemanan yang baik dan pemulihan yang andal dalam sebuah basis data.

8.6 Jenis Basis Data

Basis data terdiri dari beberapa jenis yang umum digunakan, tergantung pada struktur data dan cara data diorganisir(Yudhistira, 2019). Jenis basis data yang umum digunakan adalah sebagai berikut :

1. Basis data relasional

Basis data relasional adalah jenis basis data yang paling umum digunakan. Data disimpan dalam bentuk tabel yang terdiri dari baris dan kolom. Hubungan antar tabel ditentukan melalui kunci primer dan kunci asing. Contoh basis data relasional adalah MySQL, Oracle, SQL Server dan PostgreSQL

2. Basis data heirarki

Basis data ini mengorganisir data dalam bentuk struktur pohon atau hirarki. Setiap entitas memiliki hubungan induk-anak yang terdiri dari satu entitas induk dan beberapa entitas anak. Basis data jenis ini digunakan untuk aplikasi yang memiliki struktur data yang jelas dan baku. Contoh dari basis data hirarki adalah *Information Management System* (IMS) dan *Integrated Database System*.

3. Basis data jaringan

Basis data jaringan memiliki struktur data yang kompleks dengan hubungan yang kompleks pula. Data disimpan dalam bentuk struktur jaringan dengan entitas yang dapat terhubung dengan entitas lain. Basis data jaringan lebih fleksibel dari pada basis data hirarki, tetapi kompleksitasnya dapat menyulitkan pemodelan dan manipulasi data. Contoh DMS jaringan adalah *Integrated Database System* (IDS) dan *Integrated Data Store*.

4. Basis data objek

Basis data objek memungkinkan penyimpanan dan manipulasi objek kompleks seperti gambar, suara, video dan objek-objek lainnya. Basis data ini memperluas model relasional dengan menambahkan kemampuan untuk menyimpan dan mengelola tipe data kompleks. Contoh basis data objek adalah Oracle's Object Relational Database Management System (ORDDMS) dan PostgreSQL.

5. Basis data berorientasi dokumen

Basis data berorientasi dokumen dirancang untuk mengelola data semi-struktur dan tidak terstruktur seperti dokumen XML atau JSON. Basis data ini menyimpan data dalam format dokumen dan menyediakan kemampuan untuk pencarian dan manipulasi data berdasarkan struktur dokumen. Contoh basis data berorientasi dokumen adalah *MongoDB* dan *Couchbase*.

6. Basis data kolom

Basis data kolom membalik struktur penyimpanan tradisional dengan menyimpan data secara kolom dari pada baris. Basis data ini digunakan pada situasi dimana hanya sebagian data yang diperlukan dalam operasi tertentu, karena memungkinkan akses yang cepat kekolom tertentu. Contoh basis data kolom adalah *Cassandra* dan *Vertica*.

7. Basis data grafik

Basis data grafik dirancang khusus untuk mengelola data yang memiliki struktur grafik dengan simpul dan hubungan antara simpul-simpul tersebut. Basis data grafik memungkinkan analisis dan manipulasi data berdasarkan hubungan grafik yang kompleks. Contoh basis data grafik adalah Neo4j dan ArangoDB.

Setiap jenis basis data memiliki karakteristik, kelebihan dan kelemahan. Pemilihan jenis basis data yang tepat

tergantung pada kebutuhan aplikasi, struktur data dan persyaratan pengelolaan data yang diinginkan.

8.7 Konsep Dasar Basis Data

Dalam basis data ada beberapa konsep dasar yang perlu untuk diketahui (Tulus, 2015). Berikut ini adalah konsep dasar dalam basis data :

1. Entitas

Entitas adalah objek yang dapat diidentifikasi dalam dunia nyata atau dalam domain aplikasi yang relevan. Contohnya bisa berupa orang, tempat, barang atau konsep lainnya. Entitas memiliki atribut yang menggambarkan karakteristik atau property mereka.

2. Atribut

Atribut adalah informasi spesifik yang terkait dengan entitas. Misalnya jika entitas adalah pelanggan, atribut dapat mencakup nama pelanggan, alamat, nomor telepon dan lain sebagainya. Setiap entitas memiliki satu atau lebih atribut yang mendefinisikan propertinya.

3. Relasi

Relasi adalah hubungan antara entitas dalam basis data. Relasi ini menggambarkan bagaimana entitas saling terhubung atau berhubungan satu sama lain. Misalnya dalam basis data toko, entitas pelanggan dapat memiliki relasi dengan entitas pesanan melalui kunci asing yang menghubungkan mereka.

4. Kunci primer

Kunci primer adalah atribut atau kumpulan atribut yang unik mengidentifikasi setiap entitas dalam satu tabel. Kunci primer memastikan bahwa tidak ada duplikat data dalam tabel. Kunci primer biasanya digunakan sebagai acuan dalam hubungan dengan tabel lain.

5. Kunci asing

Kunci asing adalah atribut atau kumpulan atribut dalam sebuah tabel yang menghubungkan entitas dengan entitas lain dalam basis data. Kunci asing digunakan untuk membangun hubungan antara tabel-tabel dalam basis data. Kunci asing mengacu pada kunci primer di tabel lain.

6. Skema basis data

Skema basis data adalah struktur dan representasi logis dari basis data. Skema basis data mendefinisikan entitas, atribut, relasi, kunci primer dan kunci asing dalam basis data. Skema basis data memberikan panduan tentang bagaimana data diorganisir dan disimpan dalam basis data.

7. Normalisasi

Normalisasi adalah proses desain basis data untuk mengurangi redundansi dan anomali dalam data. Tujuannya adalah memastikan bahwa basis data memiliki struktur yang efisien, menghindari duplikasi data yang tidak perlu dan menjaga integritas data. Normalisasi dilakukan dengan memisahkan data ke dalam tabel-tabel yang lebih kecil dan mengatur hubungan antara mereka.

8. Bahasa Query

Bahasa Query digunakan untuk mengakses dan memanipulasi data dalam basis data. Contoh bahasa query yang umum digunakan adalah SQL (*Structured Query Language*). Bahasa query memungkinkan pengguna untuk melakukan operasi seperti mengambil data (SELECT), menambahkan data (INSERT), memperbarui data (UPDATE) atau menghapus data (DELETE) dalam basis data.

9. Transaksi

Transaksi adalah unit operasi yang konsisten dan atomik dalam basis data. Transaksi terdiri dari satu atau lebih operasi yang dijalankan bersama-sama. Tujuan utama transaksi adalah menjaga konsistensi dan integritas data

dalam basis data. Jika suatu transaksi gagal, semua perubahan yang dilakukan oleh transaksi tersebut akan dibatalkan, sehingga basis data tetap konsisten.

8.8 Operasi Dasar Basis Data

Operasi dasar dalam basis data melibatkan manipulasi dan pengelolaan data (Fenia, 2018). Beberapa operasi dasar yang umum digunakan dalam basis data :

1. Select

SELECT digunakan untuk mengambil data dari basis data dengan memilih kolom tertentu yang ingin ditampilkan dan menerapkan kondisi untuk memfilter data yang diambil.

Contoh penggunaan SELECT

```
SELECT column1, column2 FROM table_name WHERE condition;
```

2. Insert

INSERT digunakan untuk menyisipkan data baru ke dalam tabel basis data. Dalam operasi ini, tentukan tabel tujuan dan menyediakan nilai-nilai yang ingin dimasukkan ke dalam kolom-kolom yang sesuai.

Contoh penggunaan INSERT

```
INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

3. Update

UPDATE digunakan untuk memperbarui data yang ada dalam basis data. Untuk melakukan operasi UPDATE tentukan tabel tujuan, kolom yang ingin diperbarui dan nilai baru yang ingin diberikan ke kolom tersebut. Dalam operasi ini juga dapat digunakan kondisi untuk membatasi data yang akan diperbarui.

Contoh penggunaan UPDATE

```
UPDATE table_name SET column1 = value1, column2 =  
value2 WHERE condition;
```

4. Delete

DELETE digunakan untuk menghapus data dari tabel dalam basis data. Dalam operasi ini tentukan tabel tujuan dan kondisi untuk membatasi data yang akan dihapus.

Contoh penggunaan DELETE

```
DELETE FROM table_name WHERE condition;
```

5. Join

Operasi JOIN digunakan untuk menggabungkan data dari dua atau lebih tabel berdasarkan kunci relasional antara tabel tersebut. Operasi JOIN memungkinkan untuk mengambil data yang terkait dari beberapa tabel dalam satu hasil query.

Contoh penggunaan JOIN

```
SELECT column1, column2 FROM table1 JOIN table2 ON  
table1.key = table2.key;
```

6. Group By

GROUP BY digunakan untuk mengelompokkan baris data berdasarkan nilai pada kolom tertentu. Operasi ini biasanya digunakan bersamaan dengan fungsi agregat seperti COUNT, SUM, AVG dan lain sebagainya.

Contoh penggunaan GROUP BY

```
SELECT column1, COUNT(column2) FROM table_name  
GROUP BY column1;
```

7. Order By

Operasi ORDERED BY digunakan untuk mengurutkan hasil query berdasarkan nilai pada satu atau lebih kolom.

Pengurutan dapat dilakukan secara Ascending (ASC) atau Descending (DESC).

Contoh penggunaan ORDERED BY

```
SELECT column1, column2 FROM table_name ORDER BY  
column1 ASC;
```

8. Create

CREATE digunakan untuk membuat tabel baru dalam basis data. Tentukan nama tabel dan kolom yang ingin ditambahkan beserta tipe data dan konstrain yang sesuai.

Contoh penggunaan CREATE

```
CREATE TABLE table_name (  
    column1 datatype constraint,  
    column2 datatype constraint,  
    ...  
);
```

9. Drop

Operasi DROP digunakan untuk menghapus tabel atau objek lain dalam basis data. Operasi DROP menghapus data secara permanen, pastikan menggunakan operasi ini dengan hati-hati.

Contoh penggunaan DROP

```
DROP TABLE table_name;
```

8.9 Desain Basis Data

Desain basis data merupakan proses merancang struktur, skema dan hubungan antara tabel dalam basis data (Purnami, 2015). Tujuan utamanya adalah menciptakan struktur yang efisien, normalisasi yang baik dan kemampuan yang memadai untuk menyimpan, mengakses dan memanipulasi data.

Berikut adalah langkah-langkah umum yang dilakukan dalam desain basis data :

1. Analisis kebutuhan

Langkah pertama dalam desain basis data adalah memahami kebutuhan bisnis atau aplikasi yang akan menggunakan basis data. Identifikasi entitas utama, atribut dan hubungan yang perlu disimpan dalam basis data.

2. Indentifikasi entitas dan atribut

Identifikasi entitas dan atribut yang akan disimpan dalam basis data. Entitas dapat berupa objek atau konsep dalam dunia nyata yang relevan dengan aplikasi.

3. Normalisasi

Langkah berikutnya adalah normalisasi, proses merancang skema basis data yang menghilangkan redudansi dan anomali data. Normalisasi melibatkan membagi data kedalam tabel-tabel yang lebih kecil dan memastikan dependensi fungsional yang tepat antara atribut-atribut.

4. Hubungan antara tabel (relasi)

Berikutnya melakukan identifikasi hubungan antar tabel dengan menggunakan kunci primer dan kunci aasing. Hubungan ini menggambarkan bagaimana entitas terkait dalam basis data. Pastikan bahwa hubungan antar tabel mendukung integritas referensial dan mempertahankan konsistensi data.

5. Menentukan tipe data

Tentukan tipe data yang sesuai untuk setiap atribut dalam tabel. Pilih tipe data yang tepat berdasarkan jenis data yang akan disimpan sesuai dengan persyaratan aplikasi.

6. Membuat skema basis data

Setelah semua entitas, atribut dan hubungan ditentukan, buat skema basis data yang menggambarkan struktur dan hubungan antar tabel dalam basis data. Skema ini dapat diwakili dalam bentuk diagram ER (Entity Relationship) atau diagram lainnya.

7. Menentukan batasan dan aturan
Tentukan Batasan dan aturan dalam basis data, seperti batasan integritas referensial, batasan unik, aturan validasi dan lain sebagainya. Hal ini membantu memastikan keberlanjutan data yang konsisten dan berkualitas tinggi.
8. Membuat indeks
Menentukan indeks pada kolom-kolom yang sering digunakan untuk pencarian atau pengurutan data. Indeks meningkatkan kinerja basis data dengan mempercepat akses data.
9. Uji dan optimasi
Uji dan evaluasi skema basis data untuk memastikan bahwa desain tersebut memenuhi kebutuhan bisnis dan aplikasi. Identifikasi dan perbaiki masalah kinerja yang mungkin timbul.
10. Implementasi
Implementasikan desain basis data dengan menggunakan DBMS (*Database Management System*) yang sesuai. Buat tabel, indeks dan objek lain yang diperlukan dalam DBMS.

Desain basis data adalah proses iterative yang melibatkan analisis kebutuhan, pemodelan dan optimasi.

DAFTAR PUSTAKA

- Azizah, L. 2022. *Database: Jenis, Fungsi dan Manfaatnya*.
<https://www.gramedia.com/best-seller/apa-itu-database/>
- Fenia, R. W. 2018. *Basis Data (Database): Pengertian, Tujuan, Operasi, dan Bahasa Basis Data*. <https://www.kmtech.id/post/basis-data-database-pengertian-tujuan-operasi-dan-bahasa-basis-data>
- Purnami, C. T. 2015. *Desain Data Base*.
- Ramadhanty, N. 2020. *SEJARAH BASIS DATA*.
https://www.academia.edu/30776583/SEJARAH_BASIS_DATA_Tahun_1960
- Spada.kemdikbud.go.id. 2019. *Komponen Sistem Basis Data*.
<https://lmsspada.kemdikbud.go.id/mod/page/view.php?id=80557>
- Tulus, W. 2015. Konsep Basis Data. *Yogyakarta : CV. Yukumedia*, 1–28.
- Yudhistira. 2019. *Mengenal Database serta Fungsi dan Jenisnya*.
<https://www.bhinneka.com/blog/database-adalah/>
- ZAENAL MUSTOFA. 2021. *Basis Data dan DBMS*. <https://teknik-informatika-s1.stekom.ac.id/informasi/baca/Basis-Data-dan-DBMS/6d492170753211fcde587882d77e1e8dcce1bc27>

BAB 9

Pengenalan PHP

Oleh Yudhistira Adhitya Pratama

9.1 Pendahuluan

PHP adalah singkatan dari "PHP: *Hypertext Preprocessor*". PHP adalah bahasa pemrograman sisi server yang digunakan untuk pengembangan web. PHP dapat menghasilkan konten HTML secara dinamis, yang memungkinkan interaksi antara pengguna dan situs web. PHP sering digunakan untuk membangun aplikasi web yang kompleks dan dinamis, seperti forum online, sistem manajemen konten, toko elektronik, dan banyak lagi. PHP juga memiliki dukungan yang luas dan banyak framework populer yang memudahkan pengembangan web. (*The PHP Manual*, 2023)

PHP dapat digunakan dalam dua cara yaitu :

1. *Server-side scripting*

Pada awalnya, PHP diciptakan dengan tujuan untuk membuat konten web yang dinamis, dan hingga saat ini, PHP tetap sangat sesuai untuk tugas tersebut. Untuk menghasilkan HTML, diperlukan parser PHP dan server web untuk mengirimkan file dokumen yang telah dikodekan. PHP juga populer digunakan untuk menghasilkan konten dinamis melalui koneksi database, dokumen XML, grafik, file PDF, dan banyak lagi. (Nixon, 2014)

2. *Command-line scripting*

PHP dapat menjalankan *script command-line*, seperti PERL atau unix-shell. Mungkin saja untuk melakukan beberapa

tugas administrasi sistem seperti backup dan log parsing, yang memanfaatkan CRON job dalam menjalankannya secara rutin. (Nixon, 2014)

3. Client-side GUI applications

Menggunakan PHP-GTK, dapat membuat aplikasi GUI lintas platform yang lengkap menggunakan PHP. (Nixon, 2014)

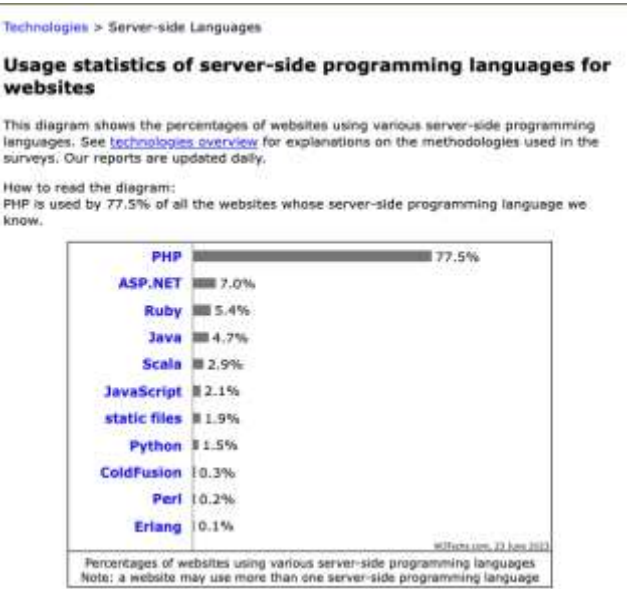
PHP dapat dijalankan pada semua sistem operasi utama, mulai dari varian Unix (termasuk Linux, FreeBSD, Ubuntu, Debian, dan Solaris) hingga Windows dan macOS. PHP dapat digunakan dengan semua server web terkemuka, termasuk server Apache, Nginx, dan OpenBSD, untuk beberapa contohnya bahkan lingkungan cloud seperti Azure dan Amazon semakin populer.

Salah satu fitur paling penting dari PHP adalah dukungannya yang luas terhadap berbagai database. PHP mendukung semua database utama (termasuk MySQL, PostgreSQL, Oracle, Sybase, MSSQL, DB2, dan database yang sesuai dengan ODBC), bahkan banyak yang kurang dikenal. Bahkan database dengan basis NoSQL terbaru seperti CouchDB dan MongoDB juga didukung. Dengan PHP, membuat halaman web dengan konten dinamis dari database sangatlah mudah. (Tatroe, MacIntyre and Lerdorf, 2014)

Selain itu, PHP menyediakan perpustakaan kode PHP untuk melakukan tugas-tugas umum, seperti abstraksi database, penanganan kesalahan, validasi form, pengiriman email dan sebagainya, yang dengan mudah dicari melalui situs <http://packagist.org>.

Pada gambar 9.1 menunjukkan penggunaan PHP yang dikompilasi oleh W3Techs pada Juni 2023. Data yang menarik disini adalah bahwa 77,5% dari semua website yang disurvei menggunakan PHP. Jika kita melihat metodologi yang digunakan dalam survei W3Techs, dimana mereka memilih 10

juta website teratas (berdasarkan lalu lintas; popularitas website) di dunia. Seperti yang terlihat, PHP benar-benar memiliki tingkat penggunaan yang sangat luas.



Gambar 9.1. Kompilasi penggunaan Server-side Scripting
(Sumber :
https://w3techs.com/technologies/overview/programming_language) Diakses pada 23 Juni 2023

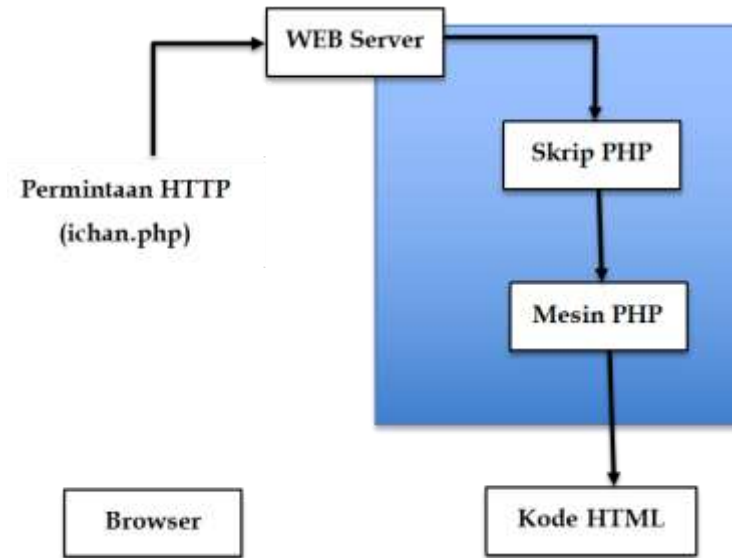
Teknologi web juga sangat banyak dimanfaatkan dalam pengembangan berbagai sistem informasi, khususnya yang mendukung bisnis, seperti :

1. Perancangan Sistem Informasi Pemesanan Tiket Bus pada PT. Putra Pelangi Perkasa. (Pratama *et al.*, 2022)
2. Pengembangan Sistem Informasi Point of Sales untuk UMKM, Studi Kasus: Arjuna Farm. (J J Pangaribuan *et al.*, 2022)

3. Pengembangan Sistem Informasi Penyewaan Kendaraan pada CV. Bitu Jaya Mandiri. (Jefri Junifer Pangaribuan, Jusin, *et al.*, 2022)
4. Pengembangan Website Pencarian Dan Pemesanan Jasa Guru Les Privat Berdasarkan Model C2C Marketplace. (Pratama, Lawi and Jusin, 2022)
5. Analisis Perancangan dan Implementasi Sistem Informasi Stationary Berbasis Web pada PT. Indako Trading Coy. (Chairis and Maulana, 2022)
6. Simpuru: Gamifikasi Pembelajaran Bahasa Jepang dalam Aplikasi Berbasis Web. (Jefri Junifer Pangaribuan, Maulana, *et al.*, 2022)

Selain beberapa sistem di atas, Teknologi web juga secara luas diterapkan dalam ecommerce yang dapat menunjang multi platform. (Muttaqin *et al.*, 2023).

Konsep kerja PHP dimulai dengan permintaan halaman web oleh browser. Browser akan mendapatkan URL dari web server. Ketika web server menerima berkas PHP yang diminta, isinya akan dikirim ke mesin PHP untuk diproses. Mesin PHP akan menghasilkan kode HTML yang akan dikembalikan ke web server. Selanjutnya, web server akan mengirimkan hasilnya kepada pengguna. Skema kerja PHP dapat dilihat di bawah ini :



Gambar 9.2. Konsep Kerja PHP (Asari, Mayatopani and others, 2023)

9.2 Instalasi

Seperti yang disebutkan sebelumnya, PHP tersedia untuk banyak sistem operasi dan platform. Oleh karena itu, untuk petunjuk instalasi PHP dapat ditemukan pada halaman berikut <https://www.php.net/manual/en/install.php>. Pada halaman tersebut dapat ditemukan berbagai langkah instalasi sesuai dengan sistem operasi yang diinginkan.

Jika ingin mengubah konfigurasi, pengaturan konfigurasi PHP disimpan dalam file `php.ini` yang dapat ditemukan pada folder instalasi PHP. (Tatroe, MacIntyre and Lerdorf, 2014)

9.3 Dasar PHP

Pada bagian ini akan dibahas dasar dari PHP, termasuk struktur dasar, tipe data, variabel, operator, pernyataan kondisi, dan pernyataan percabangan.

9.3.1 Struktur Dasar

Struktur dasar sebuah script PHP terdiri dari beberapa elemen utama. Berikut adalah contoh struktur dasar sebuah script PHP:

```
<?php
    // Kode PHP ditulis di sini
?>
```

1. Tag PHP: Setiap script PHP dimulai dengan tag **<?php** dan diakhiri dengan **?>**. Semua kode PHP akan berada di antara tag ini.
2. Kode PHP: Ini adalah bagian di mana kode PHP ditulis dan yang akan dieksekusi. Kode PHP dapat berisi pernyataan, deklarasi variabel, pemrosesan data, dan sebagainya.
3. Komentar: Juga dapat menambahkan komentar di dalam kode PHP untuk memberikan penjelasan atau dokumentasi. Komentar tidak akan dieksekusi oleh PHP dan hanya berfungsi sebagai catatan bagi pembaca kode. (Nixon, 2014)

Ada beberapa aturan dalam penulisan script PHP yang harus dipenuhi, yaitu :

1. *Case Sensitivity*

Pemberian nama class dan fungsi pada PHP, begitu juga constructs dan kata kunci bawaan seperti echo, while, class, dll. bersifat tidak *case sensitive*. Seperti pada contoh :

```
echo("hello, world");
ECHO("hello, world");
EcHo("hello, world");
```

Ketiga kode diatas akan menghasilkan tampilan yang sama,

Sebaliknya penamaan variabel bersifat case sensitive, seperti \$name, \$NAME, \$NaMe adalah tiga variabel yang berbeda. (Nixon, 2014)

2. *Statements dan Semicolons*

Statement adalah kumpulan kode PHP yang melakukan sesuatu. Pernyataan bisa se-simple pemberian nilai pada variabel atau sekompleks perulangan. Berikut adalah contoh kecil pernyataan PHP, termasuk pemanggilan fungsi, penugasan, dan pernyataan if:

```
echo "Hello, world";  
myFunction(42, "O'Reilly");  
$a = 1;  
$name = "Elphaba";  
$b = $a / 25.0;  
if ($a == $b) {  
    echo "Rhyme? And Reason?";  
}
```

PHP menggunakan titik koma (;) untuk memisahkan pernyataan sederhana. Pernyataan kompleks yang menggunakan kurung kurawal untuk menandai blok kode, seperti pengujian kondisional atau perulangan, tidak memerlukan titik koma setelah kurung kurawal penutup. Berbeda dengan bahasa pemrograman lainnya, dalam PHP titik koma sebelum kurung kurawal penutup tidak opsional. (Nixon, 2014)

3. Komentar

Komentar memberikan informasi kepada orang yang membaca kode Anda, tetapi akan diabaikan oleh PHP saat eksekusi. Komentar ini sangat dapat membantu dalam pemeriksaan kode jika terjadi error di masa yang akan datang. (Tatroe, MacIntyre and Lerdorf, 2014)

Komentar dapat dibuat dengan tanda // untuk komentar baris tunggal, /* */ untuk komentar beberapa baris sekaligus. Berikut contoh penggunaan komentar :

```
// ini komentar baris tunggal
/* ini komentar
dengan beberapa baris */
```

9.3.2 Tipe Data

PHP adalah bahasa berjenis dinamis, yang berarti tipe data dari suatu variabel ditentukan saat runtime berdasarkan nilai yang diberikan kepadanya. PHP mendukung berbagai jenis data, termasuk bilangan bulat, bilangan pecahan, string, boolean, array, dan objek.

Berikut adalah daftar jenis data yang didukung oleh PHP:

1. Integer: Bilangan bulat tanpa titik desimal. Contohnya, 42.
2. Float: Bilangan dengan titik desimal. Contohnya, 3,14.
3. String: Urutan karakter. Contohnya, "Halo, dunia!".
4. Boolean: Nilai yang dapat berupa benar (true) atau salah (false).
5. Array: Kumpulan nilai yang terurut.
6. Objek: Sebuah contoh dari kelas.
7. Null: Nilai khusus yang mewakili ketiadaan nilai.

PHP juga mendukung tipe casting, yang memungkinkan Anda mengonversi nilai dari satu tipe data ke tipe data lainnya. Misalnya, Anda dapat mengonversi sebuah string menjadi bilangan bulat menggunakan fungsi (int) atau intval(), atau mengonversi bilangan bulat menjadi string menggunakan fungsi (string) atau strval().

Selain jenis data dasar ini, PHP juga mendukung struktur data yang lainnya seperti array asosiatif, array multidimensi, dan objek. Struktur data ini memungkinkan Anda menyimpan dan memanipulasi data kompleks dengan cara yang terstruktur.

9.3.3 Variabel

Dalam PHP variabel ditulis dengan tanda (\$). Contohnya :

```
$nama  
$umur  
$jenis_kelamin
```

Sebuah variabel dapat menyimpan nilai dari tipe apa pun. Tidak ada pemeriksaan tipe pada variabel saat kompilasi atau saat runtime. Anda dapat menggantikan nilai variabel dengan nilai lain yang memiliki tipe yang berbeda.

```
$variabel = "Fred";  
$variabel = 35;  
$variabel = array("Anto", 35, "Bagas");
```

Tidak ada sintaks eksplisit untuk mendeklarasikan variabel dalam PHP. Ketika nilai variabel pertama kali diatur, variabel tersebut dibuat. Dengan kata lain, mengatur nilai ke variabel juga berfungsi sebagai deklarasi. Sebagai contoh, ini adalah program PHP yang valid dan lengkap :

```
$hari = 60 * 60 * 24;  
echo "Ada {$hari} detik dalam sehari";  
Ada 86400 detik dalam sehari.
```

Ketika membuat variabel dalam PHP, setidaknya harus mengikuti empat aturan berikut:

1. Nama variabel harus diawali dengan huruf abjad atau karakter _ (garis bawah).
2. Nama variabel hanya dapat mengandung karakter a-z, A-Z, 0-9, dan _ (garis bawah).
3. Nama variabel tidak boleh mengandung spasi. Jika sebuah variabel terdiri dari lebih dari satu kata, gunakan karakter _ (garis bawah) sebagai pemisah (misalnya, \$user_name).
4. Nama variabel bersifat case-sensitive. Variabel \$High_Score berbeda dengan variabel \$high_score. (Nixon, 2014)

9.3.4 Operator

Operator adalah perintah matematika, string, perbandingan, dan logika seperti tambah, kurang, kali, dan bagi. PHP terlihat sangat mirip dengan operasi aritmatika biasa; sebagai contoh, pernyataan berikut akan menghasilkan output 8:

```
echo 2 + 6;
```

Berikut beberapa operator yang dapat dioperasikan pada PHP :

1. Operator Aritmatika

Operator ini digunakan untuk melakukan operasi matematika, ada empat operasi utama yaitu tambah, kurang, kali, dan bagi. Serta untuk mencari sisa hasil bagi (modulus), dan untuk mengincrement atau mendecrement sebuah nilai. Pada gambar dapat dilihat jenis-jenis dan contoh dari operator aritmatika.

Operator	Description	Example
+	Addition	<code>\$j + 1</code>
-	Subtraction	<code>\$j - 6</code>
*	Multiplication	<code>\$j * 11</code>
/	Division	<code>\$j / 4</code>
%	Modulus (division remainder)	<code>\$j % 9</code>
++	Increment	<code>++\$j</code>
--	Decrement	<code>--\$j</code>

Gambar 9.3. Operator Aritmatika (Nixon, 2014)

2. Operator Penugasan

Operator penugasan digunakan untuk memberikan sebuah nilai kepada sebuah variabel, pada gambar 9.4 dapat dilihat jenis-jenis dan contoh dari operator penugasan.

Operator	Example	Equivalent to
=	\$j = 15	\$j = 15
+=	\$j += 5	\$j = \$j + 5
-=	\$j -= 3	\$j = \$j - 3
*=	\$j *= 8	\$j = \$j * 8
/=	\$j /= 16	\$j = \$j / 16
.=	\$j .= \$k	\$j = \$j . \$k
%=	\$j %= 4	\$j = \$j % 4

Gambar 9.4. Operator Penugasan (Nixon, 2014)

3. Operator Perbandingan

Operator perbandingan umumnya digunakan dalam konstruksi seperti pernyataan if di mana perbandingan dua item diperlukan. Misalnya, kita ingin mengetahui apakah sebuah variabel yang telah diincrement mencapai nilai tertentu, atau apakah variabel lain lebih kecil dari suatu nilai yang ditentukan, dan sebagainya.

Perbedaan antara operator = dan == perlu diperhatikan. Operator = adalah operator penugasan, sedangkan operator == adalah operator perbandingan. Meskipun programmer yang lebih berpengalaman kadang-kadang bisa tertukar saat menulis kode dengan terburu-buru, maka perlu untuk berhati-hati.

Operator	Description	Example
==	Is equal to	\$j == 4
!=	Is not equal to	\$j != 21
>	Is greater than	\$j > 3
<	Is less than	\$j < 100
>=	Is greater than or equal to	\$j >= 15
<=	Is less than or equal to	\$j <= 8

Gambar 9.5. Operator Penugasan (Nixon, 2014)

4. Operator Logika

Secara umum, kita menggunakan operator logika untuk menggabungkan hasil dari dua operator perbandingan yang ditunjukkan pada bagian sebelumnya. Sebuah operator logika juga dapat menjadi input untuk operator logika lainnya contoh ("Jika waktu lebih dari pukul 12 siang dan sebelum pukul 2 sore), Sebagai aturan umum, jika sesuatu memiliki nilai TRUE atau FALSE, itu dapat menjadi input untuk operator logika. Operator logika mengambil dua input true atau false dan menghasilkan hasil true atau false. pada gambar 9.6 dapat dilihat jenis-jenis dan contoh dari operator penugasan.(Nixon, 2014)

Operator	Description	Example
&&	And	<code>\$j == 3 && \$k == 2</code>
and	Low-precedence and	<code>\$j == 3 and \$k == 2</code>
	Or	<code>\$j < 5 \$j > 10</code>
or	Low-precedence or	<code>\$j < 5 or \$j > 10</code>
!	Not	<code>! (\$j == \$k)</code>
xor	Exclusive or	<code>\$j xor \$k</code>

Gambar 9.6. Operator Logika (Nixon, 2014)

9.4 Struktur Kontrol

PHP mendukung sejumlah konsep pemrograman tradisional untuk mengontrol alur eksekusi program. Pernyataan kondisional, seperti if/else dan switch, memungkinkan program untuk menjalankan potongan kode yang berbeda, atau tidak sama sekali, tergantung pada suatu kondisi. Loop, seperti while dan for, mendukung eksekusi berulang dari segmen kode tertentu. (Tatroe, MacIntyre and Lerdorf, 2014)

9.4.1 Kondisi

1. Pernyataan kondisional mempengaruhi alur program. Mereka memungkinkan Anda untuk mengajukan pertanyaan tentang hal-hal tertentu dan meresponsnya dengan cara yang berbeda. Pernyataan kondisional memiliki peran yang sangat penting dalam halaman web dinamis - yang merupakan tujuan utama penggunaan PHP - karena mempermudah pembuatan output yang berbeda setiap kali halaman tersebut dilihat.
2. Pernyataan if : berfungsi memeriksa kebenaran suatu ekspresi dan, jika ekspresi tersebut benar, maka sebuah pernyataan dieksekusi. Pernyataan if memiliki format sebagai berikut:
if (expression)
statement
else statement

Contohnya :

```
if($login)
    echo "Selamat Datang!";
else
    echo "Akses ditolak!";
```

Untuk menyertakan lebih dari satu pernyataan dalam sebuah pernyataan if, gunakan blok yang diapit oleh tanda kurung kurawal yang berisi kumpulan pernyataan:

```
if($login){
    echo "Selamat Datang!";
    $salam = true;
} else {
    echo "Akses ditolak!";
    exit;
}
```

PHP juga mendukung pengelompokan pernyataan tanpa tanda kurung kurawal, sebagai pengganti setelah baris if,

diakhiri dengan tanda (:), setelah else juga diakhiri dengan tanda (:) dan untuk mengakhiri blok digunakan kata endif. Contohnya :

```
if($login):  
    echo "Selamat Datang!";  
    $salam = true;  
else:  
    echo "Akses ditolak!";  
    exit;  
endif;
```

3. Pernyataan Switch : Nilai dari sebuah variabel tunggal dapat menentukan salah satu dari beberapa pilihan yang berbeda (misalnya, variabel tersebut menyimpan nama pengguna dan ingin melakukan sesuatu yang berbeda untuk setiap pengguna). Pernyataan switch dirancang untuk situasi seperti ini. Pernyataan switch diberikan sebuah ekspresi dan membandingkan nilainya dengan semua kasus dalam switch; semua pernyataan dalam kasus yang cocok akan dieksekusi, hingga menemukan kata kunci break pertama. Jika tidak ada yang cocok, dan terdapat pernyataan default, semua pernyataan setelah kata kunci default akan dieksekusi, hingga menemukan kata kunci break pertama yang ditemui. Contohnya :

```
switch($nama) {  
    case 'dimas':  
        // lakukan sesuatu  
        break;  
    case 'bambang':  
        // lakukan sesuatu  
        break;  
    case 'bagus':  
        // lakukan sesuatu  
        break;  
    case 'rafa':
```

```

        // lakukan sesuatu
        break;
    default :
        // lakukan sesuatu
}

```

9.4.2 Perulangan

1. Pernyataan While : Bentuk paling sederhana dari perulangan adalah pernyataan while:

`while (expression)statement`

Jika ekspresi dievaluasi sebagai benar (true), pernyataan tersebut dieksekusi, dan kemudian ekspresi dievaluasi kembali (jika masih benar, bagian utama perulangan dieksekusi lagi, dan seterusnya). Perulangan berhenti ketika ekspresi tidak lagi benar, yaitu dievaluasi bernilai salah (false).

Sebagai contoh, berikut ini adalah beberapa kode yang menjumlahkan bilangan bulat dari 1 hingga 10 :

```

$total = 0;
$i = 1;

while ($i <= 10) {
    $total += $i;
    $i++;
}

```

Alternatif bentuk while lainnya :

```

while (expr):
statement;
more statements ;
endwhile;

```

Contohnya :

```
$total = 0;  
$i = 1;  
while ($i <= 10):  
$total += $i;  
$i++;  
endwhile;
```

2. Pernyataan For hampir mirip dengan pernyataan while, walaupun ada sedikit perbedaan yaitu pernyataan for menambahkan ekspresi inisialisasi penghitung dan manipulasi penghitung, serta seringkali lebih singkat dan lebih mudah dibaca daripada perulangan while yang setara. Berikut adalah struktur dari pernyataan for :

```
for (start; condition; increment) { statement(s); }
```

Ekspresi 'start' dievaluasi sekali, pada awal pernyataan for. Setiap kali melalui perulangan, ekspresi 'condition' diuji. Jika bernilai benar, maka bagian utama perulangan dieksekusi; jika bernilai salah, perulangan berakhir. Ekspresi 'increment' dievaluasi setelah bagian utama perulangan dijalankan.

Berikut contoh penggunaan for untuk menghitung penjumlahan angka dari 1 s.d. 10 :

```
$total = 0;  
for ($i= 1; $i <= 10; $i++) {  
$total += $i;  
}
```

3. Pernyataan foreach memungkinkan untuk mengulangi elemen-elemen dalam sebuah array. Untuk melakukan

perulangan pada array dan mengakses nilai pada setiap kunci, gunakan :

```
foreach ($array as $key => $value) {  
    // ...  
}
```

9.5 Fungsi

Keperluan dasar dari setiap bahasa pemrograman meliputi tempat untuk menyimpan data, mekanisme untuk mengontrol aliran program, serta beberapa komponen seperti evaluasi ekspresi, manajemen file, dan output teks. PHP memiliki semua fitur tersebut, ditambah dengan berbagai kondisi dan perulangan yang dapat digunakan. Namun, meskipun kita memiliki semua ini dalam kotak peralatan kita, terkadang pemrograman dapat menjadi tidak efisien, terutama jika kita harus menulis ulang bagian kode yang sangat mirip setiap kali kita membutuhkannya.

Di sinilah peran fungsi dan objek sangat penting. Fungsi merupakan kumpulan pernyataan yang melakukan tugas tertentu dan secara opsional mengembalikan nilai. Kita dapat mengisolasi bagian kode yang sering digunakan ke dalam sebuah fungsi, dan memanggil fungsi tersebut dengan nama ketika kita membutuhkan kode tersebut. (Nixon, 2014)

Fungsi memiliki banyak keuntungan, seperti :

1. Mengurangi pengetikan yang lebih sedikit
2. Mengurangi kesalahan sintaks dan kesalahan pemrograman lainnya
3. Mengurangi waktu pemuatan file program
4. Mengurangi waktu eksekusi, karena setiap fungsi dikompilasi hanya sekali, tidak peduli seberapa sering Anda memanggilnya
5. Menerima argumen dan dapat digunakan untuk kasus umum maupun khusus

Objek mengambil konsep ini lebih jauh. Sebuah objek menggabungkan satu atau lebih fungsi, dan data yang mereka gunakan, ke dalam sebuah struktur tunggal yang disebut kelas.

9.5.1 Deklarasi Fungsi

Struktur dasar dari fungsi sebagai berikut :

```
function function_name([parameter [, ...]])  
{  
    // Statements  
}
```

Penjelasan dari struktur kode di atas :

1. Definisi dimulai dengan kata kunci `function`.
2. Diikuti oleh sebuah nama, yang harus dimulai dengan huruf atau garis bawah, diikuti oleh sejumlah huruf, angka, atau garis bawah.
3. Tanda kurung diperlukan.
4. Satu atau lebih parameter, yang dipisahkan oleh koma, bersifat opsional. (Tatroe, MacIntyre and Lerdorf, 2014)

Nama fungsi bersifat case-insensitive, sehingga semua string berikut dapat merujuk ke fungsi `print`: `PRINT`, `Print`, dan `PrInT`.

Berikut contoh pembuatan fungsi :

```
function halo(){  
    echo 'halo dunia';  
}
```

9.5.2 Pemanggilan Fungsi

Fungsi dalam program PHP dapat menjadi bawaan (atau secara efektif bawaan dengan berada dalam ekstensi) atau ditentukan pengguna. Terlepas dari sumbernya, semua fungsi panggil dengan cara yang sama :

```
$someValue = nama_fungsi( [ parameter, ... ] );
```

Jumlah parameter yang diperlukan oleh sebuah fungsi bervariasi dari fungsi ke fungsi. Parameter yang diberikan kepada fungsi dapat berupa ekspresi yang valid dan harus berada dalam urutan tertentu yang diharapkan oleh fungsi. Jika parameter diberikan secara tidak berurutan, fungsi mungkin tetap berjalan secara kebetulan, tetapi pada dasarnya hal itu berarti "sampah masuk = sampah keluar". Dokumentasi sebuah fungsi akan memberitahu parameter apa yang diharapkan oleh fungsi dan nilai-nilai apa yang dapat diharapkan untuk dikembalikan. (Tatroe, MacIntyre and Lerdorf, 2014)

DAFTAR PUSTAKA

- Asari, A., Mayatopani, H. and others. 2023. *PENGEMBANGAN WEBSITE*. Media Nusa Creative (MNC Publishing).
- Chairis, G. and Maulana, A. 2022. 'Analisis Perancangan dan Implementasi Sistem Informasi Stationary Berbasis Web pada PT. Indako Trading Coy', *Journal Information System Development (ISD)*, 7(2), pp. 78–90.
- Muttaqin, M. *et al.* 2023. *Pengantar Internet*. Yayasan Kita Menulis.
- Nixon, R. 2014. *Learning PHP, MySQL, JavaScript, CSS & HTML5*, O'Reilly Media. Orelly.
- Pangaribuan, Jefri Junifer, Jusin, J., *et al.* 2022. 'Pengembangan Sistem Informasi Penyewaan Kendaraan pada CV. Bita Jaya Mandiri', *ABDIKAN: Jurnal Pengabdian Masyarakat Bidang Sains dan Teknologi*, 1(4), pp. 597–603.
- Pangaribuan, J J *et al.* 2022. 'Pengembangan Sistem Informasi Point of Sales untuk UMKM, Studi Kasus: Arjuna Farm', *Journal Information System Development*, 7(2).
- Pangaribuan, Jefri Junifer, Maulana, A., *et al.* 2022. 'Simpuru: Gamifikasi Pembelajaran Bahasa Jepang dalam Aplikasi Berbasis Web', *Jurnal Teknologi Dan Ilmu Komputer Prima (JUTIKOMP)*, 5(2), pp. 60–65.
- Pratama, Y.A. *et al.* 2022. 'Perancangan Sistem Informasi Pemesanan Tiket Bus pada PT. Putra Pelangi Perkasa', *PaKMas: Jurnal Pengabdian Kepada Masyarakat*, 2(2), pp. 452–460.
- Pratama, Y.A., Lawi, L. and Jusin, J. 2022. 'Pengembangan Website Pencarian Dan Pemesanan Jasa Guru Les Privat Berdasarkan Model C2C Marketplace', *Journal Information System Development (ISD)*, 7(2), pp. 91–105.
- Tatroe, K., MacIntyre, P. and Lerdorf, R. 2014. *Programming PHP, Third Edition*, O'Reilly Media.

The PHP Manual. 2023. Available at: <https://www.php.net/manual/en/intro-what-is.php> (Accessed: 20 June 2023).

BAB 10

PENGELOLAAN FORMULIR

Oleh Emha Diambang Ramadhany

10.1 Definisi

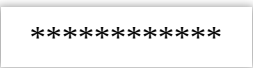
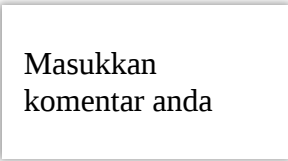
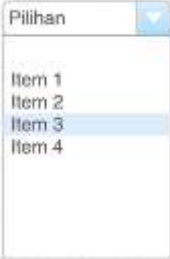
Pengelolaan form pada pemrograman web mengacu pada proses pengumpulan, validasi, pemrosesan, dan respons terhadap data yang dikirimkan oleh pengguna melalui formulir HTML. Ini melibatkan penggunaan teknik dan alat untuk mengelola data formulir dengan aman, efisien, dan akurat. Pengelolaan form meliputi validasi data, perlindungan terhadap serangan keamanan, pengolahan data di sisi server, penyimpanan data, pengiriman email, dan memberikan respons yang tepat kepada pengguna (Oktarini *et al.*, 2019).

10.2 Jenis-Jenis Formulir

Untuk mendapatkan data pada pemrosesan formulir, diperlukan suatu kontrol input dari pengguna. Terdapat beberapa tipe kontrol formulir yang paling sering digunakan pada suatu situs yang mengumpulkan informasi formulir (Duckett, 2011).

Tabel 10.1. Jenis-Jenis Kontrol Formulir

Penambahan Teks		
Input teks (satu baris)	Digunakan untuk teks satu baris seperti nama atau alamat surel	

Input kata sandi	Seperti input teks namun kata yang diinputkan akan diganti dengan tanda lain	
Area teks	Untuk teks yang membutuhkan karakter lebih panjang seperti paragraf, pesan, atau komentar	
Membuat Pilihan		
Radio Button	Memberikan hanya satu pilihan dari sekian opsi	
Checkboxes	Memberikan pengguna satu atau lebih pilihan pada formulir	
Drop-down	Memberikan pengguna satu pilihan dari beberapa daftar pilihan	
Mengumpulkan Formulir		
Submit	Tombol untuk mengirimkan	

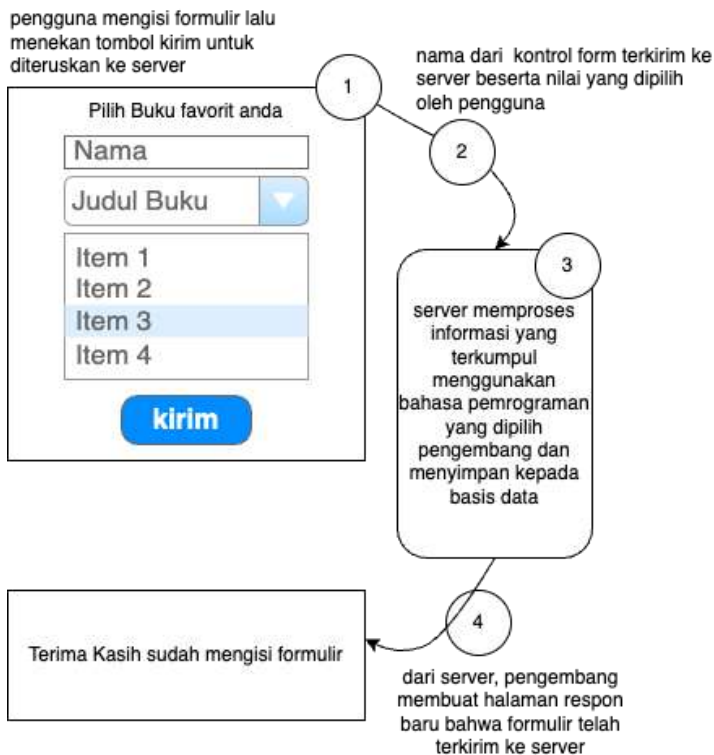
	semua data yang ada di formulir yang telah diisikan	
Unggah File	Tombol untuk mengunggah file yang ada di komputer atau direktori lain supaya masuk ke formulir	browse *file maks 2mb

Sumber: Oktarini et al., 2019

10.3 Cara Kerja Formulir

Cara kerja formulir pada pemrograman web melibatkan serangkaian langkah yang terjadi antara pengguna dan server. Pengguna mengakses halaman web yang berisi formulir dan mengisi input dengan data yang diminta, seperti nama, email, atau pesan. Setelah pengguna selesai mengisi formulir, mereka mengklik tombol "Kirim" atau tindakan serupa yang ditentukan dalam formulir. Ini akan memicu pengiriman formulir ke server. Data formulir dikirim dari peramban pengguna ke server menggunakan permintaan HTTP. Metode pengiriman data, seperti POST atau GET, ditentukan dalam atribut method pada elemen `<form>`. Server menerima data formulir yang dikirimkan dari peramban pengguna. Data ini akan tersedia untuk diproses oleh skrip server. Pada tahap ini, server biasanya akan melakukan validasi data yang diterima dari formulir. Validasi ini melibatkan pemeriksaan kebenaran, integritas, dan format data yang dikirimkan. Server akan memastikan bahwa data yang diterima sesuai dengan aturan yang ditentukan. Setelah validasi data, server akan memproses data sesuai dengan kebutuhan. Ini bisa berarti menyimpan data

ke dalam database, mengirim email, menghitung hasil, atau melakukan tindakan lain berdasarkan data yang diterima. Setelah pemrosesan selesai, server akan mengirimkan respons kembali ke peramban pengguna. Respons ini dapat berupa halaman terima kasih, pesan sukses atau kesalahan, atau pengalihan ke halaman lain. eramban pengguna menerima respons dari server dan menampilkan respons tersebut kepada pengguna. Pengguna akan melihat pesan atau tindakan yang sesuai berdasarkan respons yang diterima. Adapun ilustrasi tersebut tergambar pada gambar 10.1.



Gambar 10.1. Cara Kerja Pengelolaan Formulir
(sumber diolah : Oktarini et al., 2019)

10.4 Metode Pengiriman Data Formulir

Untuk memproses suatu form yang dikirimkan ke server, form memiliki struktur supaya dapat lebih maksimal dan diterima oleh server dengan bantuan *script*. Pada kasus ini menggunakan php sebagai bahasa pemrogramannya.

Merancang sebuah inputan form, terdapat tiga hal yang harus diperhatikan (Solichin, 2016; Welling & Thomson, 2017), antara lain :

1. Method

metode (method) adalah atribut yang didefinisikan pada elemen `<form>` yang menentukan cara pengiriman data formulir ke server saat pengguna mengirimkan formulir. Terdapat dua metode yang umum digunakan: GET dan POST.

a) Metode GET:

- Metode **GET** mengirimkan data formulir sebagai parameter yang terlihat dalam URL (gambar 10.2).
- Data formulir dikirim sebagai query string setelah tanda tanya (?) dalam URL.
- Contoh URL dengan metode GET: `http://www.example.com/proses.php?nama=John&email=john@example.com`
- Penggunaan metode GET cocok untuk permintaan yang bersifat idempoten, artinya melakukan permintaan yang sama multiple kali tidak akan menghasilkan perubahan data yang tidak diinginkan.
- Digunakan saat pengiriman data formulir bersifat non-sensitive dan tidak mengubah data yang ada di server.



Gambar 10.2. Method GET Pada Form

Sumber : (Fadil, 2020)

b) Metode POST:

- Metode POST mengirimkan data formulir sebagai bagian dari body permintaan HTTP.
- Data formulir tidak terlihat dalam URL, sehingga lebih aman untuk pengiriman data sensitif.
- Data dikirimkan dalam format yang tidak terlihat oleh pengguna biasa.
- Penggunaan metode POST cocok untuk permintaan yang mengubah atau mengirimkan data yang signifikan ke server, seperti saat menyimpan data ke dalam database.
- Penggunaan metode POST juga dapat digunakan untuk mengunggah file melalui formulir.



Gambar 10.3. Method POST Pada Form

Sumber : (Fadil, 2020)

2. Action

Atribut pada elemen `<form>` yang menentukan URL atau file skrip server yang akan menangani data formulir setelah pengguna mengirimkannya. Atribut action menentukan tempat tujuan pengiriman data formulir.

Jika action kosong (gambar 10.3), biasanya pengembang membuat pemrosesan form di halaman yang sama. Sebaliknya, *script form* dan pemrosesan dibuat berbeda halaman.

Form action harus berisikan nama file yang akan dituju untuk memproses suatu form yang telah diisi pengguna.

3. Tombol Kirim

tombol kirim merupakan *trigger* atau pemicu supaya form yang telah diisi akan diproses dan lanjut ke proses selanjutnya.



Gambar 10.4. Struktur Komponen Pengiriman Data Form

Terdapat dua metode pengiriman *form* pada pengembangan web. Pertama, *script* pemroses yang dipisahkan dari *script* tampilan (Gannev et al., 2014). Kedua, *Script* pemroses di halaman yang sama dengan *script* tampilan.

Contoh dari *script* pemroses dan tampilan ketika berada dalam satu file adalah seperti berikut. saat ini ilustrasi dinamakan dengan *formget.php* dan tampilan pada gambar 10.2.

```
<!DOCTYPE html>
<html>
  <head>
```

```

<title>Coba GET</title>
</head>
<body>
  <fieldset>
    <form action="#" method="GET">
      <label>Masukan Nama</label>
      <input type="text" name="nama">
      <input type="submit" value="submit" />
    </fieldset>
  </form>
</body>
<?php
if (isset($_GET["nama"])) {
  echo "Nama : " . $_GET["nama"];
}
?>
</html>

```

Contoh dari script pemroses dan tampilan ketika file dipisahkan adalah seperti berikut.

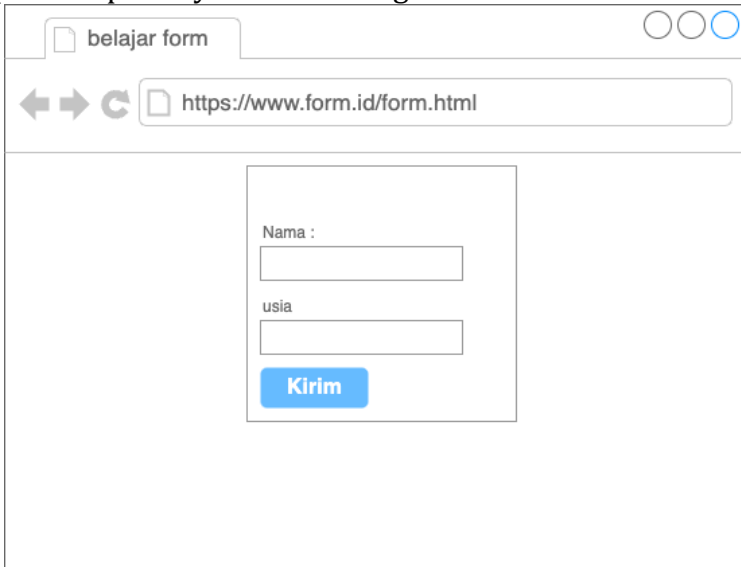
File form.html

```

<form method="post" action="tampilkan.php">
  <label>Input Nama</label><br/>
  <input type="text" name="nama"><br/>
  <label>Input Usia</label><br/>
  <input type="text" name="usia"><br/>
  <input type="submit" value="kirim">
</form>

```

Adapun tampilannya adalah sebagai berikut:

A screenshot of a web browser window. The title bar shows a tab labeled 'belajar form'. The address bar displays the URL 'https://www.form.id/form.html'. The main content area shows a simple form with two text input fields. The first field is preceded by the label 'Nama :'. The second field is preceded by the label 'usia'. Below these fields is a blue button with the text 'Kirim' in white.

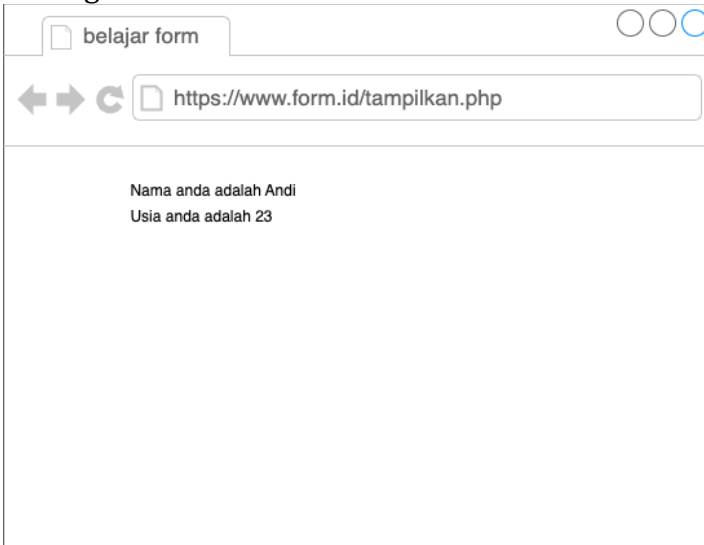
Gambar 10.5. Tampilan form.html

Untuk memproses *form* tersebut dibutuhkan *file* baru bernama tampil.php supaya isian yang telah dikumpulkan akan ditampilkan. nama *file* tampil.php harus dicantumkan pada tag action di *script* form.html.

File tampil.php

```
<?php
$nama = $_POST['nama'];
$usia = $_POST['usia'];
echo "Nama anda adalah " . $nama;
echo "<br/>";
echo "Usia anda adalah " . $usia;
?>
```

Adapun tampilan yang dihasilkan dari *script* tersebut adalah sebagai berikut :



Gambar 10.6. Tampilan tampilan.php

10.5 Membuat Formulir dengan HTML

Pada subbab ini akan dijabarkan cara membuat form kontrol yang telah dijelaskan pada sub-bab. 10.2 dengan HTML.

10.5.1 Elemen *Form*

Elemen *form* HTML digunakan sebagai wadah untuk membuat formulir. Wadah Ini dapat berisikan kotak teks, radio, checkbox, dan lain sebagainya. elemen form HTML menggunakan tag html seperti berikut :

```
<form>
.
form_elements
.
</form>
```

10.5.2 Elemen Input

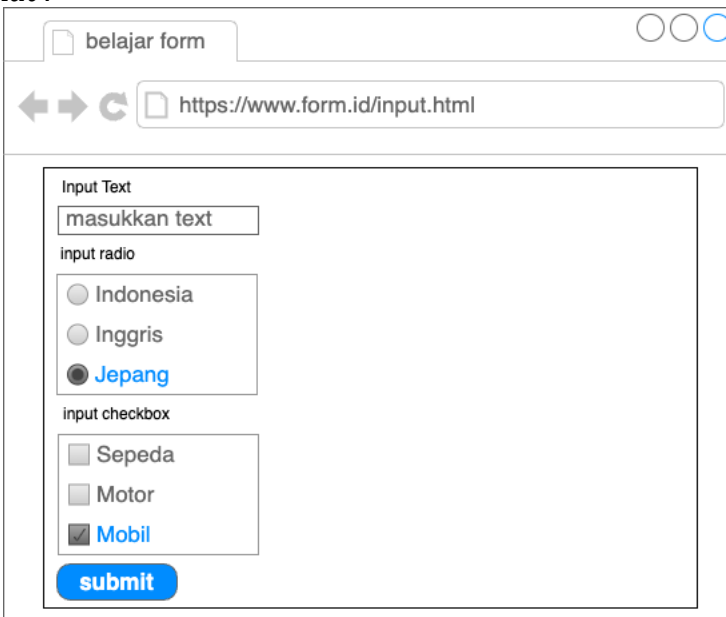
Elemen `<input>` adalah salah satu elemen HTML yang paling umum digunakan dalam pembuatan formulir pada pemrograman web. Elemen ini digunakan untuk menerima input dari pengguna dalam berbagai bentuk, seperti teks, angka, *checkbox*, *radio button*, atau pengiriman file. Elemen input dapat digunakan secara bersama-sama. Berikut adalah beberapa atribut yang umum digunakan dalam elemen `<input>` jika digunakan bersamaan pada satu file :

Form input.html

```
<form>
  <label for="fname">input text</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="fname">input radio</label><br>
  <input type="radio" id="indonesia" name="bahasa"
value="indonesia">
  <label for="indonesia">Indonesia</label><br>
  <input type="radio" id="inggris" name="bahasa"
value="inggris">
  <label for="inggris">Inggris</label><br>
  <input type="radio" id="jepang" name="bahasa"
value="jepang">
  <label for="jepang">Jepang</label></br>
  <label for="fname">input checkbox</label><br>
  <input type="checkbox" id="sepeda" name="sepeda"
value="sepeda">
  <label for="Sepeda"> Sepeda </label><br>
  <input type="checkbox" id="motor" name="motor"
value="motor">
  <label for="motor"> Motor </label><br>
```

```
<input type="checkbox" id="mobil" name="mobil" value="mobil">  
<label for="mobil"> Mobil</label></br>  
<input type="submit" value="Submit">  
</form>
```

Adapun tampilan dari hasil *script* di atas adalah sebagai berikut :



The screenshot shows a web browser window with the title 'belajar form' and the address bar displaying 'https://www.form.id/input.html'. The main content area contains a form with the following elements:

- Input Text:** A text input field with the placeholder text 'masukkan text'.
- input radio:** Three radio button options: 'Indonesia', 'Inggris', and 'Jepang'. The 'Jepang' option is selected.
- input checkbox:** Three checkbox options: 'Sepeda', 'Motor', and 'Mobil'. The 'Mobil' option is checked.
- submit:** A blue button labeled 'submit'.

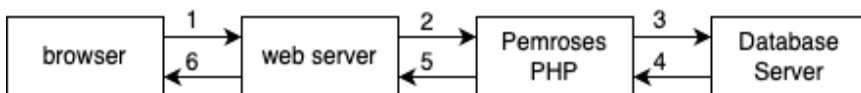
Gambar 10.7. Tampilan input.html

10.6 Pengelolaan Form Dengan *Database*

Pengelolaan form dengan database melibatkan proses menyimpan data formulir yang dikirim oleh pengguna ke dalam database. Adapun proses umum pengelolaan formulir ketika menyambungkan aplikasi dengan sistem basis data adalah sebagai berikut :

1. Membuat struktur basis data hingga tabel sesuai dengan kaidah teknis analisis basis data. setiap kolom dalam tabel akan mewakili inputan pada formulir
2. Membuat koneksi antara basis data dan web menggunakan bahasa pemrogram yang dipilih.
3. Setelah pengguna mengirimkan formulir, skrip server akan menerima data yang dikirim melalui metode POST atau GET.
4. Lakukan validasi data untuk memastikan bahwa data yang diterima dari formulir sesuai dengan aturan yang ditetapkan. Validasi dapat mencakup pemeriksaan data kosong, tipe data, panjang, format, atau aturan khusus lainnya.
5. Setelah data formulir tervalidasi, gunakan perintah SQL (*Structured Query Language*) untuk menyimpan data ke dalam tabel database. Biasanya, ini melibatkan perintah INSERT untuk memasukkan data baru ke dalam tabel.

Ilustrasi alur pemrosesan form dengan basis data dapat dipaparkan pada gambar 10.8.



Gambar 10.8. Alur Pemrosesan Form Dengan Basis Data
Sumber : Welling & Thomson, (2017)

10.7 Studi Kasus Pembuatan Formulir

Pada kasus ini kita akan berlatih membuat aplikasi sederhana buku tamu. penulis memberikan contoh dengan memberikan tiga jenis file yaitu, a) index.html sebagai halaman pengisian formulir, b) formsubmit.php sebagai pemroses form dan menyimpannya ke sistem basis data, dan c) respon.html

sebagai halaman validasi bahwa formulir telah tersimpan di basis data.

File index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Buku Tamu</title>
</head>
<body>
  <h1>Buku Tamu</h1>
  <form action="formsubmit.php" method="POST">
    <label for="nama">Nama:</label>
    <input type="text" id="nama" name="nama"
required><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"
required><br><br>
    <label for="pesan">Pesan:</label>
    <textarea id="pesan" name="pesan"
required></textarea><br><br>
    <input type="submit" value="Kirim">
  </form>
</body>
</html>
```

File formsubmit.php

```
<?php
// Koneksi ke database
$servername = "localhost";
```

```

$username = "username";
$password = "password";
$dbname = "nama_database";

$conn = new mysqli($servername, $username, $password,
$dbname);
if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}

// Mendapatkan data dari formulir
$name = $_POST['nama'];
$email = $_POST['email'];
$message = $_POST['pesan'];

// Insert data ke tabel buku_tamu
$sql = "INSERT INTO buku_tamu (nama, email, pesan) VALUES
('$name', '$email', '$message')";

if ($conn->query($sql) === TRUE) {
    header("Location: respon.html"); // Mengarahkan ke halaman
    respon.html
    exit();
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>

```

File respon.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Terima Kasih</title>
</head>
<body>
  <h1>Terima Kasih Telah Mengisi Buku Tamu</h1>
  <p>Terima kasih atas kunjungan Anda dan pesan yang Anda
sampaikan. Kami sangat menghargainya.</p>
</body>
</html>
```

Pastikan untuk mengganti nilai "*localhost*", "*username*", "*password*", dan "*nama_database*" sesuai dengan konfigurasi MySQL server Anda. Juga pastikan Anda telah membuat tabel "*buku_tamu*" dengan kolom yang sesuai (nama, email, pesan) di database yang ditentukan.

DAFTAR PUSTAKA

- Duckett, J. 2011. *HTML & CSS Design and Build Websites*. John Wiley & Sons, Inc.
- Fadil. 2020, November 26. *Perbedaan Method GET dan POST*. <https://kotakode.com/blogs/2718/Perbedaan-Method-GET-dan-POST>
- Ganney, P. S., Pisharody, S., & McDonagh, E. 2014. Web Development. *Clinical Engineering: A Handbook for Clinical and Biomedical Engineers*, 171–190. <https://doi.org/10.1016/B978-0-12-396961-3.00010-X>
- Oktarini, A., Ari, S., & Sunarti, A.; 2019. *web programming*. graha ilmu.
- Solichin, A. 2016. *Pemrograman Web dengan PHP dan MySQL* (first). budi luhur. <http://achmatim.net>
- Welling, L., & Thomson, L. 2017. *PHP and MySQL Web Development, Fifth Edition* (fifth). pearson education.

BAB 11

KEAMANAN DAN PERLINDUNGAN TERHADAP SERANGAN

Oleh Suwito Pomalingo

11.1 Pengantar Keamanan Web

11.1.1 Pentingnya Keamanan Dalam Pengembangan Web

Keamanan merupakan aspek yang krusial dalam pengembangan web modern. Dalam era di mana serangan terhadap situs web dan aplikasi web semakin kompleks, penting bagi pengembang web untuk mengutamakan keamanan. Seperti yang disebutkan oleh OWASP (*Open Web Application Security Project*), "Keamanan harus menjadi bagian integral dari setiap siklus pengembangan perangkat lunak, termasuk pengembangan web" (Turpin and Gadsden, 2010). Melalui penerapan praktik keamanan yang tepat, pengembang dapat melindungi data pengguna, mencegah kerentanan perangkat lunak, dan meminimalkan risiko serangan.

Perlindungan terhadap serangan pada sistem web adalah penting untuk menjaga integritas dan privasi data pengguna. Dalam laporan Tahun 2023, Verizon menemukan bahwa serangan siber pada aplikasi web merupakan salah satu penyebab utama pelanggaran data (Hylender *et al.*, 2023). Dengan menerapkan keamanan yang kuat, seperti penggunaan enkripsi data, validasi input yang ketat, dan pengelolaan otentikasi yang aman, pengembang dapat memastikan bahwa data pengguna terlindungi dan privasi mereka dijaga.

Selain melindungi pengguna, keamanan dalam pengembangan web juga berdampak pada reputasi dan keberlanjutan bisnis. Menurut studi yang dilakukan oleh

Accenture, hampir 68% konsumen akan kehilangan kepercayaan pada sebuah perusahaan jika data pribadi mereka bocor akibat kelemahan keamanan (Baumgarten, Socher and Yule, 2023). Keamanan yang lemah dapat menyebabkan kerugian finansial yang signifikan akibat kehilangan pelanggan dan kerugian reputasi bisnis. Selain itu, dalam rangka meningkatkan kualitas keamanan informasi internal, setiap perusahaan perlu memprioritaskan faktor seperti pendidikan dan pelatihan keamanan, serta kesadaran akan keamanan informasi (Fianty, 2023). Oleh karena itu, investasi dalam keamanan web tidak hanya melindungi pengguna, tetapi juga merupakan langkah strategis untuk menjaga kelangsungan bisnis.

11.1.2 Ancaman Umum Terhadap Situs Web Dan Aplikasi

Serangan DDoS (*Distributed Denial of Service*) merupakan salah satu ancaman umum terhadap situs web dan aplikasi. Serangan ini bertujuan untuk mengganggu ketersediaan suatu layanan dengan membanjiri server dengan lalu lintas yang tinggi. Menurut laporan yang diterbitkan oleh Akamai Technologies, serangan DDoS mengalami peningkatan signifikan dalam skala, durasi, dan kompleksitasnya (McKeay, 2020). Serangan DDoS dapat menyebabkan kerugian finansial yang besar akibat penurunan produktivitas, kehilangan pelanggan, dan kerusakan reputasi perusahaan. Perlindungan yang efektif melawan serangan DDoS melibatkan penggunaan solusi mitigasi yang canggih, seperti firewall yang dapat menangani lalu lintas jaringan yang tinggi.

Serangan XSS (*Cross-Site Scripting*) juga termasuk dalam ancaman umum terhadap situs web dan aplikasi. Serangan ini memanfaatkan celah keamanan dalam aplikasi web untuk menyisipkan skrip berbahaya ke dalam halaman web yang dilihat oleh pengguna. Menurut jurnal yang diterbitkan dalam *International Journal of Computer Applications*, serangan XSS

masih menjadi salah satu ancaman yang paling umum terhadap aplikasi web saat ini (Rodríguez *et al.*, 2019). Serangan XSS dapat memungkinkan penyerang untuk mencuri informasi pengguna, mengubah konten halaman web, atau melakukan aksi jahat lainnya. Untuk melindungi situs web dan aplikasi dari serangan XSS, penting untuk melaksanakan sanitasi data yang ketat dan mengimplementasikan mekanisme validasi input yang kuat.

Ancaman umum lainnya adalah serangan peretasan basis data, seperti serangan SQL Injection. Serangan SQL Injection melibatkan penyisipan kode SQL berbahaya ke dalam input pengguna yang tidak divalidasi dengan benar. Menurut laporan yang diterbitkan dalam *Journal of Network and Computer Applications*, serangan SQL Injection masih menjadi salah satu serangan yang paling berbahaya dan merusak pada aplikasi web (Abdullayev and Chauhan, 2023). Penyerang yang berhasil melakukan serangan ini dapat memperoleh akses tidak sah ke basis data, mengambil, memodifikasi, atau menghapus data, serta melanggar privasi pengguna. Melindungi situs web dan aplikasi dari serangan SQL Injection melibatkan penggunaan parameterisasi query, validasi input yang ketat, dan penggunaan prepared statement.

11.1.3 Konsep Dasar Keamanan Web

Keamanan web adalah area penting dalam pengembangan aplikasi web. Konsep-konsep dasar keamanan web membantu mencegah dan mengurangi risiko serangan yang dapat merusak sistem atau mencuri data. Pada dasarnya, ada tiga aspek utama dalam keamanan web, yaitu kerahasiaan, integritas, dan ketersediaan (Triad CIA) (Schneier, 1996). Kerahasiaan melindungi informasi dari akses yang tidak berwenang. Integritas menjamin bahwa informasi hanya diubah oleh pihak yang berwenang dan melalui metode yang

sah. Ketersediaan memastikan bahwa informasi dan sistem selalu tersedia untuk pihak yang berwenang saat dibutuhkan.

Lebih lanjut, otentikasi dan otorisasi juga menjadi komponen penting dalam keamanan web. Otentikasi adalah proses verifikasi identitas pengguna, sementara otorisasi adalah penentuan akses apa yang dimiliki pengguna setelah identitasnya diverifikasi (Stallings and Brown, 2018). Konsep-konsep ini memungkinkan sistem untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses dan memanipulasi data.

Pemahaman tentang berbagai jenis ancaman dan serangan juga menjadi bagian penting dari keamanan web. Ini termasuk, tetapi tidak terbatas pada, serangan seperti *Cross-Site Scripting* (XSS), serangan injeksi SQL, dan serangan *Denial of Service* (DoS). Setiap serangan ini mengancam aspek-aspek keamanan web dan memerlukan strategi perlindungan dan mitigasi yang berbeda (Li *et al.*, 2022).

11.2 Serangan Web Umum

11.2.1 *Cross-Site Scripting* (XSS)

Cross-Site Scripting (XSS) adalah salah satu serangan web paling umum yang ditujukan pada aplikasi web. XSS terjadi ketika penyerang berhasil menyisipkan skrip berbahaya ke dalam halaman web yang kemudian dieksekusi oleh browser pengguna. Seperti yang dijelaskan oleh OWASP, organisasi terkemuka di bidang keamanan web, serangan XSS dapat mengakibatkan berbagai masalah, seperti pencurian identitas pengguna, pengambilalihan sesi, dan defacement situs web (Rodríguez *et al.*, 2019).

Ada tiga jenis utama dari serangan XSS, yaitu *Reflected* XSS, *Stored* XSS, dan *DOM-based* XSS. *Reflected* XSS terjadi ketika skrip berbahaya disisipkan melalui URL dan dieksekusi saat pengguna mengunjungi URL tersebut. *Stored* XSS, juga

dikenal sebagai Persistent XSS, terjadi ketika skrip berbahaya disimpan di server dan dieksekusi setiap kali halaman diakses. Sementara itu, DOM-based XSS terjadi ketika skrip berbahaya memanipulasi *Document Object Model* (DOM) dari halaman web (Kirda *et al.*, 2006).

Pencegahan XSS umumnya melibatkan penerapan teknik sanitasi dan encoding output. Sanitasi bertujuan untuk menghapus atau mengubah input yang berpotensi berbahaya, sedangkan encoding output menjadikan skrip berbahaya tidak dieksekusi oleh browser. Selain itu, *HTTP-only cookies* dan fitur keamanan web modern seperti *Content Security Policy* (CSP) juga dapat digunakan untuk melindungi terhadap serangan XSS (Michal Zalewski, 2011).

11.2.2 Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) merupakan serangan yang memanfaatkan kepercayaan sebuah situs pada browser pengguna. Serangan ini memaksa pengguna yang telah terotentikasi untuk melakukan aksi tanpa sepengetahuan mereka. Menurut *Open Web Application Security Project* (OWASP), CSRF termasuk dalam sepuluh ancaman keamanan aplikasi web teratas (Turpin and Gadsden, 2010).

Prinsip dasar CSRF adalah memanipulasi pengguna untuk melakukan permintaan yang tidak mereka inginkan atau tidak mereka sadari. Sebagai contoh, pengguna yang telah login di sebuah bank online mungkin tanpa sadar mengklik tautan jahat yang menyebabkan transaksi tidak diinginkan. Hal ini terjadi karena browser secara otomatis melampirkan cookie session yang autentik dengan setiap permintaan ke situs, sehingga situs percaya bahwa permintaan tersebut berasal dari pengguna yang sah. Selain itu, karena serangan ini umumnya tidak memerlukan interaksi pengguna, ia seringkali sulit dideteksi (Barth, Jackson and Mitchell, 2008).

Pencegahan CSRF melibatkan beberapa pendekatan. Salah satu metode yang paling efektif adalah penggunaan token CSRF, yaitu token unik yang dihasilkan oleh server dan disertakan dalam setiap formulir atau permintaan AJAX. Token ini kemudian diperiksa oleh server pada setiap permintaan berikutnya untuk memastikan bahwa permintaan tersebut memang berasal dari pengguna yang sah (Zeller, W. and Felten, 2008) Selain itu, fitur keamanan seperti SameSite cookies dan fitur browser modern lainnya dapat digunakan untuk melindungi terhadap serangan CSRF.

11.2.3 Injection Attacks (SQL, Command, dll.)

Serangan injeksi merupakan jenis serangan keamanan yang melibatkan penyalahgunaan sistem oleh penyerang melalui inputan yang tidak aman. Dalam konteks web, serangan injeksi paling sering diasosiasikan dengan SQL Injection, namun ada juga jenis serangan injeksi lainnya seperti Command Injection dan LDAP Injection.

SQL Injection adalah serangan injeksi yang paling umum dan berpotensi merusak. Serangan ini terjadi ketika penyerang mampu menyisipkan atau "menginjeksi" SQL query yang berbahaya melalui inputan pengguna ke dalam sistem database. Hasilnya, penyerang bisa memperoleh akses yang tidak sah, memanipulasi, atau bahkan menghapus data dalam basis data (Halfond, Viegas and Orso, 2006).

Command Injection adalah serangan injeksi lainnya yang sering terjadi. Dalam serangan ini, penyerang menyisipkan perintah sistem yang akan dieksekusi oleh server. Penyerang bisa mendapatkan kontrol penuh atas sistem dengan memanfaatkan celah ini (Zhong, no date).

Untuk mencegah serangan injeksi, input dari pengguna harus selalu divalidasi, difilter, dan disanitasi sebelum digunakan dalam query atau perintah. Penggunaan prepared statements atau parameterized queries juga sangat disarankan

untuk mencegah SQL Injection. Selain itu, prinsip hak akses minimum (*principle of least privilege*) juga harus diterapkan, di mana setiap proses hanya diberikan hak akses minimum yang diperlukan untuk menjalankan tugasnya (Abikoye *et al.*, 2020).

11.2.4 Session Hijacking dan Session Fixation

Session Hijacking dan Session Fixation adalah dua jenis serangan yang berfokus pada pencurian atau manipulasi sesi pengguna dalam aplikasi web. Keduanya bertujuan untuk mendapatkan akses tidak sah ke akun pengguna atau data sensitif.

Session Hijacking, juga dikenal sebagai cookie hijacking, adalah serangan di mana penyerang mencoba mencuri cookie sesi dari pengguna untuk memalsukan identitas mereka dan mendapatkan akses ke aplikasi web. Dalam serangan ini, penyerang mencoba mendapatkan akses ke cookie sesi melalui metode seperti XSS atau sniffing jaringan. Setelah berhasil mendapatkan cookie sesi, penyerang dapat menggunakannya untuk meniru pengguna yang sah (Ogundele, Akinade and Alakiri, 2020).

Session Fixation adalah serangan di mana penyerang memaksa pengguna untuk menggunakan session ID yang telah ditentukan oleh penyerang. Dengan melakukan ini, setelah pengguna masuk, penyerang sudah memiliki session ID yang valid dan dapat mengakses akun pengguna. Dalam serangan ini, penyerang biasanya akan mencoba memanipulasi aplikasi web atau pengguna untuk menerima session ID yang telah ditentukan sebelumnya (Johns, 2006).

Untuk mencegah serangan ini, beberapa praktik keamanan dapat diimplementasikan. Untuk melindungi dari Session Hijacking, penggunaan HTTPS secara luas dapat mencegah penyerang melakukan sniffing jaringan. Selain itu, penggunaan atribut HttpOnly pada cookie dapat mencegah serangan XSS dari mengakses cookie. Sementara itu, untuk

mencegah Session Fixation, aplikasi web harus selalu memberikan session ID baru setelah setiap autentikasi berhasil.

11.2.5 Brute-Force Attacks

Serangan Brute-Force adalah teknik serangan yang berupaya mendapatkan akses ke sistem dengan menebak berbagai kombinasi kemungkinan hingga menemukan yang benar. Pada konteks aplikasi web, serangan ini biasanya ditujukan untuk memecahkan sandi pengguna atau token autentikasi. Penyerang biasanya menggunakan program atau skrip otomatis untuk mencoba berbagai kombinasi sandi atau token sampai berhasil (Yan *et al.*, 2004).

Terdapat beberapa variasi dari serangan Brute-Force. Salah satunya adalah serangan Dictionary, di mana penyerang menggunakan daftar kata-kata yang umum digunakan sebagai sandi (seperti "password123" atau "qwerty") untuk menebak sandi pengguna. Metode lain adalah serangan Rainbow Table, yang melibatkan penggunaan tabel precomputed untuk mencari padanan teks asli dari hash sandi.

Untuk mencegah serangan Brute-Force, beberapa mekanisme keamanan dapat diimplementasikan. Salah satunya adalah penggunaan CAPTCHA setelah beberapa kali percobaan login gagal, untuk mencegah program otomatis melakukan percobaan login. Selain itu, pembatasan jumlah percobaan login dalam periode waktu tertentu juga dapat efektif mengurangi risiko serangan Brute-Force. Penggunaan sandi yang kuat, panjang, dan tidak mudah ditebak juga sangat penting untuk mencegah serangan ini.

11.2.6 Man-in-the-Middle (MitM) Attacks

Serangan *Man-in-the-Middle* (MitM) adalah jenis serangan di mana penyerang memposisikan diri di tengah-tengah komunikasi antara dua pihak, memungkinkan

penyerang untuk mencegah, memanipulasi, atau memata-matai komunikasi. Dalam konteks web, serangan ini bisa terjadi ketika penyerang berhasil mencegah dan memodifikasi komunikasi antara pengguna dan server (Dierks and Rescorla, 2008).

Dalam serangan MitM, penyerang dapat melihat dan mencuri data yang dikirim antara dua pihak, seperti informasi login, data kartu kredit, atau informasi pribadi lainnya. Selain itu, penyerang juga bisa memanipulasi data yang dikirim, seperti mengubah isi pesan, memasukkan malware, atau mengubah tujuan transaksi.

Untuk mencegah serangan MitM, penggunaan HTTPS secara luas sangat dianjurkan. HTTPS memastikan bahwa komunikasi antara pengguna dan server dienkripsi dan tidak bisa dibaca atau dimodifikasi oleh pihak ketiga. Selain itu, penggunaan teknologi seperti HSTS (*HTTP Strict Transport Security*) dan pinning sertifikat juga bisa membantu mencegah serangan MitM. Fitur ini memastikan bahwa browser hanya akan berkomunikasi dengan server menggunakan koneksi yang aman.

11.2.7 Denial of Service Attacks

Serangan *Denial of Service* (DoS) adalah jenis serangan yang bertujuan untuk membuat suatu sistem, layanan, atau infrastruktur menjadi tidak tersedia bagi pengguna yang seharusnya berhak menggunakannya. Pada konteks aplikasi web, ini bisa dilakukan dengan berbagai cara, misalnya dengan menghabiskan sumber daya server atau membanjiri jaringan dengan lalu lintas yang berlebihan (Mirkovic and Reiher, 2004). Dalam varian yang lebih ekstrem, serangan *Distributed Denial of Service* (DDoS) melibatkan banyak sistem yang dikendalikan oleh penyerang (biasanya melalui botnet) untuk membanjiri target dengan lalu lintas yang besar hingga melebihi kapasitas

target dan membuatnya tidak bisa melayani pengguna sah (Sterbenz *et al.*, 2002).

Mencegah dan memitigasi serangan DoS dan DDoS bisa menjadi tantangan, karena penyerang biasanya menggunakan IP yang dipalsukan atau sistem yang dikendalikan tanpa sepengetahuan pemiliknya. Namun, beberapa metode dapat dilakukan, seperti membatasi jumlah koneksi atau permintaan dari satu sumber dalam periode waktu tertentu, atau menggunakan layanan mitigasi DDoS yang dapat menyerap dan menyaring lalu lintas yang mencurigakan.

11.3 Perlindungan Terhadap Serangan XSS

11.3.1 Menyaring Input Pengguna

Salah satu metode terpenting dalam mencegah serangan *Cross-Site Scripting* (XSS) adalah dengan melakukan penyaringan input pengguna. Input pengguna adalah data yang diberikan oleh pengguna kepada sistem, misalnya melalui form, URL, atau cookies. Jika tidak disaring dengan benar, input pengguna dapat digunakan oleh penyerang untuk menyisipkan skrip berbahaya yang dapat dieksekusi oleh browser pengguna. Penyaringan input pengguna biasanya dilakukan dengan melakukan sanitasi dan validasi. Sanitasi input melibatkan pembersihan data input untuk menghilangkan atau mengubah karakter atau elemen yang tidak diinginkan atau berpotensi berbahaya. Sebagai contoh, karakter seperti '<' dan '>' yang sering digunakan dalam skrip HTML dapat diganti dengan entitas HTML yang sesuai untuk mencegah mereka dieksekusi oleh browser. Sementara itu, validasi input adalah proses memeriksa apakah data yang dimasukkan pengguna sesuai dengan format atau kriteria yang diharapkan.

Namun, hanya mengandalkan penyaringan input mungkin tidak cukup untuk melindungi aplikasi web dari serangan XSS. Fitur keamanan lain seperti penggunaan

HTTPOnly flag pada cookies dan penerapan *Content Security Policy* (CSP) juga sangat penting dalam mencegah serangan XSS. Namun, penyaringan dan sanitasi input tetap menjadi langkah awal yang penting dalam perlindungan terhadap serangan XSS.

11.3.2 Penggunaan Fungsi Sanitasi Data

Fungsi sanitasi data merupakan komponen krusial dalam mencegah serangan *Cross-Site Scripting* (XSS). Proses ini melibatkan penggunaan fungsi atau metode khusus untuk membersihkan data input dari karakter atau elemen yang tidak diinginkan atau berpotensi berbahaya. Tujuannya adalah untuk mencegah eksekusi kode berbahaya yang mungkin disisipkan oleh penyerang melalui input pengguna.

Salah satu contoh penerapan fungsi sanitasi data adalah dalam konteks pengembangan web dengan bahasa pemrograman PHP. Fungsi seperti `htmlspecialchars()` dan `strip_tags()` sering digunakan untuk mencegah penyerang dari menyisipkan kode HTML atau JavaScript yang berpotensi berbahaya melalui input pengguna. Fungsi `htmlspecialchars()` mengubah karakter khusus menjadi entitas HTML, yang mencegah mereka dieksekusi sebagai kode oleh browser. Sementara itu, fungsi `strip_tags()` menghapus semua tag HTML dan PHP dari string, yang juga mencegah eksekusi kode berbahaya.

Sebagai contoh, berikut adalah cara dasar untuk menyaring input pengguna dan fungsi sanitasi data dalam bahasa pemrograman PHP untuk mencegah serangan XSS.

```
function bersihkan_input($data) {  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
}
```

```

return $data;
}

// Menggunakan fungsi di atas pada input pengguna
$input_pengguna = "<script>alert('XSS');</script>";
$input_pengguna_bersih =
bersihkan_input($input_pengguna);

echo      $input_pengguna_bersih;           //      Output:
&lt;script&gt;alert('XSS');&lt;/script&gt;

```

Pada contoh di atas, fungsi `bersihkan_input` melakukan beberapa operasi untuk membersihkan data yang diberikan

1. Trim menghilangkan spasi ekstra di awal dan akhir string.
2. Stripslashes menghapus 'slashes' () dari string.
3. Htmlspecialchars mengubah karakter khusus seperti '<', '>', dan '&' menjadi entitas HTML, yang mencegah mereka dieksekusi oleh browser sebagai kode HTML atau JavaScript.

Harap dicatat bahwa contoh ini sangat dasar dan mungkin tidak mencukupi untuk semua kasus penggunaan. Selalu penting untuk mempertimbangkan konteks di mana input pengguna akan digunakan dan memastikan bahwa semua input pengguna disaring dan divalidasi dengan tepat.

11.3.3 Menggunakan Header HTTP Yang Tepat (*Content Security Policy*)

X *Content Security Policy* (CSP) merupakan sebuah alat keamanan yang penting untuk mengurangi risiko serangan *Cross-Site Scripting* (XSS). CSP berfungsi dengan memberikan instruksi kepada browser mengenai sumber mana saja yang

diizinkan untuk memuat konten, seperti skrip, gambar, dan *stylesheet*, melalui sebuah header HTTP yang disebut *Content-Security-Policy*.

CSP memungkinkan pengembang web untuk membatasi asal skrip dan lainnya, mencegah eksekusi skrip inline dan mengevaluasi skrip yang tidak aman. Misalnya, jika pengembang menentukan bahwa hanya skrip dari domain mereka sendiri yang boleh dieksekusi, maka skrip yang disuntikkan oleh penyerang (serangan XSS) akan dicegah oleh browser untuk dieksekusi. Untuk mendefinisikan kebijakan ini, pengembang dapat menambahkan header *Content-Security-Policy* ke respon HTTP mereka dengan nilai yang menentukan sumber mana yang diizinkan.

Contoh penggunaan CSP adalah sebagai berikut: *Content-Security-Policy: script-src 'self'* akan menginstruksikan browser untuk hanya memuat skrip yang berasal dari domain yang sama dengan halaman tersebut. CSP lebih canggih daripada hanya mencegah serangan XSS, karena dapat membatasi berbagai jenis konten, membuatnya menjadi alat yang kuat untuk mengendalikan apa yang bisa dan tidak bisa dilakukan oleh halaman web.

Meskipun penggunaan CSP sangat membantu dalam mencegah serangan XSS, penting untuk diingat bahwa kebijakan yang tidak dikonfigurasi dengan benar bisa jadi tidak efektif. Selain itu, CSP harus digunakan sebagai bagian dari pendekatan pertahanan berlapis, bukan sebagai solusi keamanan tunggal.

Berikut contoh penerapan *Content Security Policy* (CSP) bisa dilihat pada kode berikut, yang menggunakan kerangka kerja Express.js di Node.js

```
const express = require('express');  
const helmet = require('helmet');
```

```
const app = express();

app.use(helmet.contentSecurityPolicy({
  directives: {
    defaultSrc: ["'self'"],
    scriptSrc: ["'self'", "'unsafe-inline'"],
    imgSrc: ["'self'", "https://*.example.com"],
    styleSrc: ["'self'"]
  }
}));

app.get('/', (req, res) => {
  res.send('Hello, world!');
});

app.listen(3000, () => console.log('Server berjalan di port 3000'));
```

Dalam contoh ini, kita menggunakan modul Helmet, yang merupakan middleware untuk Express.js yang menetapkan berbagai header HTTP untuk membantu melindungi aplikasi Anda. Dalam hal ini, kita menggunakan fungsi `contentSecurityPolicy` dari Helmet untuk menetapkan header `Content-Security-Policy`.

Kebijakan ini mendefinisikan beberapa direktif:

1. `defaultSrc: ["'self'"]`: Direktif ini berarti bahwa semua konten (kecuali untuk jenis konten yang mendefinisikan direktif mereka sendiri seperti `scriptSrc` atau `imgSrc`) hanya dapat dimuat dari domain yang sama dengan aplikasi web.

2. `scriptSrc: ["'self'", "'unsafe-inline'"]`: Direktif ini berarti bahwa skrip hanya dapat dimuat dari domain yang sama dengan aplikasi web, atau sebagai skrip inline. Namun, perlu dicatat bahwa penggunaan `'unsafe-inline'` dapat melemahkan kebijakan dan sebaiknya dihindari jika memungkinkan.
3. `imgSrc: ["'self'", "https://*.example.com"]`: Direktif ini berarti bahwa gambar hanya dapat dimuat dari domain yang sama dengan aplikasi web, atau dari subdomain apa pun dari `example.com`.
4. `styleSrc: ["'self'"]`: Direktif ini berarti bahwa stylesheet hanya dapat dimuat dari domain yang sama dengan aplikasi web.

Penerapan CSP ini akan mencegah pemuatan konten yang tidak diinginkan dan berpotensi berbahaya, yang membantu mencegah serangan seperti XSS. Namun, penting untuk mencatat bahwa konfigurasi CSP harus disesuaikan sesuai dengan kebutuhan aplikasi Anda dan harus selalu diuji secara menyeluruh untuk memastikan bahwa kebijakan yang diterapkan efektif dan tidak mengganggu fungsi normal dari aplikasi Anda.

11.3.4 Prinsip-Prinsip Pengkodean Yang Aman

Pengkodean yang aman adalah fondasi dari keamanan web. Ini adalah serangkaian praktik terbaik yang digunakan oleh pengembang untuk menulis kode yang tahan terhadap serangan. Prinsip pertama dan yang paling penting adalah `'selalu meragukan input pengguna'`. Sebagai aturan umum, sebaiknya anggap semua input pengguna sebagai potensi ancaman dan selalu lakukan validasi dan sanitasi sebelum diproses atau ditampilkan.

Prinsip kedua adalah `'menggunakan pertahanan berlapis'`. Jangan mengandalkan pada satu jenis pertahanan

saja. Misalnya, meskipun sanitasi input pengguna penting, Anda juga harus menggunakan teknologi keamanan lainnya seperti HTTPS dan CSP (*Content Security Policy*). Menerapkan berbagai jenis pertahanan dapat menangkap ancaman yang mungkin lolos dari satu jenis pertahanan.

Prinsip ketiga adalah 'minimalisasi hak istimewa'. Proses dan pengguna harus diberikan hak istimewa minimum yang dibutuhkan untuk menyelesaikan tugas mereka. Jika suatu bagian dari sistem diberi akses berlebihan, maka bagian tersebut menjadi target yang menarik bagi penyerang. Minimalisasi hak istimewa dapat membatasi dampak serangan jika suatu bagian dari sistem Anda dikompromikan.

Untuk mengembangkan perangkat lunak yang aman, penting untuk mengintegrasikan prinsip-prinsip ini ke dalam setiap tahap siklus hidup pengembangan perangkat lunak, mulai dari perencanaan dan desain hingga pengujian dan pemeliharaan. Pemahaman yang baik tentang prinsip-prinsip pengkodean yang aman ini akan membantu pengembang menulis kode yang lebih aman dan membangun aplikasi web yang lebih tahan terhadap serangan.

11.4 Melindungi Aplikasi dari Serangan CSRF

11.4.1 Penggunaan Token CSRF

Token CSRF adalah salah satu cara paling efektif untuk mencegah serangan Cross-Site Request Forgery (CSRF). Token ini adalah nilai unik dan acak yang disertakan dalam setiap permintaan yang berpotensi mengubah state (misalnya, permintaan POST) dari aplikasi web Anda. Tujuan dari token ini adalah untuk memastikan bahwa permintaan yang berpotensi mengubah state berasal dari situs yang sah dan bukan dari situs lain (serangan CSRF).

Prinsip kerja dari token CSRF cukup sederhana. Saat pengguna masuk ke aplikasi web Anda, sistem akan

menghasilkan token CSRF dan mengaitkannya dengan sesi pengguna tersebut. Kemudian, token ini disertakan dalam setiap form yang dikirim ke pengguna. Saat pengguna mengirimkan form tersebut, token CSRF akan dikirim kembali ke server dan server akan memvalidasinya. Jika token tidak cocok dengan token yang diharapkan, maka permintaan tersebut ditolak.

Penerapan token CSRF yang efektif harus mempertimbangkan beberapa hal. Pertama, token harus cukup acak dan sulit ditebak, sehingga penyerang tidak bisa menebak nilai token. Kedua, token harus diikat dengan sesi pengguna, sehingga meskipun penyerang berhasil mencuri token, mereka tidak dapat menggunakannya dalam konteks sesi lain. Terakhir, token harus diperiksa pada setiap permintaan yang berpotensi mengubah state, dan permintaan tersebut harus ditolak jika token tidak valid atau tidak ada.

Berikut adalah contoh sederhana tentang bagaimana Anda bisa mengimplementasikan token CSRF dalam aplikasi web menggunakan Express.js dan csurf, sebuah middleware Node.js untuk melindungi dari serangan CSRF.

```
const express = require('express');
const cookieParser = require('cookie-parser');
const csrf = require('csurf');

// inisiasi express dan csrf protection
const app = express();
const csrfProtection = csrf({ cookie: true });

app.use(cookieParser());

// form yang dilindungi dengan csrf
```

```
app.get('/form', csrfProtection, (req, res) => {  
  // token csrf disertakan di dalam form  
  res.send(<form action="/process" method="POST">  
    <input type="hidden" name="_csrf"  
value="${req.csrfToken()}">  
    <input type="text" name="inputUser">  
    <input type="submit" value="Submit">  
  </form>`);  
});  
// memproses form yang dilindungi dengan csrf  
app.post('/process', csrfProtection, (req, res) => {  
  res.send('Data received.');});  
app.listen(3000, () => console.log('Server is running on port  
3000'));
```

Dalam contoh di atas, kita menggunakan middleware `csrfProtection` yang dibuat oleh `csrf` pada setiap endpoint yang membutuhkan perlindungan CSRF. Ketika permintaan GET dibuat ke `/form`, middleware ini akan menghasilkan token CSRF dan menambahkannya ke `req.csrfToken`. Kemudian kita menyertakan token ini dalam form yang kita kirim ke pengguna sebagai bagian dari respon.

Ketika form ini dikirimkan oleh pengguna (dengan permintaan POST ke `/process`), middleware `csrfProtection` akan memeriksa apakah token CSRF dalam permintaan cocok dengan token CSRF yang diharapkan. Jika tidak, permintaan tersebut akan ditolak dan error akan dikembalikan. Jika token cocok, maka permintaan tersebut akan diproses seperti biasa.

11.4.2 Menggunakan Header HTTP yang tepat (Referer, Origin)

Dalam konteks serangan CSRF, penggunaan header HTTP yang tepat dapat memberikan lapisan perlindungan tambahan. Khususnya, dua header yang sering digunakan dalam mitigasi CSRF adalah Referer dan Origin.

Header Referer digunakan untuk menunjukkan URL asal suatu permintaan HTTP. Jika permintaan berasal dari situs yang sah, maka header Referer akan sesuai dengan situs tersebut. Namun, dalam kasus serangan CSRF, header Referer akan menunjukkan URL situs penyerang. Oleh karena itu, memeriksa header Referer bisa menjadi metode efektif untuk mencegah serangan CSRF. Namun, metode ini memiliki beberapa kelemahan. Misalnya, beberapa proxy dan firewall akan menghapus header Referer, dan beberapa pengguna mungkin memilih untuk menonaktifkan pengiriman header Referer.

Header Origin, yang diperkenalkan dalam spesifikasi CORS (*Cross-Origin Resource Sharing*), juga bisa digunakan untuk mitigasi CSRF. Mirip dengan header Referer, header Origin menunjukkan URL asal permintaan HTTP. Namun, berbeda dengan header Referer, header Origin hanya akan dikirim oleh browser dalam kasus permintaan cross-origin. Oleh karena itu, jika aplikasi web Anda hanya menerima permintaan dari origin yang sama, Anda bisa memeriksa header Origin dan menolak permintaan yang tidak memiliki header Origin atau memiliki header Origin yang tidak cocok.

Menggunakan header Referer dan Origin dapat memberikan perlindungan yang baik terhadap serangan CSRF, namun metode ini tidak bisa digunakan sebagai satu-satunya lapisan pertahanan. Serangan CSRF yang lebih canggih mungkin bisa mengatasi metode ini. Oleh karena itu, sangat disarankan untuk menggunakan metode ini dalam kombinasi dengan metode perlindungan CSRF lainnya, seperti token CSRF.

11.4.3 Verifikasi Permintaan Sisi Server

Dalam konteks keamanan web, verifikasi permintaan sisi server adalah proses dimana server memeriksa dan memvalidasi permintaan yang diterima sebelum melanjutkan untuk memproses permintaan tersebut. Proses ini sangat penting dalam mencegah serangan berbasis web seperti *Cross-Site Request Forgery* (CSRF).

Dalam kasus serangan CSRF, penyerang mencoba memanipulasi korban untuk mengirimkan permintaan yang tidak sah ke server dengan cara menyamar sebagai korban. Oleh karena itu, salah satu cara terbaik untuk melindungi aplikasi web dari serangan seperti ini adalah dengan memastikan bahwa setiap permintaan yang berpotensi merusak yang diterima oleh server telah diverifikasi dengan benar. Ada beberapa metode yang bisa digunakan untuk verifikasi permintaan ini, termasuk penggunaan token CSRF, pengecekan header Referer atau Origin, dan pengecekan autentikasi berbasis sesi.

Token CSRF adalah nilai acak yang dihasilkan oleh server dan disertakan dalam setiap form yang dikirim ke pengguna. Saat form dikirimkan kembali ke server, token ini juga dikirim kembali dan diverifikasi oleh server. Jika token tidak cocok atau tidak ada, maka permintaan tersebut ditolak. Pengecekan header Referer dan Origin bekerja dengan cara memastikan bahwa permintaan berasal dari situs yang sah dan bukan dari situs penyerang. Sementara pengecekan autentikasi berbasis sesi bekerja dengan cara memastikan bahwa permintaan tersebut berasal dari sesi pengguna yang telah terautentikasi.

Verifikasi permintaan sisi server adalah bagian penting dari pertahanan yang kuat terhadap serangan CSRF. Namun, penting untuk diingat bahwa metode ini sebaiknya digunakan dalam kombinasi dengan metode perlindungan lainnya untuk memastikan keamanan yang maksimal.

11.5 Keamanan Basis Data dan Serangan Injection

11.5.1 Menggunakan Prepared Statements atau ORM

Dalam konteks keamanan basis data, penggunaan prepared statements atau Object-Relational Mapping (ORM) adalah dua strategi yang efektif dalam melindungi aplikasi web Anda dari serangan SQL Injection.

Prepared statements, juga dikenal sebagai parameterized queries, adalah fitur yang disediakan oleh banyak library basis data. Prepared statements memisahkan antara instruksi SQL dan data yang akan dimasukkan ke dalam instruksi tersebut. Ini berarti bahwa data pengguna tidak pernah diinterpretasikan sebagai instruksi SQL, yang menghilangkan kemungkinan serangan SQL injection. Sebagai tambahan, prepared statements juga memiliki beberapa keuntungan lain, seperti peningkatan performa untuk query yang dijalankan berulang kali.

ORM adalah teknik pemrograman yang memungkinkan aplikasi Anda untuk berinteraksi dengan basis data menggunakan objek dan metode, bukan instruksi SQL langsung. ORM biasanya memiliki fitur bawaan untuk melindungi dari serangan SQL Injection, termasuk penggunaan prepared statements atau metode input sanitasi lainnya. Sebagai tambahan, ORM dapat membantu meningkatkan produktivitas dan maintainability dengan memungkinkan Anda untuk bekerja pada tingkat abstraksi yang lebih tinggi dan menghindari sebagian besar SQL biasa.

Meskipun penggunaan prepared statements dan ORM dapat memberikan perlindungan yang sangat baik terhadap serangan SQL Injection, penting untuk diingat bahwa mereka tidak menawarkan solusi lengkap. Misalnya, mereka mungkin tidak melindungi dari serangan yang mengincar bagian lain dari aplikasi Anda, atau dari serangan SQL Injection yang melibatkan bagian instruksi SQL yang tidak bisa

diparameterisasi. Oleh karena itu, mereka sebaiknya digunakan dalam kombinasi dengan metode pertahanan lainnya, seperti validasi dan sanitasi input, dan penggunaan prinsip-prinsip pengkodean yang aman.

11.5.2 Validasi Data Pengguna

Validasi data pengguna adalah prinsip penting dalam keamanan aplikasi web dan basis data. Konsep ini merujuk pada praktek memeriksa data masukan pengguna sebelum diproses oleh aplikasi atau disimpan dalam basis data. Tujuannya adalah untuk mencegah masuknya data yang tidak sah atau berbahaya, yang dapat merusak sistem atau menjadi pintu masuk untuk serangan, seperti serangan *SQL Injection* atau *Cross-Site Scripting (XSS)*.

Metode validasi data pengguna dapat bervariasi tergantung pada konteks dan tipe data. Misalnya, jika aplikasi Anda menerima alamat email dari pengguna, Anda mungkin ingin memeriksa apakah data yang dimasukkan sesuai dengan format alamat email yang sah. Ini bisa dilakukan dengan menggunakan regular expressions atau fungsi validasi email yang tersedia dalam banyak library pemrograman. Untuk data lain seperti nomor telepon, tanggal, atau nilai numerik, metode validasi yang sesuai juga harus diterapkan.

11.5.3 Mengenkripsi Data Sensitif

Pengamanan informasi sensitif merupakan aspek penting dalam pengembangan aplikasi web. Salah satu cara yang efektif untuk melindungi data ini adalah dengan enkripsi. Enkripsi adalah proses mengubah informasi atau data menjadi kode yang sulit dipahami atau diakses tanpa kunci enkripsi yang tepat. Teknik ini dapat digunakan untuk melindungi data yang disimpan (*at rest*) maupun data yang ditransmisikan (in transit).

Untuk data yang disimpan, seperti password atau informasi pribadi pengguna dalam basis data, biasanya digunakan teknik enkripsi satu arah atau hashing. Hashing mengubah data menjadi rangkaian karakter yang tetap dan unik, tanpa cara yang praktis untuk mengembalikannya ke bentuk aslinya. Teknik ini sering digunakan untuk menyimpan password dengan aman, karena meski basis data disusupi, penyerang tidak akan dapat memahami password asli. Teknik ini biasanya dilengkapi dengan "salt", yaitu data acak yang ditambahkan ke data asli sebelum di-hash untuk mencegah serangan kamus atau rainbow table.

Sementara itu, untuk melindungi data yang ditransmisikan melalui jaringan, seperti saat pengguna mengirimkan informasi pribadi ke server, teknik enkripsi dua arah atau symmetric encryption biasanya digunakan. Dalam teknik ini, data dienkripsi sebelum ditransmisikan dan kemudian didekripsi oleh penerima menggunakan kunci yang sama. HTTPS adalah contoh implementasi ini, di mana semua komunikasi antara browser pengguna dan server web dienkripsi untuk melindungi terhadap serangan man-in-the-middle atau penyadapan.

11.5.4 Konfigurasi Yang Aman Untuk Basis Data

Konfigurasi basis data yang aman sangat penting untuk menjaga integritas dan keamanan informasi yang tersimpan dalam basis data Anda. Hal pertama yang perlu diperhatikan adalah pemberian hak akses. Prinsip pemberian hak minimal atau principle of least privilege adalah cara terbaik untuk mengatur hak akses. Ini berarti pengguna atau aplikasi hanya diberikan hak akses yang benar-benar diperlukan untuk menjalankan tugasnya, dan tidak lebih. Sebagai contoh, jika aplikasi web Anda hanya perlu membaca data dari basis data, maka tidak ada alasan untuk memberikan hak untuk memodifikasi data.

Selanjutnya, memastikan bahwa basis data dan perangkat lunak yang digunakan selalu diperbarui adalah langkah penting lainnya. Pembaharuan seringkali mencakup perbaikan keamanan yang penting untuk melindungi basis data Anda dari kerentanan yang telah dikenali dan dapat dieksploitasi oleh penyerang. Karena itu, penting untuk menjaga agar sistem Anda selalu up-to-date.

Memastikan bahwa basis data Anda berada di balik firewall dan tidak dapat diakses secara langsung dari internet adalah langkah penting lainnya. Mengizinkan akses langsung ke basis data Anda dari internet membuka potensi untuk serangan yang dapat mengkompromikan data Anda. Dalam banyak kasus, ini dapat dicegah dengan membatasi akses jaringan ke basis data Anda dan memastikan bahwa hanya aplikasi atau layanan yang tepercaya yang dapat berinteraksi dengan basis data.

11.6 Keamanan Sesi dan Otentikasi

11.6.1 Penggunaan Token Sesi Yang Aman

Penggunaan token sesi yang aman merupakan bagian penting dalam mengelola otentikasi pengguna dan keamanan sesi. Token sesi adalah string unik yang dibuat dan dikelola oleh server untuk mengidentifikasi dan mengautentikasi pengguna selama interaksi mereka dengan aplikasi web. Token ini biasanya dibuat saat pengguna pertama kali masuk dan digunakan selama durasi sesi pengguna tersebut.

Token sesi harus dihasilkan dengan cara yang acak dan cukup kompleks sehingga tidak dapat ditebak atau diprediksi oleh penyerang. Ini seringkali dicapai dengan menggunakan algoritme generasi nomor acak yang kuat. Selain itu, token sesi haruslah unik untuk setiap sesi dan pengguna, serta harus dihapus atau dibuat tidak valid saat pengguna logout atau setelah periode waktu tertentu. Hal ini penting untuk

mencegah serangan replay dan serangan session hijacking, di mana penyerang mencoba menggunakan token sesi lama atau dicuri untuk mendapatkan akses.

Penggunaan protokol yang aman seperti HTTPS juga penting untuk melindungi token sesi saat transmisi antara klien dan server. Tanpa enkripsi, penyerang dapat melakukan serangan man-in-the-middle dan mencuri token sesi saat ditransmisikan melalui jaringan. Penggunaan HTTPS membantu mencegah hal ini dengan enkripsi data sebelum transmisi.

11.6.2 Melakukan Otentikasi Yang Kuat

Melakukan otentikasi yang kuat adalah prasyarat dalam pengamanan aplikasi web. Otentikasi yang kuat membantu memastikan bahwa pengguna adalah siapa yang mereka klaim, dan bukan orang lain yang mencoba mengakses akun mereka. Salah satu pendekatan dalam mencapai ini adalah dengan menggunakan metode otentikasi multi-faktor (MFA). MFA menggabungkan dua atau lebih metode otentikasi yang independen: sesuatu yang pengguna tahu (misalnya, password), sesuatu yang pengguna miliki (misalnya, token atau perangkat keras), dan sesuatu yang merupakan pengguna itu sendiri (misalnya, biometrik).

Selanjutnya, praktik yang baik lainnya dalam otentikasi yang kuat adalah penggunaan password yang kuat. Password harus cukup panjang dan kompleks untuk mencegah serangan brute force atau serangan kamus. Selain itu, password tidak boleh disimpan dalam bentuk teks biasa, tetapi harus di-hash dan di-salt untuk memberikan lapisan keamanan ekstra dalam hal basis data disusupi.

Terakhir, pembatasan jumlah upaya login yang gagal juga dapat membantu mencegah serangan brute force. Setelah jumlah upaya tertentu, akun dapat dikunci untuk periode waktu tertentu, atau sampai pengguna berhasil memverifikasi

identitasnya melalui cara lain. Ini membantu mencegah penyerang mencoba berbagai kombinasi password hingga berhasil.

11.6.3 Menggunakan Metode Hashing Yang Aman

Menggunakan metode hashing yang aman merupakan prinsip penting dalam mengelola keamanan sesi dan otentikasi. Hashing adalah proses mengubah data menjadi nilai hash yang unik dan tidak dapat dikembalikan ke bentuk aslinya. Dalam konteks otentikasi pengguna, metode hashing digunakan untuk menyimpan password dengan aman.

Penting untuk menggunakan algoritme hashing yang kuat dan tahan terhadap serangan brute force dan serangan rainbow table. Algoritme seperti bcrypt, Argon2, atau scrypt direkomendasikan karena kekuatan kriptografis dan perlawanan terhadap serangan yang cenderung menggunakan komputasi yang intensif. Selain itu, metode hashing yang baik juga akan menggabungkan teknik salt, yang adalah data acak yang ditambahkan ke input sebelum di-hash. Salt membantu melindungi dari serangan serupa, di mana penyerang menggunakan tabel hash pra-dihasilkan untuk mencocokkan nilai hash dengan kata sandi yang mungkin.

Selain itu, sangat penting untuk menghindari penggunaan algoritme hashing yang sudah tidak aman atau rentan terhadap serangan. Algoritme hashing seperti MD5 atau SHA-1 tidak lagi direkomendasikan karena rentan terhadap serangan kolisi atau serangan penambangan hash yang canggih. Penggunaan algoritme hashing yang aman dan terbaru penting untuk menjaga integritas otentikasi pengguna dan melindungi password dari ekstraksi yang tidak sah.

Menerapkan metode hashing yang aman adalah bagian penting dalam strategi keamanan sesi dan otentikasi. Dalam kombinasi dengan praktik keamanan lainnya, seperti penggunaan salt, penggunaan algoritme hashing yang kuat dan

terpercaya akan membantu melindungi password pengguna dengan efektif.

11.6.4 Manajemen Token Akses

Manajemen token akses adalah aspek penting dalam mengelola keamanan sesi dan otentikasi dalam aplikasi web. Token akses adalah tanda atau kredensial yang digunakan untuk mengidentifikasi dan mengautentikasi pengguna saat mereka berinteraksi dengan sistem. Pengelolaan token akses yang baik melibatkan proses pembuatan, penyimpanan, validasi, pembaruan, dan pencabutan token secara aman.

Pertama, penting untuk menggunakan metode pembuatan token akses yang kuat. Token akses harus dihasilkan dengan menggunakan algoritme yang aman dan acak sehingga sulit untuk ditebak atau diprediksi oleh penyerang. Menggunakan metode yang kuat dan acak dalam pembuatan token akses membantu mencegah serangan seperti serangan pencarian token (*token guessing*) atau serangan brute force.

Selanjutnya, token akses harus disimpan dengan aman untuk mencegah akses yang tidak sah. Idealnya, token akses tidak boleh disimpan dalam bentuk teks biasa atau dalam cookie yang dapat diakses oleh penyerang. Sebagai gantinya, token akses harus dienkripsi dan disimpan dalam media yang aman seperti session storage atau local storage di sisi klien, atau dalam basis data terenkripsi di sisi server.

Terakhir, proses validasi, pembaruan, dan pencabutan token akses harus diterapkan dengan hati-hati. Saat menerima token akses dari pengguna, aplikasi harus memvalidasi token untuk memastikan keasliannya dan bahwa pengguna terautentikasi dengan benar. Selain itu, jika ada kebutuhan untuk memperbarui atau mencabut token akses, proses ini harus dilakukan secara aman dan sesuai dengan kebijakan yang telah ditetapkan. Pencabutan token akses yang tidak tepat

dapat membuka celah bagi penyerang untuk tetap mengakses sumber daya atau melanjutkan sesi yang tidak sah.

Melalui manajemen yang baik, token akses dapat menjadi alat yang kuat dalam mengelola otentikasi pengguna dan keamanan sesi dalam aplikasi web.

11.7 Praktek Pengembangan Aplikasi yang Aman

11.7.1 Menggunakan Framework Yang Aman

Berbagai kerangka kerja web atau framework telah dikembangkan sepanjang waktu untuk membantu dalam mempermudah pengembangan aplikasi web, termasuk aspek keamanannya. Sebagaimana ditunjukkan oleh (Reaves *et al.*, 2017), penerapan kerangka kerja yang tepat dapat secara signifikan membantu dalam mengurangi kesalahan pemrograman yang dapat menimbulkan kerentanan keamanan. Kerangka kerja seperti Django untuk Python atau Laravel untuk PHP, misalnya, telah memasukkan berbagai fitur keamanan langsung ke dalam paket mereka, seperti perlindungan terhadap serangan Cross Site Scripting (XSS) dan SQL Injection. Selain itu, mereka juga sering diperbarui untuk mengatasi kerentanan keamanan baru yang muncul seiring waktu, memberikan tingkat keamanan tambahan bagi aplikasi yang dibangun di atasnya.

Salah satu prinsip utama dalam pemrograman yang aman adalah "Jangan mengulangi roda", atau dengan kata lain, jangan mencoba membuat sendiri fitur yang sudah ada dan telah diuji secara luas. Seperti yang dicatat oleh (Green and Smith, 2016), banyak pengembang meremehkan kompleksitas dari implementasi fitur keamanan yang efektif, seperti hashing password atau sanitasi input. Dengan menggunakan kerangka kerja yang telah mencakup fitur-fitur tersebut, pengembang dapat fokus pada fungsi utama aplikasi mereka sambil memastikan bahwa mereka tetap aman.

Namun, penting juga untuk diingat bahwa penggunaan kerangka kerja yang aman tidak menjamin aplikasi akan sepenuhnya aman. Sebagai tambahan, pengembang harus tetap berupaya untuk memahami prinsip-prinsip keamanan dasar dan terus memperbarui pengetahuan mereka tentang ancaman keamanan terbaru. Setiap kali versi baru dari kerangka kerja dikeluarkan, penting bagi pengembang untuk memperbarui aplikasi mereka dengan versi terbaru tersebut untuk memastikan keamanannya.

11.7.2 Menerapkan Pembaruan Keamanan Secara Teratur

Menjaga aplikasi web Anda selalu up-to-date adalah salah satu cara paling efektif untuk melindunginya dari ancaman keamanan. Dalam penelitian mereka, (Shahriar and Zulkernine, 2012) menemukan bahwa sebagian besar kerentanan keamanan muncul dari software yang tidak diperbarui. Para pengembang perangkat lunak dan kerangka kerja biasanya merilis pembaruan secara berkala untuk memperbaiki bug dan kerentanan yang ditemukan. Oleh karena itu, menerapkan pembaruan ini secara teratur sangat penting untuk mempertahankan keamanan aplikasi web Anda.

Tidak hanya perangkat lunak dan kerangka kerja yang digunakan, sistem operasi dan perangkat keras server juga harus diperbarui secara teratur. Sebagai contoh, dalam laporan tahunannya, Symantec mencatat bahwa banyak serangan berhasil karena sistem operasi dan perangkat keras server tidak diperbarui. Selain itu, Symantec juga menemukan bahwa serangan semacam itu dapat dihindari dengan menerapkan pembaruan keamanan secara teratur.

Menerapkan pembaruan keamanan tidak selalu mudah dan dapat menimbulkan tantangan tersendiri. Sebagai contoh, pembaruan yang tidak kompatibel dengan perangkat lunak atau kerangka kerja yang ada dapat menyebabkan aplikasi berhenti bekerja. Oleh karena itu, penting untuk selalu

mencoba pembaruan pada lingkungan yang terisolasi sebelum menerapkannya pada sistem produksi. Selain itu, memantau berbagai sumber informasi keamanan seperti CERT (*Computer Emergency Response Teams*) atau vendor perangkat lunak Anda juga akan membantu Anda tetap up-to-date tentang pembaruan terbaru.

11.7.3 Memeriksa Kerentanan Terkait Keamanan

Pemeriksaan kerentanan terkait keamanan adalah bagian penting dari proses pengembangan dan pemeliharaan aplikasi web. Ini melibatkan identifikasi dan penanganan potensi kerentanan dalam kode aplikasi, yang bisa diserang oleh penjahat siber. Pengujian kerentanan adalah salah satu pendekatan yang paling efektif untuk mendeteksi dan mengurangi risiko ancaman keamanan dalam aplikasi.

Ada berbagai metode dan alat yang dapat digunakan untuk memeriksa kerentanan, termasuk pemindaian statis dan dinamis, dan pengujian penetrasi. Pemindaian statis melibatkan pengecekan kode aplikasi untuk menemukan kerentanan potensial tanpa harus menjalankan program, sementara pemindaian dinamis melibatkan pengujian aplikasi saat berjalan. Pengujian penetrasi adalah teknik yang lebih lanjut, melibatkan simulasi serangan pada aplikasi untuk menemukan titik lemah.

Namun, hanya melakukan pemeriksaan kerentanan bukanlah solusi lengkap. Hal yang sama pentingnya adalah memahami hasil dari pemeriksaan tersebut dan melakukan tindakan yang diperlukan untuk memperbaiki kerentanan yang ditemukan. Tindakan ini mungkin mencakup memperbaiki bug dalam kode, memperbarui perangkat lunak atau sistem operasi, atau mengubah konfigurasi sistem untuk lebih mengamankan aplikasi.

11.7.4 Mengelola Akses Dan Izin Dengan Bijak

Mengelola akses dan izin dengan bijak adalah elemen kunci dalam menjaga keamanan aplikasi web. Prinsip dasarnya, dikenal sebagai prinsip hak istimewa minimal atau *Principle of Least Privilege* (PoLP), menegaskan bahwa setiap pengguna, program, atau proses sistem harus diberikan hak istimewa minimal yang diperlukan untuk melaksanakan tugasnya. Menurut beberapa studi, penerapan PoLP dapat mengurangi risiko kerentanan keamanan dengan membatasi akses yang tidak perlu kepada sistem atau data.

Mengelola akses dan izin melibatkan sejumlah tindakan, seperti otentikasi pengguna, penentuan hak akses, dan pengauditan akses. Otentikasi pengguna bertujuan untuk memverifikasi identitas pengguna, sementara penentuan hak akses menentukan apa yang dapat diakses dan dilakukan oleh pengguna setelah otentikasi berhasil. Pengauditan akses, di sisi lain, melibatkan pelacakan dan mencatat aktivitas pengguna untuk mendeteksi dan mencegah perilaku yang mencurigakan.

Mengelola akses dan izin juga dapat menjadi tantangan. Dalam prakteknya, organisasi seringkali memberikan lebih banyak hak istimewa kepada pengguna atau proses daripada yang sebenarnya dibutuhkan. Misalnya, pengguna yang diberikan akses admin yang tidak perlu, atau proses sistem yang berjalan dengan hak istimewa tinggi. Oleh karena itu, perlu ada upaya yang berkelanjutan untuk meninjau dan menyesuaikan hak akses dan izin sesuai dengan kebutuhan nyata.

11.8 Melakukan Pengujian Keamanan

11.8.1 Metodologi Pengujian Keamanan

Metodologi pengujian keamanan merupakan kerangka kerja yang digunakan oleh tim pengujian untuk mengidentifikasi dan mengevaluasi potensi kerentanan dalam

sistem atau aplikasi. Salah satu metodologi yang paling dikenal dan diterima secara luas adalah metodologi Open Web Application Security Project (OWASP) yang merujuk pada berbagai proses, teknik, dan alat yang digunakan dalam pengujian keamanan aplikasi web.

Metodologi OWASP memandu pengguna melalui berbagai tahapan pengujian keamanan, mulai dari pengumpulan informasi, penentuan lingkup pengujian, identifikasi dan analisis kerentanan, hingga laporan penemuan. Pengujian keamanan sebaiknya tidak hanya dilakukan sekali, tetapi sebaiknya menjadi bagian dari siklus hidup pengembangan perangkat lunak yang berkelanjutan. Selain itu, OWASP menyarankan penggunaan kombinasi pendekatan "*black box*" dan "*white box*" dalam pengujian, yang melibatkan pengujian baik tanpa pengetahuan tentang bagaimana sistem dibangun dan juga dengan pengetahuan lengkap tentang kode dan struktur sistem.

"*Black box*" dan "*white box*" adalah istilah yang digunakan dalam pengujian perangkat lunak, termasuk dalam konteks pengujian keamanan.

Pengujian "*black box*" adalah pendekatan pengujian di mana sistem dianalisis tanpa informasi tentang bagaimana sistem tersebut bekerja di dalam. Pengujian ini difokuskan pada input dan output dari sistem, tanpa mempertimbangkan detail internal seperti struktur kode, implementasi, dan prosedur internal. Tujuannya adalah untuk mengevaluasi fungsionalitas dan perilaku sistem dari perspektif pengguna, tanpa mempertimbangkan bagaimana fungsionalitas dan perilaku tersebut dicapai. Dalam konteks pengujian keamanan, pengujian "*black box*" dapat digunakan untuk mendeteksi kerentanan yang dapat dieksploitasi oleh penyerang yang tidak memiliki pengetahuan tentang sistem.

Sementara itu, pengujian "*white box*" adalah pendekatan pengujian di mana sistem dianalisis dengan pengetahuan

lengkap tentang bagaimana sistem tersebut bekerja di dalam. Pengujian ini melibatkan pemeriksaan struktur kode, algoritma, dan struktur data internal. Tujuannya adalah untuk memeriksa kualitas kode dan menemukan kesalahan atau kerentanan yang mungkin ada di dalam kode. Dalam konteks pengujian keamanan, pengujian "white box" dapat digunakan untuk menemukan kerentanan yang mungkin tidak terdeteksi saat pengujian "black box", seperti kerentanan yang hanya dapat dieksploitasi jika penyerang memiliki pengetahuan internal tentang sistem.

11.8.2 Tools dan Teknik Pengujian Yang Umum

Dalam melakukan pengujian keamanan, ada banyak alat dan teknik yang dapat digunakan oleh pengembang dan penester. Alat ini dapat membantu mengotomatisasi sejumlah besar tugas pengujian dan memungkinkan tim untuk lebih efisien dalam menemukan dan memperbaiki kerentanan. Salah satu alat yang paling populer dan sering digunakan adalah OWASP ZAP (*Zed Attack Proxy*). ZAP adalah alat pengujian penetrasi yang gratis dan open-source untuk menemukan kerentanan keamanan dalam aplikasi web. ZAP menyediakan serangkaian fitur yang komprehensif untuk pengujian penetrasi, termasuk pemindaian otomatis, pencegahan proxy, dan fuzzer.

Selain ZAP, alat lain yang populer di kalangan penester adalah Burp Suite. Burp Suite adalah alat pengujian keamanan web yang menyediakan serangkaian fitur untuk pengujian penetrasi, termasuk pencegahan proxy, pemindaian kerentanan otomatis, dan analisis lalu lintas HTTP(S). Alat ini memungkinkan pengguna untuk melihat dan memanipulasi lalu lintas antara browser dan target aplikasi, memungkinkan penester untuk menemukan dan mengeksploitasi kerentanan yang mungkin ada.

Selain menggunakan alat, penester juga seringkali mengandalkan berbagai teknik manual dalam melakukan pengujian keamanan. Teknik-teknik ini dapat mencakup pengujian penetrasi manual, di mana penester secara aktif mencoba mengeksploitasi kerentanan potensial, serta teknik seperti social engineering dan phishing, yang berfokus pada faktor manusia dalam keamanan. Meski lebih memakan waktu, pengujian manual seringkali diperlukan untuk menemukan kerentanan yang mungkin terlewat oleh alat otomatis.

11.8.3 Melakukan Pengujian Penetrasi

Pengujian penetrasi, atau pen testing, adalah praktik untuk mengevaluasi keamanan sistem atau jaringan dengan mencoba mengeksploitasi kerentanan yang ada. Tujuannya adalah untuk mengidentifikasi kerentanan sebelum penyerang yang berpotensi melakukannya. Pengujian penetrasi dapat mencakup berbagai teknik, mulai dari percobaan untuk mendapatkan akses tanpa izin, melakukan serangan *Denial of Service* (DoS), hingga mencoba mendapatkan data sensitif.

Pada dasarnya, pengujian penetrasi melibatkan empat tahap utama: perencanaan dan pengumpulan informasi, pemindaian, mendapatkan akses, dan mempertahankan akses. Pada tahap perencanaan, pen tester mengumpulkan informasi tentang target dan menentukan ruang lingkup dan tujuan pengujian. Tahap pemindaian melibatkan identifikasi poin akses ke target dan pemindaian untuk kerentanan. Setelah kerentanan telah diidentifikasi, tahap mendapatkan akses melibatkan usaha untuk mengeksploitasi kerentanan tersebut. Tahap terakhir, mempertahankan akses, melibatkan upaya untuk tetap berada dalam sistem selama mungkin untuk simulasi ancaman yang nyata.

Meski pengujian penetrasi dapat sangat efektif dalam mendeteksi kerentanan, perlu diingat bahwa ada batasannya. Pengujian penetrasi biasanya berfokus pada kerentanan yang

diketahui dan dapat dieksploitasi, dan mungkin tidak selalu dapat mendeteksi ancaman baru atau tidak diketahui. Oleh karena itu, pengujian penetrasi sebaiknya digunakan sebagai bagian dari strategi keamanan yang lebih luas yang mencakup berbagai bentuk evaluasi dan mitigasi risiko.

11.8.4 Pemantauan Keamanan Dan Perbaikan Kerentanan

Pemantauan keamanan dan perbaikan kerentanan adalah dua komponen penting dalam strategi keamanan yang efektif. Pemantauan keamanan melibatkan pengawasan yang berkelanjutan atas sistem dan jaringan untuk mendeteksi ancaman dan aktivitas mencurigakan. Ini dapat mencakup berbagai tugas, mulai dari pemantauan lalu lintas jaringan dan log sistem, hingga analisis ancaman dan tanggapan insiden.

Salah satu alat yang penting dalam pemantauan keamanan adalah *Security Information And Event Management* (SIEM). SIEM mengumpulkan dan menganalisis data dari berbagai sumber di dalam organisasi untuk membantu mendeteksi, memonitor, dan melaporkan ancaman keamanan. SIEM dapat membantu mengidentifikasi pola dan tren yang mungkin mengindikasikan aktivitas berbahaya atau mencurigakan, memungkinkan tim keamanan untuk merespons lebih cepat.

Pemantauan keamanan sendiri tidak cukup. Ketika kerentanan atau ancaman terdeteksi, penting untuk melakukan tindakan perbaikan secepat mungkin. Perbaikan kerentanan dapat mencakup berbagai tugas, mulai dari memperbarui dan memperbaiki perangkat lunak, mengubah konfigurasi sistem, hingga pelatihan kembali karyawan untuk mengurangi risiko kesalahan manusia. Penting untuk diingat bahwa keamanan adalah proses berkelanjutan, dan pemantauan dan perbaikan harus menjadi bagian dari siklus hidup keamanan yang berkelanjutan.

DAFTAR PUSTAKA

- Abdullayev, V. and Chauhan, D.A.S. 2023. 'SQL Injection Attack: Quick View', *Mesopotamian Journal of Cyber Security*, (February), pp. 30–34. Available at: <https://doi.org/10.58496/mjcs/2023/006>.
- Abikoye, O.C. *et al.* 2020. 'A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm', *EURASIP Journal on Information Security*, 2020(1), p. 14. Available at: <https://doi.org/10.1186/s13635-020-00113-y>.
- Barth, A., Jackson, C. and Mitchell, J.C. 2008. 'Robust defenses for cross-site request forgery', in *Proceedings of the 15th ACM conference on Computer and communications security*. New York, NY, USA: ACM, pp. 75–88. Available at: <https://doi.org/10.1145/1455770.1455782>.
- Baumgarten, C., Socher, B. and Yule, J. 2023. *Private equity The rising cost of cyberattacks About the authors*.
- Dierks, T. and Rescorla, E. 2008. *The Transport Layer Security (TLS) Protocol Version 1.2, Network Working Group*. Available at: <https://datatracker.ietf.org/doc/html/rfc5246>.
- Fianty, M.I. 2023. 'THE IMPACT OF EMPLOYEES' INFORMATION SECURITY AWARENESS ON INFORMATION SECURITY BEHAVIOUR', *International Journal of Information System and Technology*, 6(5). Available at: <https://doi.org/10.30645/ijistech.v6i5.280>.
- Green, M. and Smith, M. 2016. 'Developers are Not the Enemy!: The Need for Usable Security APIs', *IEEE Security & Privacy*, 14(5), pp. 40–46. Available at: <https://doi.org/10.1109/MSP.2016.111>.

- Halfond, W.G.J., Viegas, J. and Orso, A. 2006. 'A Classification of SQL-Injection Attacks and Countermeasures'.
- Hylender, C.D. *et al.* 2023. *DBIR 2023 Data Breach Investigation Report*.
- Johns, M. 2006. 'SessionSafe: Implementing XSS Immune Session Handling', in, pp. 444–460. Available at: https://doi.org/10.1007/11863908_27.
- Kirda, E. *et al.* 2006. 'Noxes: a client-side solution for mitigating cross-site scripting attacks', in *Proceedings of the 2006 ACM symposium on Applied computing*. New York, NY, USA: ACM, pp. 330–337. Available at: <https://doi.org/10.1145/1141277.1141357>.
- Li, R. *et al.* 2022. 'Cyber Security', in *Communications in Computer and Information Science*, pp. 226–234. Available at: https://doi.org/10.1007/978-981-16-9229-1_14.
- McKeay, M. 2020. *Loyalty for sale: Retail and Hospitality Fraud, Hospitality Upgrade*. Available at: <https://www.akamai.com/our-thinking/the-state-of-the-internet/state-of-the-internet-security-reports-2020>.
- Michal Zalewski. 2011. *The Tangled Web: A Guide to Securing Modern Web Applications*. 1st edn. No Starch Press.
- Mirkovic, J. and Reiher, P. 2004. 'A taxonomy of DDoS attack and DDoS defense mechanisms', *ACM SIGCOMM Computer Communication Review*, 34(2), pp. 39–53. Available at: <https://doi.org/10.1145/997150.997156>.
- Ogundele, I.O., Akinade, A.O. and Alakiri, H.O. 2020. 'Detection and Prevention of Session Hijacking in Web Application Management', *IJARCCCE*, 9(7), pp. 1–10. Available at: <https://doi.org/10.17148/IJARCCCE.2020.9601>.
- Reaves, B. *et al.* 2017. 'Mo(bile) Money, Mo(bile) Problems', *ACM Transactions on Privacy and Security*, 20(3), pp. 1–31. Available at: <https://doi.org/10.1145/3092368>.

- Rodríguez, G.E. *et al.* 2019. 'Cross-site scripting (XSS) attacks and mitigation: A survey', *Computer Networks*, 166(November). Available at: <https://doi.org/10.1016/j.comnet.2019.106960>.
- Schneier, B. 1996. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley.
- Shahriar, H. and Zulkernine, M. 2012. 'Mitigating program security vulnerabilities', *ACM Computing Surveys*, 44(3), pp. 1–46. Available at: <https://doi.org/10.1145/2187671.2187673>.
- Stallings, W. and Brown, L. 2018. *Computer Security: Principles and Practice*. 4th edn. Pearson.
- Sterbenz, J.P.G. *et al.* 2002. 'Survivable mobile wireless networks', in *Proceedings of the 1st ACM workshop on Wireless security*. New York, NY, USA: ACM, pp. 31–40. Available at: <https://doi.org/10.1145/570681.570685>.
- Turpin, K. and Gadsden, J. 2010. *OWASP Secure Coding Practices-Quick Reference Guide*. Available at: <https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/> (Accessed: 1 July 2023).
- Yan, J. *et al.* 2004. 'Password memorability and security: empirical results', *IEEE Security & Privacy Magazine*, 2(5), pp. 25–31. Available at: <https://doi.org/10.1109/MSP.2004.81>.
- Zeller, W., W., E. and Felten. 2008. 'Cross-Site Request Forgeries: Exploitation and Prevention'.
- Zhong, W. (no date) *Command Injection*. Available at: https://owasp.org/www-community/attacks/Command_Injection.

BAB 12

TREND DAN PERKEMBANGAN TERKINI DALAM PEMROGRAMAN WEB

Oleh Angga Aditya Permana

12.1 Pendahuluan



Gambar 12.1. Ilustrasi Tren Pemrograman Web

Tren saat ini untuk pemrograman berbasis web terus berkembang seiring dengan kemajuan teknologi dan kebutuhan pasar. Berikut adalah beberapa tren yang sedang terjadi dalam pemrograman berbasis web: (Hidayatullah, Priyanto, 2014)

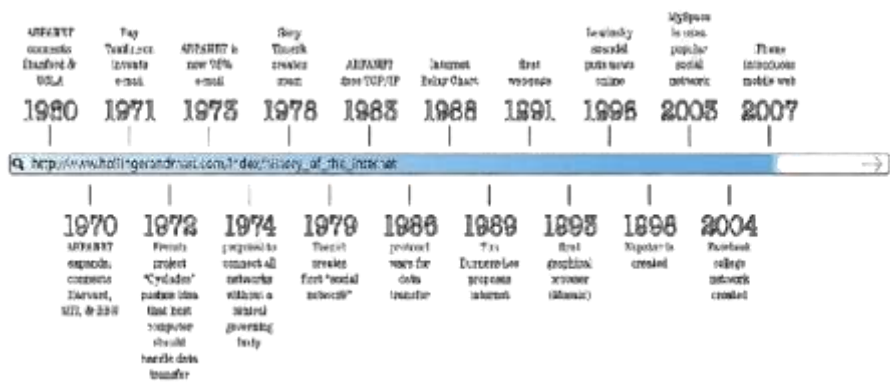
1. *Single-Page Applications (SPA): Single-Page Applications* terus menjadi tren yang populer. SPA memungkinkan pengembang untuk membuat aplikasi web yang berjalan di satu halaman tanpa perlu memuat ulang seluruh halaman saat pengguna berinteraksi. Ini memberikan pengalaman pengguna yang lebih cepat dan responsif.
2. *Progressive Web Apps (PWA): Progressive Web Apps* menggabungkan keunggulan web dan aplikasi native. Mereka dapat diinstal pada perangkat pengguna, bekerja secara offline, dan memberikan pengalaman yang mirip dengan aplikasi native. PWA semakin populer karena dapat meningkatkan retensi pengguna dan memberikan pengalaman pengguna yang lebih baik.
3. *Serverless Architecture: Serverless Architecture* (arsitektur tanpa server) sedang naik daun. Dalam model ini, pengembang tidak perlu mengelola infrastruktur server secara langsung. Fungsi cloud (*cloud functions*) digunakan untuk menjalankan kode berdasarkan permintaan, sehingga mengurangi kompleksitas operasional.
4. *Microservices: Pendekatan pengembangan berbasis mikro layanan (microservices)* terus berkembang. Dalam arsitektur ini, aplikasi dibangun sebagai kumpulan layanan yang independen, terpisah, dan dapat dikembangkan secara terpisah. Ini memungkinkan skalabilitas dan fleksibilitas yang lebih baik dalam pengembangan dan pemeliharaan aplikasi web.
5. *JavaScript Frameworks: Framework JavaScript* seperti *React*, *Angular*, dan *Vue.js* tetap menjadi tren utama dalam pengembangan aplikasi web. Mereka menyediakan alat dan struktur yang kuat untuk membangun antarmuka pengguna yang interaktif dan kompleks.
6. *Machine Learning dan Artificial Intelligence: Penggunaan machine learning dan kecerdasan buatan* semakin terintegrasi dengan pemrograman berbasis web.

Pengembang memanfaatkan library dan alat seperti TensorFlow.js untuk membangun model pembelajaran mesin di sisi klien. Ini memungkinkan aplikasi web untuk memberikan pengalaman yang lebih personal dan cerdas kepada pengguna.

7. *Mobile-First Design*: Dengan penggunaan web melalui perangkat seluler yang terus meningkat, desain mobile-first menjadi prioritas. Pengembang berfokus pada membangun antarmuka pengguna yang optimal untuk perangkat seluler terlebih dahulu, dan kemudian meningkatkannya untuk tampilan desktop.
8. *Web Assembly (Wasm)*: *Web Assembly* adalah format biner yang dapat dieksekusi di browser modern. Ini memungkinkan pengembang untuk menulis kode menggunakan bahasa pemrograman selain JavaScript, seperti C++ atau Rust, dan menjalankannya dengan performa hampir setara dengan kode asli.
9. *Keamanan dan Privasi*: Keamanan dan privasi menjadi perhatian utama dalam pemrograman berbasis web. Pengembang perlu mengimplementasikan langkah-langkah keamanan seperti enkripsi data, perlindungan terhadap serangan siber, dan pengaturan privasi yang kuat sesuai dengan regulasi yang berlaku.

Itulah beberapa tren saat ini dalam pemrograman berbasis web. Tetap mengikuti tren ini dan memperbarui pengetahuan teknis Anda adalah penting untuk mengembangkan aplikasi web yang inovatif dan relevan.

12.2 Sejarah Pemrograman Website



Gambar 12.2. Sejarah Perkembangan Web

Sejarah pemrograman web dimulai pada awal tahun 1990-an. Berikut adalah beberapa peristiwa penting dalam perkembangan pemrograman web: (Hidayatullah, Priyanto, 2014)

1. *Tim Berners-Lee dan World Wide Web (WWW)*: Pada tahun 1989, Tim Berners-Lee, seorang ilmuwan komputer di CERN (Organisasi Eropa untuk Riset Nuklir), menciptakan konsep *World Wide Web (WWW)*. Dia mengembangkan protokol HTTP (*Hypertext Transfer Protocol*) dan HTML (*Hypertext Markup Language*) sebagai dasar untuk berbagi informasi dan dokumen melalui internet.
2. *Mosaic Browser*: Pada tahun 1993, *National Center for Supercomputing Applications (NCSA)* merilis Mosaic, yang dianggap sebagai browser web pertama yang populer. Mosaic memperkenalkan antarmuka grafis untuk browsing web dan mendukung gambar, teks, dan tautan dalam satu dokumen.
3. *Bahasa Pemrograman*: Pada awalnya, web hanya menggunakan HTML untuk mengatur tampilan dan

- struktur halaman. Namun, dengan berkembangnya web, bahasa pemrograman seperti JavaScript, PHP, dan Perl mulai digunakan untuk memberikan interaktivitas dan fungsionalitas yang lebih kompleks.
4. *Dot-com Boom*: Pada pertengahan hingga akhir 1990-an, terjadi ledakan situs web komersial yang dikenal sebagai dot-com boom. Perusahaan mulai menyadari potensi bisnis yang dapat ditawarkan oleh web, dan banyak perusahaan internet didirikan dalam periode ini.
 5. *Web 2.0*: Pada awal 2000-an, konsep Web 2.0 muncul. Ini menggambarkan pergeseran dari web yang statis menjadi web yang lebih interaktif dan kolaboratif. Situs web seperti Wikipedia, YouTube, dan Facebook mulai muncul, memungkinkan pengguna untuk berkontribusi dan berbagi konten mereka.
 6. *Mobile Web dan Responsif Desain*: Dengan munculnya smartphone dan tablet, penggunaan web melalui perangkat seluler meningkat pesat. Pengembang mulai fokus pada pengembangan situs web yang responsif, yang dapat menyesuaikan tampilan dengan baik di berbagai perangkat.
 7. *HTML5 dan CSS3*: Pada tahun 2014, HTML5 dan CSS3 diperkenalkan sebagai standar baru dalam pengembangan web. HTML5 membawa elemen dan atribut baru yang lebih kuat, termasuk dukungan untuk video dan audio. CSS3 memperkenalkan kemampuan desain yang lebih maju, seperti animasi, bayangan, dan transisi.
 8. *Single-Page Applications (SPA)*: Konsep Single-Page Applications mulai populer pada tahun 2010-an. SPA memungkinkan pengguna untuk menjelajahi aplikasi web di satu halaman tanpa harus memuat ulang halaman yang lengkap. Ini memberikan pengalaman pengguna yang lebih cepat dan responsif.

9. *Internet of Things* (IoT): Pada beberapa tahun terakhir, pemrograman web telah berkembang ke arah *Internet of Things* (IoT). Pengembang menggunakan teknologi web untuk menghubungkan perangkat fisik seperti lampu, kulkas, dan pengunci pintu ke internet, memungkinkan kontrol dan pengawasan jarak jauh.

Itulah beberapa peristiwa penting dalam sejarah pemrograman web. Pemrograman web terus berkembang dengan adanya teknologi baru dan tren yang muncul, seperti kecerdasan buatan, realitas virtual, dan blockchain.

12.3 Kebutuhan Pasar Terhadap Pengembangan Website



Gambar 12.3. Ilustrasi Kebutuhan Pasar

Kebutuhan pasar terhadap pengembangan website terus berkembang seiring dengan perubahan tren dan permintaan pengguna. Berikut adalah beberapa kebutuhan pasar yang

relevan dalam pengembangan website: (Hidayatullah, Priyanto, 2014)

1. Responsif dan *Mobile-Friendly*: Dengan semakin banyaknya pengguna yang mengakses web melalui perangkat seluler, kebutuhan untuk memiliki website yang responsif dan mobile-friendly sangat penting. Website harus mampu menyesuaikan tampilan dengan baik di berbagai ukuran layar dan perangkat.
2. Kecepatan Muat yang Cepat: Pengguna modern mengharapkan website untuk memuat dengan cepat. Kebutuhan akan kecepatan muat yang cepat adalah kunci dalam memberikan pengalaman pengguna yang baik. Pengembang harus mengoptimalkan kinerja website, menggunakan teknik seperti caching, kompresi file, dan pemrosesan sisi klien untuk mengurangi waktu muat.
3. Keamanan dan Perlindungan Data: Dalam era yang semakin kompleks dari serangan siber dan pelanggaran data, kebutuhan akan keamanan website menjadi sangat penting. Pengembang harus mengimplementasikan praktik keamanan yang kuat, seperti enkripsi data, penggunaan protokol HTTPS, dan perlindungan terhadap serangan yang umum, seperti serangan injeksi SQL atau serangan *cross-site scripting* (XSS).
4. *User Experience* yang Baik: Pengguna mengharapkan pengalaman pengguna yang baik saat mengunjungi website. Kebutuhan akan antarmuka yang intuitif, navigasi yang mudah dipahami, desain yang menarik, dan konsistensi visual semuanya merupakan faktor penting dalam memberikan pengalaman pengguna yang memuaskan.
5. Integrasi dengan Platform dan Layanan Eksternal: Website sering kali perlu terintegrasi dengan platform dan layanan eksternal, seperti sistem manajemen konten (CMS), sistem pembayaran, analitik, media sosial, dan API pihak ketiga.

Kebutuhan akan integrasi yang lancar dan dapat diandalkan adalah hal yang penting dalam mengembangkan website yang dapat berfungsi secara efektif dalam ekosistem yang lebih luas.

6. Optimisasi SEO: Kebutuhan untuk mengoptimalkan website agar mudah ditemukan di mesin pencari (SEO) tetap menjadi faktor penting dalam strategi pemasaran digital. Pengembang harus memperhatikan praktik pengkodean yang SEO-friendly, seperti penggunaan tag heading yang benar, penggunaan kata kunci yang tepat, dan pemuatan cepat halaman.
7. Analitik dan Pelacakan: Pengembang harus menyediakan kemampuan pelacakan dan analitik yang memadai pada website. Ini memungkinkan pemilik website untuk memahami perilaku pengguna, mengidentifikasi tren, dan mengambil keputusan berdasarkan data yang relevan.
8. Aksesibilitas: Dalam upaya untuk mencakup semua pengguna, kebutuhan akan aksesibilitas yang baik semakin meningkat. Pengembang harus memperhatikan panduan aksesibilitas web, seperti *Web Content Accessibility Guidelines* (WCAG), untuk memastikan bahwa website dapat diakses oleh pengguna dengan berbagai kebutuhan dan disabilitas.
9. Integrasi dengan Aplikasi Mobile dan Teknologi Terkini: Kebutuhan akan integrasi antara website dan aplikasi mobile, teknologi baru seperti kecerdasan buatan (AI), *Internet of Things* (IoT), atau realitas virtual dan *augmented reality* (VR/AR) semakin meningkat. Pengembang harus mampu memanfaatkan potensi teknologi terkini dan menyediakan pengalaman yang terhubung secara mulus di seluruh platform.

Dengan memahami dan memenuhi kebutuhan pasar ini, pengembang website dapat memberikan solusi yang relevan, inovatif, dan memuaskan untuk pengguna dan pemilik bisnis.

12.4 Tujuh Frameworks Populer



Gambar 12.4. Framework Populer

Ada banyak frameworks yang dapat digunakan untuk mengembangkan dan membuat website. Di bawah ini, saya akan menjelaskan beberapa frameworks yang populer dan sering digunakan dalam pengembangan website, beserta fitur-fitur utama dan kegunaannya: (Rosmala, Ichwan and Gandalisha, 2011)

1. React.js:

- a. Deskripsi: React.js adalah *library JavaScript* yang digunakan untuk membangun antarmuka pengguna yang interaktif dan responsif.
- b. React.js menggunakan konsep komponen yang memisahkan tampilan menjadi bagian-bagian yang dapat digunakan kembali. Setiap komponen memiliki state dan props yang dapat diperbarui untuk mengatur tampilan. Ketika ada perubahan pada state atau props,

React akan memperbarui Virtual DOM dan membandingkannya dengan DOM aktual untuk melakukan perubahan yang efisien.

c. Fitur Utama:

- a) Komponen berbasis: Memungkinkan pemisahan logika aplikasi ke dalam komponen-komponen yang dapat digunakan kembali, meningkatkan efisiensi pengembangan.
- b) Virtual DOM: Menggunakan Virtual DOM untuk memperbarui hanya bagian-bagian yang berubah, memberikan kinerja yang lebih baik.
- c) Ekosistem yang kuat: Dukungan dan library pihak ketiga yang melimpah, seperti React Router untuk routing dan Redux untuk manajemen state.

2. Angular:

- a. Deskripsi: Angular adalah framework JavaScript yang dikembangkan oleh Google untuk membangun aplikasi web yang kompleks.
- b. Angular menggunakan pola *Model-View-Controller* (MVC) untuk memisahkan logika aplikasi menjadi tiga komponen utama. Model mengelola data aplikasi, View mengatur tampilan yang ditampilkan kepada pengguna, dan Controller mengatur interaksi antara Model dan View. Angular menggunakan *two-way* data binding untuk memperbarui secara otomatis tampilan saat ada perubahan pada model, serta menggunakan *dependency injection* untuk pengelolaan dependensi.
- c. Fitur Utama:
 - a) Arsitektur MVC: Menggunakan pola *Model-View-Controller* untuk memisahkan logika aplikasi dan antarmuka pengguna.

- b) *Two-way data binding*: Menyediakan pembaruan otomatis antara model dan tampilan, mengurangi kode yang harus ditulis.
- c) *Dependency Injection*: Memudahkan pengelolaan dependensi dan pengujian unit.
- d) *Routing* dan modularitas: Menyediakan sistem routing yang kuat dan memungkinkan modularitas dalam pengembangan aplikasi.

3. Vue.js:

- a. Deskripsi: Vue.js adalah framework JavaScript yang mudah dipelajari dan digunakan untuk membangun aplikasi web yang interaktif.
- b. Vue.js juga menggunakan konsep komponen, di mana aplikasi dibangun dari komponen yang dapat digunakan kembali. Setiap komponen memiliki template yang menggabungkan HTML dan sintaksis Vue untuk mengatur tampilan. Vue.js menggunakan Virtual DOM untuk memperbarui tampilan dengan efisien, serta menyediakan fitur-fitur seperti Vue Router untuk sistem routing dan Vuex untuk manajemen state aplikasi.
- c. Fitur Utama:
 - a) Struktur komponen: Memungkinkan pemisahan logika menjadi komponen-komponen yang terfokus dan dapat digunakan kembali.
 - b) Vue Router: Menyediakan sistem routing yang mudah digunakan.
 - c) Vuex: State management library yang memudahkan manajemen state aplikasi.
 - d) Virtual DOM: Menggunakan Virtual DOM untuk mencapai kinerja yang cepat.

4. Laravel:

- a. Deskripsi: Laravel adalah *framework* PHP yang kuat untuk pengembangan aplikasi web.
- b. Laravel adalah *framework* PHP yang mengikuti pola *Model-View-Controller* (MVC). Pengembang dapat membuat model yang mewakili entitas dalam aplikasi, view yang mengatur tampilan yang diperlihatkan kepada pengguna, dan controller yang mengatur interaksi antara model dan view. Laravel menggunakan routing yang kuat untuk menghubungkan URL dengan aksi di controller, serta menyediakan berbagai fitur seperti ORM untuk berinteraksi dengan database dan fitur keamanan bawaan.
- c. Fitur Utama:
 - a) MVC Architecture: Menggunakan pola *Model-View-Controller* untuk memisahkan logika aplikasi.
 - b) Routing yang kuat: Memudahkan manajemen rute dalam aplikasi.
 - c) ORM (*Object-Relational Mapping*): Memungkinkan interaksi dengan database menggunakan kode PHP alih-alih SQL mentah.
 - d) *Security*: Menyediakan fitur keamanan seperti hashing *password* dan perlindungan terhadap serangan CSRF.

5. Django:

- a. Deskripsi: Django adalah *framework* Python yang populer untuk pengembangan web.
- b. Django menggunakan pendekatan *Model-View-Controller* (MVC) dengan nama yang berbeda: *Model-View-Template* (MVT). Model mewakili data aplikasi, View mengatur tampilan yang ditampilkan kepada pengguna, dan Template mengatur tampilan menggunakan sintaksis Django. Django menyediakan

sistem routing yang kuat, fitur-fitur seperti ORM untuk berinteraksi dengan database, dan kemampuan bawaan untuk administrasi konten.

- c. Fitur Utama:
 - a) Object-Relational Mapping (ORM): Memungkinkan pengembang untuk berinteraksi dengan database menggunakan kode Python.
 - b) Admin Interface: Menyediakan antarmuka administrasi yang siap pakai untuk mengelola konten dan data.
 - c) Keamanan: Memiliki fitur keamanan yang terintegrasi, seperti perlindungan terhadap serangan injeksi SQL dan CSRF.
 - d) Scalability: Dibangun dengan skala dalam pikiran, memungkinkan pengembangan aplikasi web yang mudah diperluas.

6. Ruby on Rails:

- a. Deskripsi: *Ruby on Rails (Rails)* adalah *framework Ruby* yang sangat populer untuk pengembangan aplikasi web.
- b. Ruby on Rails mengikuti pola *Model-View-Controller (MVC)* untuk memisahkan logika aplikasi. Model mewakili data dalam aplikasi, View mengatur tampilan, dan **Controller** mengatur aliran informasi antara Model dan View. Rails menggunakan pendekatan konvensi daripada konfigurasi, yang berarti pengembang hanya perlu menentukan perubahan dari konvensi default untuk mengatur aplikasi. Rails menyediakan fitur-fitur seperti scaffolding untuk pembuatan cepat fitur standar dan sistem migrasi database.
- c. Fitur Utama:
 - a) Konvensi daripada Konfigurasi: Menyediakan kerangka kerja yang konvensional dengan konfigurasi minimal, mempercepat pengembangan.

- b) MVC Architecture: Memisahkan logika aplikasi ke dalam Model, View, dan Controller.
- c) Scaffolding: Memungkinkan pembuatan cepat fitur-fitur standar seperti CRUD (*Create, Read, Update, Delete*).
- d) Database Migrations: Memudahkan pengembangan database dan perubahan skema dengan lancar.

7. Flask:

- a. Deskripsi: *Flask* adalah *framework Python* yang ringan dan fleksibel untuk membangun aplikasi web.
- b. *Flask* adalah *framework Python* yang ringan dan fleksibel. Dibandingkan dengan framework lain yang lebih besar, Flask memberikan kebebasan lebih besar kepada pengembang untuk memilih komponen yang diperlukan. Flask menggunakan routing sederhana untuk menghubungkan URL dengan fungsi tampilan. Pengembang dapat membuat template dengan menggunakan pemrograman template untuk menghasilkan tampilan dinamis. *Flask* juga memiliki integrasi yang baik dengan *library Python* lainnya.
- c. Fitur Utama:
 - a) Mikro framework: Menyediakan inti yang minimalis dan fleksibel, memungkinkan pengembang untuk memilih komponen yang diperlukan.
 - b) Routing yang sederhana: Mendefinisikan rute dengan mudah dan menghubungkannya dengan fungsi tampilan.
 - c) Pemrograman template: Memungkinkan penggunaan template untuk membuat tampilan halaman web yang dinamis.
 - d) Integrasi yang baik dengan library Python lainnya.

Itulah beberapa frameworks yang sering digunakan dalam pengembangan website. Pemilihan framework yang tepat tergantung pada kebutuhan proyek, preferensi bahasa pemrograman, dan skala aplikasi yang ingin dikembangkan. Setiap framework memiliki kelebihan dan kekurangan sendiri-sendiri, dan penting untuk mempelajari dan memahami fitur-fitur utama serta dokumentasi yang disediakan oleh setiap framework.

12.5 Contoh Penggunaan Framework Populer dalam Studi Kasus Pembuatan E-Commerce



Gambar 12.5. Ilustrasi E-Commerce

Berikut adalah contoh sederhana menggunakan masing-masing *framework* dalam pembuatan website e-commerce: (Rosmala, Ichwan and Gandalisha, 2011)

1. Contoh React.js untuk e-commerce:

- a. Penggunaan React.js dalam e-commerce dapat melibatkan komponen seperti "*ProductList*" yang menampilkan daftar produk, "*ProductDetail*" yang

menampilkan detail produk, dan "ShoppingCart" yang mengelola keranjang belanja.

- b. Komponen tersebut akan berinteraksi dengan state dan props untuk menampilkan data produk, mengelola jumlah produk dalam keranjang, dan memperbarui tampilan secara dinamis saat pengguna berinteraksi dengan situs.
- c. React.js akan membantu dalam membuat tampilan yang responsif, interaktif, dan memperbarui komponen yang relevan tanpa perlu memuat ulang seluruh halaman.

2. Contoh Angular untuk e-commerce:

- a. Dalam Angular, komponen seperti "*ProductListComponent*" akan menampilkan daftar produk, "*ProductDetailComponent*" akan menampilkan detail produk, dan "*CartComponent*" akan mengelola keranjang belanja.
- b. Angular menggunakan konsep *two-way* data binding untuk memastikan bahwa perubahan dalam model produk atau keranjang belanja secara otomatis diperbarui dalam tampilan.
- c. Angular juga menyediakan fitur routing untuk mengarahkan pengguna ke halaman detail produk, keranjang belanja, dan halaman pembayaran.

3. Contoh Vue.js untuk e-commerce:

- a. Dalam Vue.js, komponen seperti "*ProductList*" akan menampilkan daftar produk, "*ProductDetail*" akan menampilkan detail produk, dan "*Cart*" akan mengelola keranjang belanja.
- b. Vue.js menggunakan fitur reaktivitas untuk memastikan bahwa perubahan dalam model produk atau keranjang belanja secara otomatis diperbarui dalam tampilan.

- c. Vue.js menyediakan kemudahan dalam mengatur rute dengan menggunakan Vue Router, sehingga pengguna dapat melihat detail produk, mengelola keranjang belanja, dan melanjutkan ke halaman pembayaran.

4. Contoh Laravel untuk e-commerce:

- a. Dalam Laravel, pengembang dapat menggunakan konsep *Model-View-Controller* (MVC) untuk mengorganisir logika aplikasi.
- b. Model dalam Laravel akan mewakili entitas seperti produk dan keranjang belanja, serta berinteraksi dengan database untuk menyimpan dan mengambil data.
- c. View akan menangani tampilan, seperti halaman daftar produk, halaman detail produk, dan halaman keranjang belanja.
- d. Controller akan mengatur aliran informasi antara model dan view, mengambil data dari model, dan memperbarui tampilan yang relevan.

5. Contoh Django untuk e-commerce:

- a. Dalam Django, pengembang dapat membuat model seperti "Product" dan "Cart" untuk merepresentasikan produk dan keranjang belanja dalam database.
- b. View dalam Django akan menangani permintaan pengguna dan menampilkan tampilan HTML yang sesuai, seperti halaman daftar produk, halaman detail produk, dan halaman keranjang belanja.
- c. Django menggunakan ORM untuk berinteraksi dengan database, sehingga pengembang dapat melakukan operasi seperti menyimpan produk ke dalam keranjang belanja atau mengambil daftar produk dari database.

6. Contoh Ruby on Rails untuk e-commerce:

- a. Dalam Ruby on Rails, pengembang dapat menggunakan model seperti "Product" dan "Cart" untuk merepresentasikan produk dan keranjang belanja dalam aplikasi.
- b. Controller dalam Rails akan mengatur aliran informasi antara model dan view, memproses permintaan pengguna, dan memperbarui tampilan yang relevan.
- c. View dalam Rails akan menampilkan tampilan HTML yang sesuai, seperti halaman daftar produk, halaman detail produk, dan halaman keranjang belanja.

7. Contoh Flask untuk e-commerce:

- a. Dalam Flask, pengembang dapat menggunakan fungsi tampilan untuk menangani permintaan pengguna, seperti menampilkan halaman daftar produk, halaman detail produk, dan halaman keranjang belanja.
- b. Flask memungkinkan pengembang untuk menggunakan pemrograman template untuk menghasilkan tampilan dinamis dengan menyatukan data produk dan keranjang belanja dalam tampilan HTML.
- c. Pengembang juga dapat menggunakan ORM dan database untuk mengelola produk dan keranjang belanja.

Catatan: Contoh-contoh di atas adalah contoh sederhana dan tidak mencakup semua aspek yang terlibat dalam pengembangan website e-commerce. Sebuah proyek e-commerce yang sesungguhnya akan melibatkan lebih banyak fitur seperti proses pembayaran, autentikasi pengguna, pengelolaan persediaan, dan lainnya.

12.6 Bahasa Pemrograman untuk Web Based

Ada beberapa bahasa pemrograman yang dapat digunakan untuk membuat sebuah website. Berikut adalah penjelasan rinci tentang beberapa bahasa pemrograman yang umum digunakan dalam pengembangan website: (Endra *et al.*, 2021)

1. HTML (*Hypertext Markup Language*):

- a. HTML adalah bahasa markah yang digunakan untuk membangun struktur dan konten dasar sebuah halaman web.
- b. HTML memberikan kerangka dasar untuk tampilan dan memungkinkan penempatan elemen seperti teks, gambar, tautan, formulir, dan lainnya.
- c. Pengembang menggunakan tag HTML untuk mengelompokkan dan mengatur elemen secara hierarkis dalam halaman.

Contoh Script:

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh E-commerce</title>
</head>
<body>
  <h1>Selamat Datang di E-commerce</h1>
  <p>Produk yang tersedia:</p>
  <ul>
    <li>Produk 1</li>
    <li>Produk 2</li>
    <li>Produk 3</li>
  </ul>
</body>
</html>
```

Gambar 12.6. Contoh Script HTML

2. CSS (*Cascading Style Sheets*):

- a. CSS adalah bahasa gaya yang digunakan untuk mengatur tampilan dan layout elemen dalam halaman web.
- b. CSS digunakan untuk menerapkan warna, font, ukuran, margin, padding, dan atribut visual lainnya pada elemen HTML.
- c. Dengan CSS, pengembang dapat membuat tampilan yang konsisten dan menarik, serta mengatur responsivitas halaman web pada berbagai ukuran layar.

Contoh Script:

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh E-commerce</title>
  <style>
    h1 {
      color: blue;
    }
    ul {
      list-style-type: square;
    }
  </style>
</head>
<body>
  <h1>Selamat Datang di E-commerce</h1>
  <p>Produk yang tersedia:</p>
  <ul>
    <li>Produk 1</li>
    <li>Produk 2</li>
    <li>Produk 3</li>
  </ul>
</body>
</html>
```

Gambar 12.7. Contoh Script CSS

3. JavaScript:

- a. JavaScript adalah bahasa pemrograman yang digunakan untuk membuat interaktivitas dan fungsi dalam halaman web.
- b. Dengan JavaScript, pengembang dapat memanipulasi elemen HTML, merespons peristiwa pengguna, memvalidasi formulir, mengatur animasi, berkomunikasi dengan server melalui AJAX, dan banyak lagi.
- c. JavaScript juga digunakan dalam pengembangan aplikasi web berbasis kerangka kerja (*framework*) seperti React, Angular, dan Vue.js.

Contoh Script:

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh E-commerce</title>
</head>
<body>
  <h1>Selamat Datang di E-commerce</h1>
  <p>Produk yang tersedia:</p>
  <ul id="product-list">
    <li>Produk 1</li>
    <li>Produk 2</li>
    <li>Produk 3</li>
  </ul>

  <script>
    // Script untuk menambahkan event click pada setiap produk
    var productList = document.getElementById('product-list');
    var products = productList.getElementsByTagName('li');

    for (var i = 0; i < products.length; i++) {
      products[i].addEventListener('click', function() {
        alert('Anda memilih ' + this.innerText);
      });
    }
  </script>
</body>
</html>
```

Gambar 12.8. Contoh Script JavaScript

4. PHP (Hypertext Preprocessor):

- PHP adalah bahasa pemrograman server-side yang dirancang khusus untuk pengembangan web.
- PHP memungkinkan pengembang untuk membuat aplikasi web yang dinamis dengan kemampuan seperti pengolahan formulir, akses ke database, pengelolaan sesi, pengiriman email, dan banyak lagi.
- PHP biasanya dijalankan di sisi server dan menghasilkan output HTML yang akan dikirimkan ke browser.

Contoh Script:

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh E-commerce</title>
</head>
<body>
  <h1>Selamat Datang di E-commerce</h1>
  <p>Produk yang tersedia:</p>
  <ul>
    <?php
      $products = array("Produk 1", "Produk 2", "Produk 3");

      foreach ($products as $product) {
        echo "<li>$product</li>";
      }
    ?>
  </ul>
</body>
</html>
```

Gambar 12.9. Contoh Script PHP

5. Python:

- Python adalah bahasa pemrograman yang serbaguna dan dapat digunakan untuk pengembangan web.
- Dalam pengembangan web, Python sering digunakan dengan framework seperti Django dan Flask.
- Django adalah framework web Python yang kuat, dengan fitur-fitur seperti ORM (Object-Relational Mapping), sistem routing yang kuat, dan fitur keamanan bawaan.
- Flask adalah framework web Python yang lebih ringan, memberikan fleksibilitas yang lebih besar dalam memilih komponen dan berintegrasi dengan library Python lainnya.

Contoh Script:

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def index():
    products = ["Produk 1", "Produk 2", "Produk 3"]
    return render_template('index.html', products=products)

if __name__ == '__main__':
    app.run()
```

Gambar 12.10. Contoh Script Python

6. Ruby:

- Ruby adalah bahasa pemrograman yang sederhana dan elegan yang dapat digunakan untuk pengembangan web.
- Ruby on Rails* (Rails) adalah *framework* web Ruby yang populer dan mengikuti pola *Model-View-Controller* (MVC).

- c. Rails menyediakan struktur yang konvensional dengan konfigurasi minimal, yang mempercepat proses pengembangan.
- d. Rails memungkinkan pengembang untuk dengan cepat membangun fitur-fitur standar seperti CRUD (Create, Read, Update, Delete) dan memiliki integrasi yang baik dengan database.

Contoh Script:

```
# routes.rb
Rails.application.routes.draw do
  root 'products#index'
end

# products_controller.rb
class ProductsController < ApplicationController
  def index
    @products = ["Produk 1", "Produk 2", "Produk 3"]
  end
end

# index.html.erb
<!DOCTYPE html>
<html>
<head>
  <title>Contoh E-commerce</title>
</head>
<body>
  <h1>Selamat Datang di E-commerce</h1>
  <p>Produk yang tersedia:</p>
  <ul>
    <% @products.each do |product| %>
      <li><%= product %></li>
    <% end %>
  </ul>
</body>
</html>
```

Gambar 12.11. Contoh Script Ruby

7. Java:

- a. Java adalah bahasa pemrograman yang banyak digunakan dalam pengembangan web.
- b. Dalam pengembangan web, Java sering digunakan dengan framework seperti Spring dan *JavaServer Faces* (JSF).
- c. Spring adalah framework Java yang kuat dan kaya fitur, dengan dukungan untuk pengembangan aplikasi web berbasis MVC dan manajemen dependensi.
- d. JSF adalah framework Java yang fokus pada pengembangan antarmuka pengguna web, dengan komponen siap pakai dan dukungan untuk validasi dan manajemen form.

Contoh Script:

```
import java.util.ArrayList;
import java.util.List;

public class Ecommerce {
    public static void main(String[] args) {
        System.out.println("Selamat Datang di E-commerce");
        System.out.println("Produk yang tersedia:");

        List<String> products = new ArrayList<>();
        products.add("Produk 1");
        products.add("Produk 2");
        products.add("Produk 3");

        for (String product : products) {
            System.out.println(product);
        }
    }
}
```

Gambar 12.12. Contoh Script Java

Itulah beberapa bahasa pemrograman yang umum digunakan dalam pengembangan website. Pilihan bahasa

pemrograman yang tepat tergantung pada kebutuhan proyek, preferensi pengembang, dan ekosistem teknologi yang digunakan.

12.7 Tips Menjadi Programer Website



Gambar 12.13. Sistem Organisasi Kehidupan makhluk Hidup

Berikut adalah beberapa tips dasar yang dapat membantu pemula dalam mempelajari pemrograman berbasis website: (Endra *et al.*, 2021)

1. Pilih bahasa pemrograman yang sesuai: Ada banyak bahasa pemrograman yang dapat digunakan untuk pengembangan web. Pilihlah satu bahasa pemrograman yang sesuai dengan minat dan tujuan Anda. Beberapa bahasa populer untuk pengembangan web termasuk HTML, CSS, JavaScript, Python, dan PHP.
2. Mulailah dengan HTML dan CSS: HTML adalah bahasa markah yang digunakan untuk membangun struktur dasar

halaman web, sedangkan CSS digunakan untuk mengatur tampilan dan layout halaman. Memahami dasar-dasar HTML dan CSS adalah langkah awal yang penting dalam mempelajari pemrograman web.

3. Pelajari JavaScript: JavaScript adalah bahasa pemrograman yang penting dalam pengembangan web. Pelajari konsep dasar JavaScript seperti variabel, tipe data, operasi, pengontrol aliran, dan fungsi. Kemudian, pelajari juga topik lebih lanjut seperti manipulasi DOM, AJAX, dan kerangka kerja JavaScript populer seperti React, Angular, atau Vue.js.
4. Gunakan sumber belajar yang tepat: Ada banyak sumber belajar yang tersedia secara online, seperti tutorial, buku, video tutorial, dan kursus interaktif. Cari sumber belajar yang mudah dipahami dan sesuai dengan gaya belajar Anda. Situs web seperti MDN Web Docs, W3Schools, dan FreeCodeCamp menyediakan dokumentasi dan tutorial yang bagus untuk pemula.
5. Praktikkan dengan proyek sederhana: Setelah mempelajari dasar-dasar pemrograman web, mulailah membuat proyek sederhana. Misalnya, buat halaman web sederhana, buat formulir, terapkan efek animasi dengan CSS, atau buat interaksi sederhana menggunakan JavaScript. Praktik yang konsisten akan membantu Anda memahami konsep dan memperkuat keterampilan pemrograman.
6. Ikuti komunitas dan forum: Bergabunglah dengan komunitas pemrograman web, baik di forum online maupun kelompok lokal. Komunitas ini dapat memberikan dukungan, tips, dan saran yang berharga dalam perjalanan belajar Anda. Mereka juga dapat membantu menjawab pertanyaan dan mengatasi masalah yang mungkin Anda hadapi.
7. Analisis dan pelajari kode sumber: Selain membaca dan mencoba kode sumber yang ada, penting juga untuk mempelajari kode sumber dari proyek-proyek open-source

yang relevan dengan tujuan Anda. Melalui analisis kode sumber, Anda dapat memahami pendekatan terbaik dalam pengembangan web dan mempelajari praktik terbaik dari pengembang berpengalaman.

8. Terus berlatih dan eksperimen: Pemrograman web melibatkan pembelajaran sepanjang hayat. Selalu berlatih, eksperimen dengan teknologi baru, dan tetap mengikuti tren dan perkembangan terkini dalam industri. Terus mengasah keterampilan Anda dan selalu berusaha untuk meningkatkan diri.

Ingatlah bahwa mempelajari pemrograman berbasis web membutuhkan waktu dan dedikasi. Mulailah dengan dasar-dasar dan bangun fondasi yang kuat, lalu perluas pengetahuan dan keterampilan Anda seiring berjalannya waktu. Jangan takut untuk mencoba dan berpikir kreatif dalam pengembangan web. Selamat belajar!

DAFTAR PUSTAKA

- Endra, R.Y. *et al.* 2021. 'Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website', *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, 11(1), p. 48. Available at: <https://doi.org/10.36448/expert.v11i1.2012>.
- Hidayatullah, Priyanto, J.K.K. 2014. 'Pemrograman WEB'.
- Rosmala, D., Ichwan, M. and Gandalisha, M.I. 2011. 'Komparasi Framework Mvc(Codeigniter, Dan Cakephp) Pada Aplikasi Berbasis Web ', *Jurnal Informatik*, 2(8), pp. 22–30.

BIODATA PENULIS



Ade Maulana, S.Kom., M.T.I., CITSM

Dosen Program Studi Sistem Informasi
Fakultas Ilmu Komputer Universitas Pelita Harapan
Kampus Kota Medan

Ade Maulana merupakan dosen Program Studi Sistem Informasi Universitas Pelita Harapan kampus kota Medan. Ia tercatat sebagai lulusan Magister Teknologi Informasi Universitas Indonesia. Sebelum mengajar, dia bekerja di beberapa perusahaan pada bidang teknologi informasi.

BIODATA PENULIS



Rahmat Taufik R.L Bau, M.Kom.

Dosen Program Studi Sistem Informasi
Fakultas Teknik Universitas Negeri Gorontalo

Penulis lahir di Gorontalo tanggal 27 Juli 1989. Penulis adalah seorang dosen pada Program Studi Sistem Informasi Fakultas Teknik, Universitas Negeri Gorontalo. Menyelesaikan pendidikan S1 dan S2 pada Jurusan Teknologi Informasi. Penulis mengajar dan meneliti di bidang keahlian analisis dan perancangan. Hobby penulis adalah scrabbling ide desain antarmuka dan menonton video tutorial di youtube. Sejak awal tahun 2023 bergabung di komunitas menulis bab chapter oleh Get Press Indonesia. Beberapa buku yang telah terbit adalah Teknologi Digital di Era Modern dan Layanan Digital di Era 5.0.

BIODATA PENULIS



Hermila A. S.Si., M.Pd.

Dosen Program Studi Pendidikan Teknologi Informasi
Fakultas Teknik Universitas Negeri Gorontalo

Penulis lahir di Polewali Mandar tanggal 25 Juli 1993. Penulis adalah dosen tetap pada Program Studi Pendidikan Teknologi Informasi Fakultas Teknik, Universitas Negeri Gorontalo. Menyelesaikan pendidikan S1 pada Jurusan Sistem Informasi STMIK Profesional Makassar dan melanjutkan S2 pada Jurusan Pendidikan Teknologi Kejuruan Kekhususan Pendidikan Teknik Informatika dan Komputer Universitas Negeri Makassar. Penulis sekarang mulai menekuni bidang Menulis.

BIODATA PENULIS



Zen Munawar.

Dosen Program Studi Manajemen Informatika
Politeknik LP3I Bandung

Penulis memiliki jabatan akademik lektor, dan alumni dari TI UNPAS, Pasca Sarjana AdPen UNINUS, Sistem Informasi dari STMIK IM dan Pasca Sarjana Sistem Informasi STMIK LIKMI Bandung serta mengikuti Program Doktorat Teknologi Informasi dan Komunikasi di UTeM Melaka Malaysia. Penulis aktif menjadi editor board dan reviewer jurnal nasional dan internasional terakreditasi Sinta serta aktif dalam menulis artikel di jurnal nasional maupun internasional serta menulis buku. Selain itu pula penulis aktif sebagai pengurus bidang kerjasama industri pada organisasi APTIKOM Prov. Jabar, sebagai sekretaris DPW Jawa Barat pada Ikatan Ahli Informatika Indonesia (IAII) sebagai Dewan Pengawas pada organisasi Perkumpulan Auditor Teknologi Informasi dan Komunikasi Indonesia (PASTIKINDO). Penulis memiliki kompetensi Information Technology Expert Auditor, ICT Project Manager, System Analyst dan Assesor Of Competency dari BNSP.

BIODATA PENULIS



Rony Setiawan, S.Kom., M.Kom.

Dosen Program Studi Manajemen Informatika
Politeknik LP3I kampus utama Bandung

Penulis lahir di Bandung tanggal 7 Januari 1965. Penulis adalah dosen tetap pada Program Studi Manajemen Informatika Politeknik LP3I Kampus Utama Bandung. Menyelesaikan pendidikan S1 pada Jurusan Sistem Informasi dan melanjutkan S2 pada Jurusan Sistem Informatika. Penulis menekuni bidang Menulis.

Selain menekuni bidang menulis, penulis juga menekuni pembuatan aplikasi berbasis web seperti Learning Management System untuk Politeknik LP3I, Penilaian Kinerja berbasis Object Key Result (OKR), Sistem Informasi Pengelolaan Wisuda, Sistem Informasi Akademik berbasis desktop, dan aplikasi-aplikasi lainnya.

BIODATA PENULIS



Sitti Aisa, S.Kom.,M.T

Dosen Program Studi Teknik Informatika
Universitas Dipa Makassar

Penulis lahir di Ujung Pandang, pada 28 Oktober 1987. Saat menempuh Sekolah Menengah Kejuruan (SMK) pada SMK Negeri 04 Makassar ditahun 2002. Kemudian, melanjutkan Pendidikan S1 di tahun 2005 pada kampus STMIK Dipanegara Makassar program studi Sistem Informasi dan Lulus S1 pada tahun 2009. Kemudian lanjut ditahun 2013 tercatat sebagai Lulusan S2 Teknik Informatika Universitas Hasanuddin. Keseharian selalu di panggil dengan sapaan icha ini terdaftar sebagai dosen tetap pada Universitas Dipa Makassar sejak tahun 2010. Hobi yang disenangi adalah jalan – jalan dan membaca buku. Penulis memiliki kepakaran pada bidang data mining, Pemrograman serta database Management. Selain itu, penulis juga aktif menulis artikel baik pada Jurnal Nasional maupun Jurnal Internasional.

BIODATA PENULIS



Wirawan Istiono, S.Kom., M.Kom.

Dosen Informatika

Fakultas Teknik dan Informatika, Universitas Multimedia
Nusantara

Penulis lahir di Jakarta tanggal 13 April 1983. Penulis adalah dosen pada Program Studi Informatika Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara. Menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika dan melanjutkan S2 pada bidang jurusan yang Teknik Informatika. Pada saat ini, penulis menekuni bidang Penelitian terkait komputer sains, sistem komputer dan teknologi informasi

BIODATA PENULIS

Andi Irfan, S.Kom., M.Kom.

Dosen Program Studi Teknik Informatika
Universitas Lamappapoleonro

Penulis lahir di Cangadi, 16 Agustus 1984. Penulis adalah dosen tetap pada Program Studi Teknik Informatika, Universitas Lamappapoleonro, Soppeng. Menyelesaikan pendidikan S1 pada program studi Teknik Informatika dan melanjutkan S2 pada program studi Sistem Komputer Universitas Handayani Makassar.

BIODATA PENULIS



Yudhistira A. Pratama, S.Kom, M.Kom, MTA, MTCNA
Dosen Program Studi Sistem Informasi
Fakultas Ilmu Komputer – Universitas Pelita Harapan

Lulus S1 di Program Studi Teknik Informatika STMIK Mikroskil Medan tahun 2013. Lulus S2 di Program Magister Teknik Informatika Universitas Sumatera Utara tahun 2019. Saat ini tercatat sebagai Dosen Program Studi Sistem Informasi Universitas Pelita Harapan Kampus Medan sejak tahun 2016, dengan beberapa mata kuliah yang diampu diantaranya : Pemrograman Aplikasi berbasis Web, Sistem Basis Data, Algoritma Pemrograman dan Jaringan Komputer. Selain itu juga berkarir secara professional sebagai praktisi pada bidang pengembangan aplikasi web dan pengelolaan infrastruktur bagi perusahaan-perusahaan swasta, kampus dan kantor pemerintahan yang ada di kota medan.

BIODATA PENULIS



Emha Diambang Ramadhany, S.Kom., M.Kom.

Dosen Program Studi Bisnis Digital

Fakultas Ekonomi dan Bisnis

Universitas dr. Soebandi

Penulis lahir di Banyuwangi tanggal 22 Februari 1993. Penulis adalah dosen tetap pada Program Studi Bisnis Digital Fakultas Ekonomi dan Bisnis, Universitas dr. Soebandi. Menyelesaikan pendidikan S1 dan S2 pada Jurusan Sistem Informasi dan Penulis menekuni bidang sistem informasi. Penulis memiliki ketertarikan topik penelitian mengenai pengembangan aplikasi, digitalisasi, dan analisis data. Saat ini penulis sedang merancang ekosistem bisnis digital di instansi tempat bekerja. Penulis membuka kerja sama pada aspek TriDharma dan lainnya untuk memajukan pendidikan tinggi di Indonesia.

BIODATA PENULIS



Suwito Pomalingo, S.Kom., M.Kom.

Dosen Program Studi Informatika
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis adalah dosen tetap pada Program Studi Informatika Fakultas Teknik dan Informatika Universitas Multimedia Nusantara. Menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika dan melanjutkan S2 pada Program Magister Informatika Universitas Islam Indonesia. Saat ini sementara menyelesaikan studi Doktorat dengan topik penelitian di bidang Security Science dengan pendekatan Deep Learning. Penulis menekuni bidang keilmuan meliputi Cyber Security, Malware Analytics, Data Science, Machine Learning, dan Deep Learning, selain itu, penulis juga memiliki beberapa sertifikat internasional di bidang Cyber Security, Digital Forensics, Data Science dan Machine Learning.

BIODATA PENULIS



Angga Aditya Permana
Dosen Informatika
Universitas Multimedia Nusantara

Angga Aditya Permana (lahir pada Desember 1989 di Jakarta) seorang dosen full time di bidang Computer Science pada Universitas Multimedia Nusantara sejak tahun 2021, fokus penelitian pada kriptografi, steganografi serta keamanan computer, namun pada tahun 2021 sedang mendalami topik bioinformatika dan network science. Angga juga memiliki hobi yaitu touring dan juga bermain bulutangkis, saat ini sedang menjadi mahasiswa program doctoral pada IPB University. Memulai karir pertama kali sebagai Network Enginner tahun 2011 dan memulai profesinya sebagai dosen pada 2013 di kampus swasta yang ada di Jakarta, pernah juga mengajar di kampus negeri yang berada di Jakarta dan Tangerang, juga pernah di undang menjadi dosen tamu untuk mengenalkan Bioinformatika di kampus negeri di Jakarta. Terimakasih..