

**LAPORAN AKHIR PRAKTIKUM
PEMROGRAMAN WEB II**



Oleh:

Nursyahna Puteri

NIM. 2210817120007

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
2024**

LEMBAR PENGESAHAN
LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN WEB II

Laporan Akhir Praktikum Pemrograman Web II

Modul 1: Variabel, Operator, dan Array

Modul 2: HTTP Request dan Kondisional

Modul 3: Perulangan dan kondisional

Modul 4: Array

Modul 5: Function dan Database

Modul 6: CRUD dan Login pada Codeigniter

ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Web II. Laporan Akhir Praktikum ini dikerjakan oleh:

Nama Praktikan : Nursyahna Puteri

NIM : 2210817120007

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

Ir. Eka Setya Wijaya, S.T., M.Kom.
NIP. 198205082008011010

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI.....	3
DAFTAR GAMBAR.....	10
DAFTAR TABEL	12
MODUL 1: VARIABEL, OPERATOR, DAN ARRAY	14
SOAL 1	14
A. Source Code	14
B. Output Program	15
C. Pembahasan	15
SOAL 2	17
A. Source Code	17
B. Output Program	19
C. Pembahasan	19
SOAL 3	21
A. Source Code	21
B. Output Program	22
C. Pembahasan	22
SOAL 4	24
A. Source Code	24
B. Output Program	26
C. Pembahasan	26
SOAL 5	32

A. Source Code	32
B. Output Program	34
C. Pembahasan	34
MODUL 2: HTTPS REQUEST DAN KONDISIONAL	40
SOAL 1	40
A. Source Code	40
B. Output Program	43
C. Pembahasan	43
SOAL 2	50
A. Source Code	51
B. Output Program	54
C. Pembahasan	54
SOAL 3	63
A. Source Code	63
B. Output Program	68
C. Pembahasan	69
SOAL 4	82
A. Source Code	82
B. Output Program	84
C. Pembahasan	84
MODUL 3: PERULANGAN DAN KONDISIONAL	89
SOAL 1	89
A. Source Code	89
B. Output Program	91

C. Pembahasan	92
SOAL 2	97
A. Source Code	98
B. Output Program	100
C. Pembahasan	100
SOAL 3	107
A. Source Code	107
B. Output Program	109
C. Pembahasan	109
SOAL 4	115
A. Source Code	116
B. Output Program	118
C. Pembahasan	119
SOAL 5	126
A. Source Code	127
B. Output Program	128
C. Pembahasan	129
MODUL 4: ARRAY.....	134
SOAL 1	134
A. Source Code	135
B. Output Program	138
C. Pembahasan	138
SOAL 2	147
A. Source Code	148

B. Output Program	151
C. Pembahasan	151
SOAL 3	158
A. Source Code	159
B. Output Program	164
C. Pembahasan	164
MODUL 5: FUNCTION DAN DATABASE	176
SOAL PRAKTIKUM	176
A. Source Code Koneksi.php	177
B. Pembahasan Koneksi.php.....	178
C. Source Code Model.php	180
D. Pembahasan Model.php.....	191
E. Source Code Index.php	216
F. Output Program Index.php	221
G. Pembahasan Index.php	221
H. Source Code Member.php	231
I. Output Program Member.php.....	239
J. Pembahasan Member.php.....	239
K. Source Code FormMember.php	255
L. Output Program FormMember.php	262
M. Pembahasan FormMember.php.....	262
N. Source Code Buku.php.....	277
O. Output Program Buku.php	284
P. Pembahasan Buku.php	284

Q. Source Code FormBuku.php	296
R. Output Program FormBuku.php	303
S. Pembahasan FormBuku.php	303
T. Source Code Peminjaman.php	317
U. Output Program Peminjaman.php	324
V. Pembahasan Peminjaman.php	325
W. Source Code FormPeminjaman.php	339
X. Output Program FormPeminjaman.php	347
Y. Pembahasan FormPeminjaman.php	347
MODUL 6: CRUD DAN LOGIN PADA CODEIGNITER	363
SOAL PRAKTIKUM	363
A. Source Code <i>Routes</i> .php	364
B. Pembahasan <i>Routes</i> .php	366
C. Source Code <i>BaseController</i> .php	368
D. Pembahasan <i>BaseController</i> .php	370
E. Source Code <i>BukuController</i> .php	372
F. Pembahasan <i>BukuController</i> .php	375
G. Source Code <i>Home</i> .php	379
H. Pembahasan <i>Home</i> .php	380
I. Source Code <i>RegisterController</i> .php	381
J. Pembahasan <i>RegsiterController</i> .php	383
K. Source <i>UserController</i> .php	386
L. Pembahasan <i>UserController</i> .php	388
M. Source Code <i>AuthFilter</i> .php	391

N.	Pembahasan <i>AuthFilter.php</i>	392
O.	Source Code <i>Create.php</i>	394
P.	Ouput Program <i>create.php</i>	400
Q.	Pembahasan <i>Create.php</i>	400
R.	Source Code <i>edit.php</i>	409
S.	Ouput Program <i>edit.php</i>	415
T.	Pembahasan <i>edit.php</i>	415
U.	Source Code <i>index.php</i>	423
V.	Output Program <i>index.php</i>	428
W.	Pembahasan <i>index.php</i>	428
X.	Source Code <i>login.php</i>	435
Y.	Output Program <i>login.php</i>	439
Z.	Pembahasan <i>login.php</i>	439
AA.	Source Code <i>register.php</i>	445
BB.	Ouput Program <i>register.php</i>	451
CC.	Pembahasan <i>register.php</i>	451
DD.	Source Code <i>welcome_mesaage.php</i>	458
EE.	Output Program <i>welcome_mesaage.php</i>	461
FF.	Pembahasan <i>welcome_mesaage.php</i>	461
GG.	Source Code <i>BukuModel.php</i>	465
HH.	Pembahasan <i>BukuModel.php</i>	466
II.	Source Code <i>UserModel.php</i>	469
JJ.	Pembahasan <i>UserModel.php</i>	471
KK.	Source Code <i>UserSeeder.php</i>	476

LL.	Pembahasan <i>UserSeeder.php</i>	477
MM.	Source Code 2024-06-12-202936_Buku	478
NN.	Pembahasan 2024-06-12-202936_Buku.....	480
OO.	Source Code 2024-06-12-202940_ <i>User</i>	482
PP.	Pembahasan 2024-06-12-202940_ <i>User</i>	484
QQ.	Source Code buku.sql	487
RR.	Pembahasan buku.sql.....	491
TAUTAN GITHUB.....		491

DAFTAR GAMBAR

MODUL 1: VARIABEL, OPERATOR, DAN ARRAY

Gambar 1. Screenshot Hasil Jawaban Modul 1 Soal 1	15
Gambar 2. Screenshot Hasil Jawaban Modul 1 Soal 2	19
Gambar 3. Screenshot Hasil Jawaban Modul 1 Soal 3	22
Gambar 4. Screenshot Hasil Jawaban Modul 1 Soal 4	26
Gambar 5. Screenshot Hasil Jawaban Modul 1 Soal 5	34

MODUL 2: HTTPS REQUEST DAN KONDISIONAL

Gambar 6. Screenshot Hasil Jawaban Modul 2 Soal 1	43
Gambar 8. Screenshot Hasil Jawaban Modul 2 Soal 2 (1).....	54
Gambar 9. Screenshot Hasil Jawaban Modul 2 Soal 2 (2).....	54
Gambar 10. Screenshot Hasil Jawaban Modul 2 Soal 3	68
Gambar 11. Screenshot Hasil Jawaban Modul 2 Soal 4	84

MODUL 3: PERULANGAN DAN KONDISIONAL

Gambar 12. Screenshot Hasil Jawaban Modul 3 Soal 1 (1).....	91
Gambar 13. Screenshot Hasil Jawaban Modul 3 Soal 1 (2).....	92
Gambar 14. Screenshot Hasil Jawaban Modul 3 Soal 2 (1).....	100
Gambar 15. Screenshot Hasil Jawaban Modul 3 Soal 2 (2).....	100
Gambar 16. Screenshot Hasil Jawaban Modul 3 Soal 3 (1).....	109
Gambar 17. Screenshot Hasil Jawaban Modul 3 Soal 3 (2).....	109
Gambar 18. Screenshot Hasil Jawaban Modul 3 Soal 4 (1).....	118
Gambar 19. Screenshot Hasil Jawaban Modul 3 Soal 4 (2).....	118
Gambar 20. Screenshot Hasil Jawaban Modul 3 Soal 4 (3).....	118
Gambar 21. Screenshot Hasil Jawaban Modul 3 Soal 5 (1).....	128
Gambar 22. Screenshot Hasil Jawaban Modul 3 Soal 5 (2).....	128

MODUL 4: ARRAY

Gambar 23. Screenshot Hasil Jawaban Modul 4 Soal 1	138
Gambar 24. Screenshot Hasil Jawaban Modul 4 Soal 2	151
Gambar 25. Screenshot Hasil Jawaban Modul 4 Soal 3	164

MODUL 5: FUNCTION DAN DATABASE

Gambar 26. Screenshot Hasil Jawaban Modul 5 Index.php	221
Gambar 27. Screenshot Hasil Jawaban Modul 5 Member.php	239
Gambar 28. Screenshot Hasil Jawaban Modul 5 FormMember.php	262
Gambar 29. Screenshot Hasil Jawaban Modul 5 Buku.php.....	284
Gambar 30. Screenshot Hasil Jawaban Modul 5 FormBuku.php	303
Gambar 31. Screenshot Hasil Jawaban Modul 5 Peminjaman.php	324
Gambar 32. Screenshot Hasil Jawaban Modul 5 FormPeminjaman.php.....	347

MODUL 6: CRUD DAN LOGIN PADA CODEIGNITER

Gambar 33. Screenshot Hasil Jawaban Modul 6 create.php	400
Gambar 34. Screenshot Hasil Jawaban Modul 6 edit.php	415
Gambar 35. Screenshot Hasil Jawaban Modul 6 index.php.....	428
Gambar 36. Screenshot Hasil Jawaban Modul 6 login.php	439
Gambar 37. Screenshot Hasil Jawaban Modul 6 register.php	451
Gambar 38. Screenshot Hasil Jawaban Modul 6 welcome_message.php	461

DAFTAR TABEL

MODUL 1: VARIABEL, OPERATOR, DAN ARRAY

Table 1. Source Code Modul 1 Soal 1	14
Table 2. Source Code Modul 1 Soal 2	17
Table 3. Source Code Modul 1 Soal 3	21
Table 4. Source Code Modul 1 Soal 4	24
Table 5. Source Code Modul 1 Soal 5	32

MODUL 2: HTTPS REQUEST DAN KONDISIONAL

Table 6. Source Code Modul 2 Soal 1	40
Table 7. Source Code Modul 2 Soal 2	51
Table 8. Source Code Modul 2 Soal 3	63
Table 9. Source Code Modul 2 Soal 4	82

MODUL 3: PERULANGAN DAN KONDISIONAL

Table 10. Source Code Modul 3 Soal 1	89
Table 11. Source Code Modul 3 Soal 2	98
Table 12. Source Code Modul 3 Soal 3	107
Table 13. Source Code Modul 3 Soal 4	116
Table 14. Source Code Modul 3 Soal 5	127

MODUL 4: ARRAY

Table 15. Source Code Modul 4 Soal 1	135
Table 16. Source Code Modul 4 Soal 2	148
Table 17. Source Code Modul 4 Soal 3	159

MODUL 5: FUNCTION DAN DATABASE

Table 18. Source Code Modul 5 Koneksi.php	177
Table 19. Source Code Modul 5 Model.php	180
Table 20. Source Code Modul 5 Index.php	216
Table 21. Source Code Modul 5 Member.php	231
Table 22. Source Code Modul 5 FormMember.php	255

Table 23. Source Code Modul 5 Buku.php.....	277
Table 24. Source Code Modul 5 FormBuku.php	296
Table 25. Source Code Modul 5 Peminjaman.php	317
Table 26. Source Code Modul 5 FormPeminjaman.php.....	339

MODUL 6: CRUD DAN LOGIN PADA CODEIGNITER

Table 27. Source Code Modul 6 Routes.php	364
Table 28. Source Code Modul 6 BaseController.php	368
Table 29. Source Code Modul 6 BukuController.php	372
Table 30. Source Code Modul 6 Home.php.....	379
Table 31. Source Code Modul 6 RegisterController.php.....	381
Table 32. Source Code Modul 6 UserController.php	386
Table 33. Source Code Modul 6 AuthFilter.php.....	391
Table 34. Source Code Modul 6 Create.php	394
Table 35. Source Code Modul 6 edit.php	409
Table 36. Source Code Modul 6 index.php.....	423
Table 37. Source Code Modul 6 login.php	435
Table 38. Source Code Modul 6 register.php	445
Table 39. Source Code Modul 6 welcome_message.php	458
Table 40. Source Code Modul 6 BukuModel.php	465
Table 41. Source Code Modul 6 UserModel.php	469
Table 42. Source Code Modul 6 UserSeeder.php	476
Table 43. Source Code Modul 6 2024-06-12-202936_Buku.....	478
Table 44. Source Code Modul 6 2024-06-12-202940_User.....	482
Table 45. Source Code Modul 6 buku.sql.....	487

MODUL 1: VARIABEL, OPERATOR, DAN ARRAY

SOAL 1

Buatlah program yang dapat mencetak kalimat “Hello World in Php” Menggunakan Bahasa Pemrograman PHP.

Output
Hello World in PHP
Nama Praktikan : {diganti Nama Anda}
NIM : {diganti NIM anda}

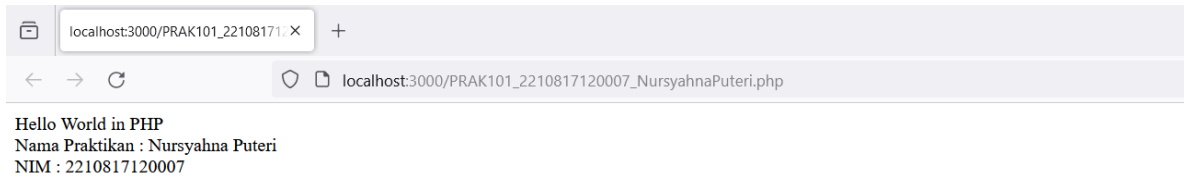
Simpan dengan nama file: PRAK101.php

A. Source Code

Table 1. Source Code Modul 1 Soal 1

1	<?php
2	
3	\$nama = "Nursyahna Puteri";
4	\$nim = "2210817120007";
5	
6	echo "Hello World in PHP Nama Praktikan : \$nama
	NIM : \$nim";
7	
8	?>

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Modul 1 Soal 1

C. Pembahasan

Pada baris [1], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [3], syntax `$nama = "Nursyahna Puteri";` pada baris ini terdapat sebuah variabel yang bernama 'nama', kegunaannya pada baris ini adalah sebagai tempat deklarasi dan inisialisasi dengan nilai string berupa 'Nursyahna Puteri'. Pendeklarasian tersebut ditandai dengan dimulainya simbol '\$' dimana menandakan bahwa nilai yang disimpan pada baris tersebut merupakan sebuah variabel yang akan digunakan untuk menyimpan nilai tertentu.

Pada baris [3], syntax `$nim = "2210817120007";` pada baris ini terdapat sebuah variabel yang bernama 'nim', kegunaannya pada baris ini adalah sebagai tempat deklarasi dan inisialisasi dengan nilai string berupa '2210817120007'. Pendeklarasian tersebut ditandai dengan dimulainya simbol '\$' dimana menandakan bahwa nilai yang disimpan pada baris tersebut merupakan sebuah variabel yang akan digunakan untuk menyimpan nilai tertentu.

Pada baris [6], syntax `echo "Hello World in PHP
 Nama Praktikan : $nama
 NIM : $nim";` penggunaan baris ini digunakan untuk menampilkan teks atau nilai ke layar. Pada baris ini terdapat 'echo' yang merupakan perintah dari PHP untuk dapat menampilkan teks atau nilai yang diinginkan tersebut

sebagai output, pada baris ini dengan menggunakan 'echo' teks yang akan dihasilkan berupa tiga baris teks yaitu 'Hello World in PHP', 'Nursyahna Puteri' yang disimpan dalam variabel '\$nama', serta '2210817120007' yang disimpan dalam variabel 'nim'. Baris-baris tersebut dipisahkan dengan tag '
' yang merupakan tag HTML, penggunaan tag ini adalah untuk membuat baris baru di dalam tampilan web. Sehingga ketiga teks tersebut akan ditampilkan diikuti dengan baris baru.

Pada baris [8], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

SOAL 2

Buatlah program dengan Bahasa pemrograman PHP yang dapat menghitung volume bangun ruang dan menampilkan hasilnya dengan 3 desimal dibelakang koma. Parameter awal yang diperlukan dideklarasikan secara statis. Bangun ruang yang digunakan disesuaikan dengan ketentuan berikut:

- NIM dengan akhiran 0/1 : Bangun ruang tabung
- NIM dengan akhiran 2/3 : Bangun ruang kerucut
- NIM dengan akhiran 4/5 : Bangun ruang bola
- NIM dengan akhiran 6/7 : Bangun ruang prisma alas segitiga
- NIM dengan akhiran 8/9 : Bangun ruang limas alas persegi Panjang

Gunakan nilai berikut yang diperlukan sesuai bangun ruang	Output
Jari-jari = 4.2 Tinggi = 5.4 Panjang = 8.9 Lebar = 14.7 Sisi = 7.9	? m3

Simpan dengan nama file: PRAK102.php

A. Source Code

Table 2. Source Code Modul 1 Soal 2

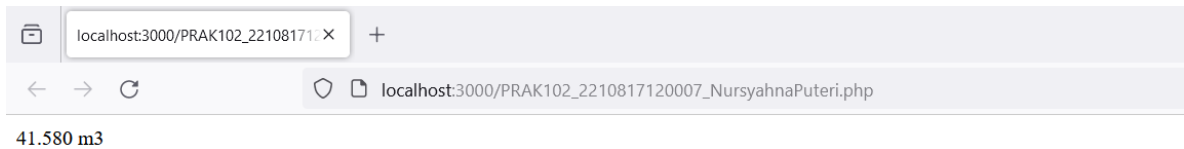
1	<?php
2	
3	\$jari = 4.2;
4	\$tinggi = 5.4;
5	\$panjang = 8.9;
6	\$lebar = 14.7;
7	\$sisi = 7.9;

```

8  $alasprisma = 4;
9  $tinggiprisma = 8;
10 $tinggilimas = 10;
11 $phi = 3.14;
12
13 $tabung = $phi * $jari * $jari * $tinggi;
14 $kerucut = 1/3 * $phi * $jari * $jari * $tinggi;
15 $bola = 4/3 * $phi * $jari * $jari * $jari;
16 $prismaalassegitiga = 1/2 * $alasprisma *
    $tinggiprisma;
17 $limasalaspersegipanjang = 1/3 * $panjang * $lebar *
    $tinggilimas;
18
19 // NIM : 2210817120007
20 $nim = 7;
21
22 if( $nim == 0 || $nim == 1 ) {
23     echo number_format($tabung, 3)." m3";
24 } else if ( $nim == 2 || $nim == 3 ) {
25     echo number_format($kerucut, 3)." m3";
26 } else if ( $nim == 4 || $nim == 5 ) {
27     echo number_format($bola, 3)." m3";
28 } else if ( $nim == 6 || $nim == 7 ) {
29     echo number_format($prismaalassegitiga, 3)." m3";
30 } else if ( $nim == 8 || $nim == 9 ) {
31     echo number_format($limasalaspersegipanjang, 3)."
    m3";
32 }
33
34 ?>

```

B. Output Program



Gambar 2. Screenshot Hasil Jawaban Modul 1 Soal 2

C. Pembahasan

Pada baris [1], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [5] [6] [7], syntax

```
$alassegitiga = 3.5;
```

```
$tinggisegitiga = 4.4;
```

```
$tinggiprisma = 5.4;
```

baris-baris tersebut merupakan deklarasi variabel. Pada baris-baris tersebut variabel-variabel yang diperlukan untuk perhitungan rumus-rumus geometri dideklarasikan. Pada variabel-variabel tersebut diawalnya terdapat atau digunakan simbol ‘\$’ dimana menandakan bahwa nilai yang disimpan pada baris tersebut merupakan sebuah variabel yang akan digunakan untuk menyimpan nilai tertentu. Setiap variabel mewakili alas segitiga, tinggi segitiga, serta tinggi prisma untuk menghitung volume dari prisma alas segitiga. Setiap variabel memiliki nilai yang sudah disesuaikan dari bangun ruang yang akan dihitung. Pada variabel-variabel tersebut.

Pada baris [9], syntax `$volumeprismaalassegitiga = ($alassegitiga * $tinggisegitiga / 2) * $tinggiprisma;` baris-baris tersebut

merupakan perhitungan rumus bangun ruang menggunakan variabel-variabel yang telah dideklarasikan sebelumnya. Pada variabel-variabel tersebut diawalnya juga terdapat atau digunakan simbol ‘\$’ dimana menandakan bahwa nilai yang disimpan pada baris tersebut merupakan sebuah variabel yang akan digunakan untuk menyimpan nilai tertentu.

Pada baris [11], syntax `echo number_format($volumeprismaalassegitiga, 3)." m3";` pada baris tersebut dengan menggunakan perintah ‘echo’ maka hasil dari perhitungan volume prisma akan ditampilkan. Hasil perhitungan tersebut ditampilkan dengan menggunakan fungsi ‘number_format()’ untuk dapat memformat hasil dengan tiga desimal di belakang koma, dan disertai juga dengan satuan volume ‘m3’.

Pada baris [13], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

SOAL 3

Buatlah program dalam Bahasa pemrograman php yang dapat mengkonversi nilai suhu celcius menjadi Reamur, Fahrenheit dan Kelvin. Tampilkan hasilnya dengan format 4 desimal dibelakang koma. Nilai input dideklarasikan secara statis.

Input	Output
Celcius = 37.841	Fahrenheit (F) = 100,1138 Reamur (R) = 30,2728 Kelvin (K) = 310,991

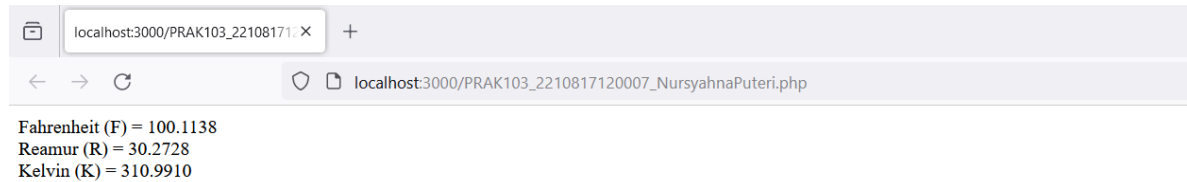
Simpan dengan nama file: PRAK103.php

A. Source Code

Table 3. Source Code Modul 1 Soal 3

1	<?php
2	
3	\$celcius = 37.841;
4	
5	\$fahrenheit = (9/5) * \$celcius + 32;
6	\$reamur = (4/5) * \$celcius;
7	\$kelvin = \$celcius + 273.15;
8	
9	echo "Fahrenheit (F) = ".number_format(\$fahrenheit, 4). " ";
10	echo "Reamur (R) = ".number_format(\$reamur, 4). " ";
11	echo "Kelvin (K) = ".number_format(\$kelvin, 4);
12	
13	?>

B. Output Program



Gambar 3. Screenshot Hasil Jawaban Modul 1 Soal 3

C. Pembahasan

Pada baris [1], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [3], syntax `$celcius = 37.841;` pada baris ini merupakan pendeklarasian variabel '`$celcius`' dengan memberikan nilai 37.841 kedalamnya. Variabel ini akan digunakan untuk menyimpan suhu dalam satuan Celsius yang akan dikonversi ke dalam Fahrenheit, Reamur, dan Kelvin.

Pada baris [5] [6] [7], syntax

```
$fahrenheit = (9/5) * $celcius + 32;
```

```
$reamur = (4/5) * $celcius;
```

```
$kelvin = $celcius + 273.15;
```

pada baris-baris tersebut mendeklarasikan rumus-rumus untuk mengkonversi suhu dari Celcius ke Fahrenheit, Reamur, dan Kelvin. Nilai-nilai dari hasil perhitungan tersebut akan disimpan dalam variabel '`$fahrenheit`', '`$reamur`', dan '`$kelvin`'.

Pada baris [9] [10] [11], syntax

```
echo "Fahrenheit (F) = ".number_format($fahrenheit,  
4). "<br>";
```

```
echo "Reamur (R) = ".number_format($reamur, 4). "<br>";
```

```
echo "Kelvin (K) = ".number_format($kelvin, 4);
```

pada baris-baris tersebut dipergunakan untuk menampilkan hasil perhitungan konversi suhu ke layar dengan menggunakan perintah 'echo'. Perintah 'echo' merupakan perintah dari PHP untuk dapat menampilkan teks atau nilai yang diinginkan tersebut sebagai output. Selanjutnya pada baris-baris tersebut juga menggunakan fungsi 'number_format()' yang digunakan untuk memformat angka sehingga memiliki 4 desimal di belakang koma. Hasil perhitungan untuk Fahrenheit, Reamur, dan Kelvin akan ditampilkan secara berurutan dengan disertai unit masing-masing suhu dalam kurung kurawal. Selain itu pada baris-baris tersebut juga terdapat tag '
' yang merupakan tag HTML, penggunaan tag ini adalah untuk membuat baris baru di dalam tampilan web.

Pada baris [13], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

SOAL 4

Buatlah program dalam Bahasa pemrograman PHP yang menampilkan daftar data seperti ditunjukkan pada Gambar 1. Gunakan indexed array dalam implementasinya

Daftar Smartphone Samsung
Samsung Galaxy S22
Samsung Galaxy S22+
Samsung Galaxy A03
Samsung Galaxy Xcover 5

serta Gunakan kombinasi HTML, CSS dan PHP.

Simpan dengan nama file: PRAK104.php

A. Source Code

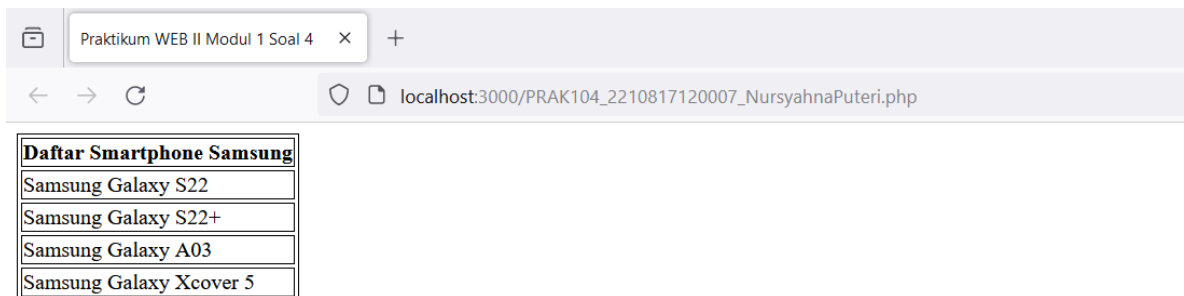
Table 4. Source Code Modul 1 Soal 4

1	<?php
2	
3	\$daftarsmartphonesamsung = ["Samsung Galaxy S22", "Samsung Galaxy S22+", "Samsung Galaxy A03", "Samsung Galaxy Xcover 5"];
4	
5	?>
6	
7	<!DOCTYPE html>
8	<html lang="en">
9	<head>
10	<meta charset="UTF-8">
11	<meta http-equiv="X-UA-Compatible" content="IE=edge">
12	

	<code><meta name="viewport" content="width=device-width,</code>
13	<code>initial-scale=1.0"></code>
14	<code><title>Praktikum WEB II Modul 1 Soal 4</title></code>
15	<code><style></code>
16	<code>table, th, td {</code>
17	<code>border: 1px solid;</code>
18	<code>}</code>
19	<code></style></code>
20	<code></head></code>
21	
22	<code><body></code>
23	<code><table></code>
24	<code><tr></code>
25	<code><th>Daftar Smartphone Samsung</th></code>
26	<code></tr></code>
27	<code><tr></code>
28	<code><td><?php echo \$daftarsmartphonesamsung[0];</code>
	<code>?></td></code>
29	<code></tr></code>
30	<code><tr></code>
31	<code><td><?php echo \$daftarsmartphonesamsung[1];</code>
	<code>?></td></code>
32	<code></tr></code>
33	<code><tr></code>
34	<code><td><?php echo \$daftarsmartphonesamsung[2];</code>
	<code>?></td></code>
35	<code></tr></code>
36	<code><tr></code>
37	<code><td><?php echo \$daftarsmartphonesamsung[3];</code>
	<code>?></td></code>

38	<code></tr></code>
39	<code></table></code>
40	<code></body></code>
	<code></html></code>

B. Output Program



Gambar 4. Screenshot Hasil Jawaban Modul 1 Soal 4

C. Pembahasan

Pada baris [1], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [3], syntax `$daftarsmartphonesamsung = ["Samsung Galaxy S22", "Samsung Galaxy S22+", "Samsung Galaxy A03", "Samsung Galaxy Xcover 5"];` pada baris ini mendeklarasikan sebuah array dengan nama `'$daftarsmartphonesamsung'`. Array ini merupakan indexed array, yaitu jenis array di dalam pemrograman yang menyimpan sejumlah nilai atau elemen yang terurut secara berurutan, dimulai dari indeks 0 dan seterusnya. Dalam penggunaan indexed array, setiap elemen memiliki indeks numerik yang berfungsi sebagai alamat dalam array. Pada baris ini array bersikan daftar smartphone Samsung dengan empat elemen yaitu "Samsung Galaxy S22", "Samsung Galaxy S22+", "Samsung Galaxy A03", dan "Samsung Galaxy Xcover 5".

Pada baris [5], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [7], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [8], `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa English.

Pada baris [9], `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML.

Pada baris [10], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [11], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [12], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [13], syntax `<title>raktikum WEB II Modul 1 Soal 4</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [14], syntax `<style>` baris ini merupakan internal CSS, digunakan untuk menyisipkan kode style atau CSS ke dalam sebuah dokumen HTML. Untuk penempatan tag ini berada di dalam tag `<head>` dan tag penutupnya yaitu `</head>`.

Pada baris [15] [16] [17], syntax

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

pada baris-baris tersebut adalah bagian dari kode CSS yang terletak di dalam tag `<style>`. Baris-baris tersebut mendefinisikan gaya untuk elemen `<table>`, `<th>`, dan `<td>`, di mana setiap elemen tersebut akan memiliki garis tepi dengan ketebalan 1 piksel.

Pada baris [18], syntax `</style>` merupakan tag penutup dari baris 14.

Pada baris [19], syntax `</head>` merupakan tag penutup HTML dari baris 9.

Pada baris [21], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [22], syntax `<table>` baris ini terdapat tag tabel pembuka, digunakan untuk membungkus tabel yang akan dibuat.

Pada baris [23] [24] [25], syntax

```
<tr>  
    <th>Daftar Smartphone Samsung</th>  
</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Pada baris ini terdapat juga teks "Daftar Smartphone Samsung" yang terletak di dalam elemen <th>, dimana teks tersebut akan ditampilkan di bagian atas tabel, yang berfungsi sebagai judul kolom atau baris yang berkaitan dengan daftar smartphone Samsung.

Pada baris [26] [27] [28], syntax

```
<tr>

    <td><?php echo $daftarsmartphonesamsung[0];
?></td>

</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Di dalam sel tabel tersebut, terdapat juga kode PHP yang berfungsi untuk menampilkan nilai dari array \$daftarsmartphonesamsung yang telah dibuat pada indeks 0. Dengan adanya pengkodean tersebut, maka nilai dari elemen pertama dalam array tersebut akan ditampilkan di dalam sel tabel yang dibuat tersebut. Penggunaan kode tersebut akan menampilkan satu nilai dari array untuk setiap barisnya, sehingga akan membentuk daftar smartphone Samsung yang sesuai dengan urutan ideks dalam array.

Pada baris [29] [30] [31], syntax

```
<tr>

    <td><?php echo $daftarsmartphonesamsung[1];
?></td>

</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Di dalam sel tabel tersebut, terdapat juga kode PHP yang berfungsi untuk menampilkan nilai dari array \$daftarsmartphonesamsung yang telah dibuat pada indeks 1. Dengan adanya pengkodean tersebut, maka nilai dari elemen pertama dalam array tersebut akan ditampilkan di dalam sel tabel yang sudah dibuat tersebut. Penggunaan kode tersebut akan menampilkan satu nilai dari array untuk setiap barisnya, sehingga akan membentuk daftar smartphone Samsung yang sesuai dengan urutan ideks dalam array.

Pada baris [32] [33] [34], syntax

```
<tr>
    <td><?php    echo    $daftarsmartphonesamsung[2];
?></td>
</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Di dalam sel tabel tersebut, terdapat juga kode PHP yang berfungsi untuk menampilkan nilai dari array \$daftarsmartphonesamsung yang telah dibuat pada indeks 2. Dengan adanya pengkodean tersebut, maka nilai dari elemen pertama dalam array tersebut akan ditampilkan di dalam sel tabel yang sudah dibuat tersebut. Penggunaan kode tersebut akan menampilkan satu nilai dari array untuk setiap barisnya, sehingga akan membentuk daftar smartphone Samsung yang sesuai dengan urutan ideks dalam array.

Pada baris [35] [36] [37], syntax

```
<tr>
```

```

        <td><?php echo $daftarsmartphonesamsung[3];
?></td>

    </tr>

```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Di dalam sel tabel tersebut, terdapat juga kode PHP yang berfungsi untuk menampilkan nilai dari array \$daftarsmartphonesamsung yang telah dibuat pada indeks 3. Dengan adanya pengkodean tersebut, maka nilai dari elemen pertama dalam array tersebut akan ditampilkan di dalam sel tabel yang sudah dibuat tersebut. Penggunaan kode tersebut akan menampilkan satu nilai dari array untuk setiap barisnya, sehingga akan membentuk daftar smartphone Samsung yang sesuai dengan urutan ideks dalam array.

Pada baris [38], syntax </table> pada baris ini adalah tag penutup dari tag pembuatan table.

Pada baris [39], </body> pada baris ini merupakan tag penutup dari tag pembuka baris 21.

Pada baris [40], </html> baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 5

Buatlah program dalam Bahasa pemrograman PHP yang menampilkan daftar data seperti ditunjukkan pada Gambar 1. Gunakan associative array dalam implementasinya

Daftar Smartphone Samsung
Samsung Galaxy S22
Samsung Galaxy S22+
Samsung Galaxy A03
Samsung Galaxy Xcover 5

serta Gunakan kombinasi HTML, CSS dan PHP.

Simpan dengan nama file: PRAK105.php

A. Source Code

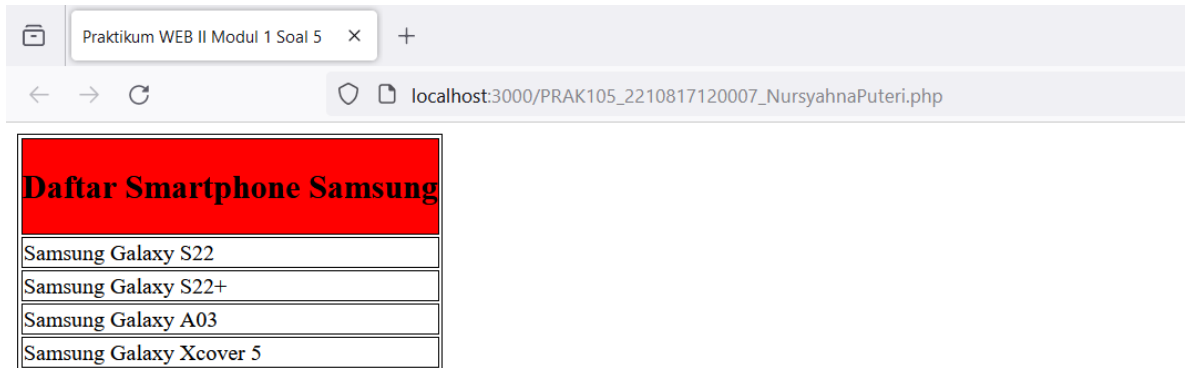
Table 5. Source Code Modul 1 Soal 5

1	<?php
2	
3	\$daftarsmartphonesamsung = ["S22" => "Samsung Galaxy S22", "S22+" => "Samsung Galaxy S22+", "A03" => "Samsung Galaxy A03", "Xcover5" => "Samsung Galaxy Xcover 5"];
4	
5	?>
6	
7	<!DOCTYPE html>
8	<html lang="en">
9	<head>
10	<meta charset="UTF-8">
11	<meta http-equiv="X-UA-Compatible" content="IE=edge">
12	<meta name="viewport" content="width=device-width, initial-scale=1.0">

	<title>Praktikum WEB II Modul 1 Soal 5</title>
13	<style>
14	table, th, td {
15	border: 1px solid;
16	}
17	th {
18	font-size: x-large;
19	background-color: red;
20	padding: 20px 0px;
21	}
22	</style>
23	</head>
24	
25	<body>
26	<table>
27	<tr>
28	<th>Daftar Smartphone Samsung</th>
29	</tr>
30	<tr>
31	<td><?php echo
32	\$daftarsmartphonesamsung["S22"]; ?></td>
	</tr>
33	<tr>
34	<td><?php echo
35	\$daftarsmartphonesamsung["S22+"]; ?></td>
	</tr>
36	<tr>
37	<td><?php echo
38	\$daftarsmartphonesamsung["A03"]; ?></td>
	</tr>

39	<tr>
40	<td><?php echo
41	\$daftarsmartphonesamsung["Xcover5"]; ?></td>
	</tr>
42	</table>
43	</body>
44	</html>
45	

B. Output Program



Gambar 5. Screenshot Hasil Jawaban Modul 1 Soal 5

C. Pembahasan

Pada baris [1], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [3], syntax `$daftarsmartphonesamsung = ["S22" => "Samsung Galaxy S22", "S22+" => "Samsung Galaxy S22+", "A03" => "Samsung Galaxy A03", "Xcover5" => "Samsung Galaxy Xcover 5"];` pada baris ini merupakan pendefinisian sebuah associative array dengan nama `'$daftarsmartphonesamsung'`. Array associative pada PHP merupakan jenis array di mana setiap elemen memiliki sebuah key atau index yang ditentukan.

Pada baris [5], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [7], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisikan informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [8], `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa English.

Pada baris [9], `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML.

Pada baris [10], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [11], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [12], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [13], syntax `<title>raktikum WEB II Modul 1 Soal 5</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [14], syntax `<style>` baris ini merupakan internal CSS, digunakan untuk menyisipkan kode style atau CSS ke dalam sebuah dokumen HTML. Untuk penempatan tag ini berada di dalam tag `<head>` dan tag penutupnya yaitu `</head>`.

Pada baris [15] [16] [17], syntax

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

pada baris-baris tersebut adalah bagian dari kode CSS yang terletak di dalam tag `<style>`. Baris-baris tersebut mendefinisikan gaya untuk elemen `<table>`, `<th>`, dan `<td>`, di mana setiap elemen tersebut akan memiliki garis tepi dengan ketebalan 1 piksel.

Pada baris [18] [19] [20] [21] [22] [23] [24], syntax

```
th {  
  
    font-size: x-large;  
  
    background-color: red;  
  
    padding: 20px 0px;  
  
}
```

pada baris-baris tersebut adalah bagian dari kode CSS yang terletak di dalam tag `<style>`. Baris-baris tersebut dipergunakan untuk mengatur sel header yaitu ditandai dengan adanya tag `<th>`. Pada baris tersebut yang akan diatur adalah ukuran font teks pada sel header menjadi besar dengan menggunakan `'x-large'`, lalu terdapat juga penggunaan `'background-color: red;'` untuk menetapkan latar belakang sel header

menjadi warna merah, dan selanjutnya terdapat penggunaan ‘padding: 20px 0px;’ yang berguna untuk menambahkan ruang padding di sekitar isi pada sel header.

Pada baris [23], syntax `</style>` merupakan tag penutup dari baris 14.

Pada baris [24], syntax `</head>` merupakan tag penutup HTML dari baris 9.

Pada baris [26], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [27], syntax `<table>` baris ini terdapat tag tabel pembuka, digunakan untuk membungkus tabel yang akan dibuat.

Pada baris [28] [29] [30], syntax

```
<tr>
    <th>Daftar Smartphone Samsung</th>
</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Pada baris ini terdapat juga teks "Daftar Smartphone Samsung" yang terletak di dalam elemen <th>, dimana teks tersebut akan ditampilkan di bagian atas tabel, yang berfungsi sebagai judul kolom atau baris yang berkaitan dengan daftar smartphone Samsung.

Pada baris [31] [32] [33], syntax

```
<tr>
    <td><?php echo
$daftarsmartphonesamsung["S22"]; ?></td>
</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah

tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Pada baris tersebut juga disertakan kode php berupa ‘><?php echo \$daftarsmartphonesamsung["S22"]; ?>’ dimana kegunaanlah untuk mengakses nilai dari associative array ‘\$daftarsmartphonesamsung’ dengan menggunakan kunci ‘S22’. Sehingga dengan demikian php akan menemukan nilai yang terkait dengan kunci tersebut dalam array yang telah dibuat, yaitu ‘Samsung Galaxy S22’.

Pada baris [34] [35] [36], syntax

```
<tr>

        <td><?php echo
$daftarsmartphonesamsung["S22+"]; ?></td>

</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Pada baris tersebut juga disertakan kode php berupa ‘><?php echo \$daftarsmartphonesamsung["S22+"]; ?>’ dimana kegunaanlah untuk mengakses nilai dari associative array ‘\$daftarsmartphonesamsung’ dengan menggunakan kunci ‘S22+’. Sehingga dengan demikian php akan menemukan nilai yang terkait dengan kunci tersebut dalam array yang telah dibuat, yaitu ‘Samsung Galaxy S22+’.

Pada baris [37] [38] [39], syntax

```
<tr>

        <td><?php echo
$daftarsmartphonesamsung["A03"]; ?></td>

</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah

tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Pada baris tersebut juga disertakan kode php berupa ‘<>?php echo \$daftarsmartphonesamsung["A03"]; ?>’ dimana kegunaanlah untuk mengakses nilai dari associative array ‘\$daftarsmartphonesamsung’ dengan menggunakan kunci ‘A03’. Sehingga dengan demikian php akan menemukan nilai yang terkait dengan kunci tersebut dalam array yang telah dibuat, yaitu ‘Samsung Galaxy A03’.

Pada baris [40] [41] [42], syntax

```
<tr>

        <td><?php echo
$daftarsmartphonesamsung["Xcover5"]; ?></td>

</tr>
```

pada baris-baris tersebut merupakan struktur table html yang akan dibuat. Penggunaan tag ‘<tr>’ yang merupakan table row digunakan untuk membuat baris dalam sebuah tabel. Selanjutnya untuk tag ‘<th>’ yang merupakan table head dipergunakan untuk membuat judul pada header table. Pada baris tersebut juga disertakan kode php berupa ‘<>?php echo \$daftarsmartphonesamsung["Xcover5"]; ?>’ dimana kegunaanlah untuk mengakses nilai dari associative array ‘\$daftarsmartphonesamsung’ dengan menggunakan kunci ‘Xcover5’. Sehingga dengan demikian php akan menemukan nilai yang terkait dengan kunci tersebut dalam array yang telah dibuat, yaitu ‘Samsung Galaxy Xcover5’.

Pada baris [43], syntax </table> pada baris ini adalah tag penutup dari tag pembuatan table.

Pada baris [44], </body> pada baris ini merupakan tag penutup dari tag pembuka baris 26.

Pada baris [45], </html> baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

MODUL 2:

HTTPS REQUEST DAN KONDISIONAL

SOAL 1

Buatlah sebuah program untuk mengurutkan 3 nama inputan dari urutan huruf abjad terkecil ke terbesar menggunakan metode kondisional. Input merupakan nama secara acak dengan output adalah hasil dari pengurutan.

Hasil yang diinginkan:

Nama: 1	<input type="text" value="andry"/>
Nama: 2	<input type="text" value="andre"/>
Nama: 3	<input type="text" value="andra"/>
<input type="button" value="Urutkan"/>	

Output
andra andre andry

Simpan dengan nama file: PRAK201.php

A. Source Code

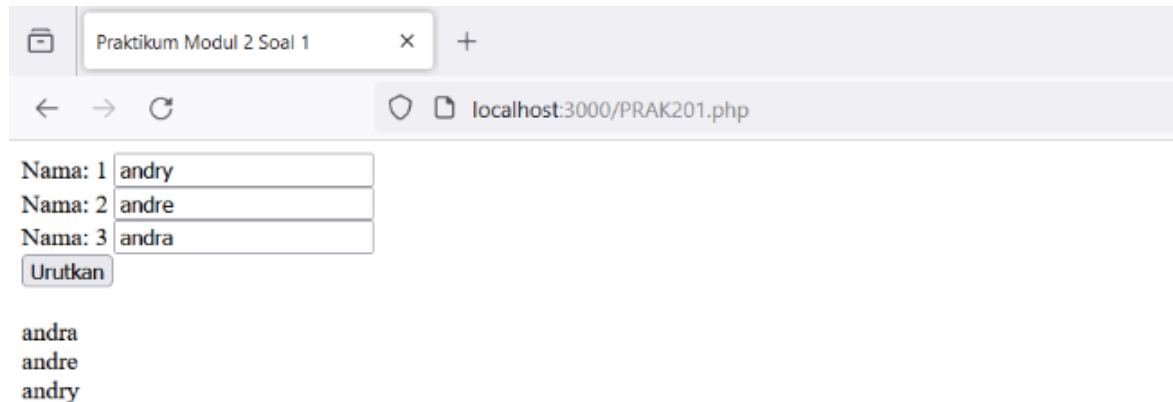
Table 6. Source Code Modul 2 Soal 1

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	<code> <meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>
6	<code> <meta name="viewport" content="width=device-width,</code>
	<code>initial-scale=1.0"></code>
7	<code> <title>Praktikum Modul 2 Soal 1</title></code>

8	</head>
9	
10	<body>
11	<form method="POST">
12	Nama: 1 <input type="text" name="masukannama1" value="<?php if(isset(\$_POST['masukannama1'])) { echo \$_POST['masukannama1']; } ?>"></br>
13	Nama: 2 <input type="text" name="masukannama2" value="<?php if(isset(\$_POST['masukannama2'])) { echo \$_POST['masukannama2']; } ?>"></br>
14	Nama: 3 <input type="text" name="masukannama3" value="<?php if(isset(\$_POST['masukannama3'])) { echo \$_POST['masukannama3']; } ?>"></br>
15	<input ><="" name="submit" td="" type="submit" value="Urutkan"/>
16	</form></br>
17	
18	<?php
19	if (isset(\$_POST['submit'])) {
20	\$masukannama1 = \$_POST['masukannama1'];
21	\$masukannama2 = \$_POST['masukannama2'];
22	\$masukannama3 = \$_POST['masukannama3'];
23	
24	if (\$masukannama1 < \$masukannama2 && \$masukannama1 < \$masukannama3) {
25	if (\$masukannama2 < \$masukannama3) {
26	echo "\$masukannama1 \$masukannama2 \$masukannama3";
27	} else {

28	echo "\$masukannama1 \$masukannama3 \$masukannama2";
29	}
30	} elseif (\$masukannama2 < \$masukannama1 && \$masukannama2 < \$masukannama3) {
31	if (\$masukannama1 < \$masukannama3) {
32	echo "\$masukannama2 \$masukannama1 \$masukannama3";
33	} else {
34	echo "\$masukannama2 \$masukannama3 \$masukannama1";
35	}
36	} else {
37	if (\$masukannama1 < \$masukannama2) {
38	echo "\$masukannama3 \$masukannama1 \$masukannama3";
39	} else {
40	echo "\$masukannama3 \$masukannama2 \$masukannama1";
41	}
42	}
43	}
44	?>
45	</body>
46	</html>

B. Output Program



The screenshot shows a web browser window with the title "Praktikum Modul 2 Soal 1". The address bar shows "localhost:3000/PRAK201.php". The form contains three input fields labeled "Nama: 1", "Nama: 2", and "Nama: 3" with values "andry", "andre", and "andra" respectively. Below the inputs is a button labeled "Urutkan". The output below the button shows the names sorted alphabetically: "andra", "andre", and "andry".

Gambar 6. Screenshot Hasil Jawaban Modul 2 Soal 1

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [3], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [4], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk

menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [5], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [6], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [7], syntax `<title>Praktikum Modul 2 Soal 1</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [8], syntax `</head>` merupakan tag penutup HTML dari baris 3.

Pada baris [10], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [11], syntax `<form method="POST">` merupakan tag form yang digunakan untuk membuat formulir di dalam halaman web. Pada baris ini menggunakan atribut berupa `method="POST"` dimana metode tersebut digunakan untuk pengiriman data formulir ke server untuk diproses.

Pada baris [12] [13] [14], syntax

```
Nama: 1 <input type="text" name="masukannama1"
value="<?php if(isset($_POST['masukannama1'])) { echo
$_POST['masukannama1']; } ?>"></br>
```

```
Nama: 2 <input type="text" name="masukannama2"
value="<?php if(isset($_POST['masukannama2'])) { echo
$_POST['masukannama2']; } ?>"></br>
```

```
Nama: 3 <input type="text" name="masukannama3"
value="<?php if(isset($_POST['masukannama3'])) { echo
$_POST['masukannama3']; } ?>"></br>
```

penggunaan pada baris-baris tersebut merupakan bagian dari formulir di mana pengguna diminta untuk memasukkan nama pertama, nama kedua, dan nama ketiga. Keterangan Nama: 1, 2 atau 3 tersebut merupakan label yang nantinya akan ditampilkan di halaman web, lalu terdapat juga penggunaan `type="text"` menyatakan bahwa inputan berupa teks. Lalu penggunaan `name="masukannama1"`; `name="masukannama2"`; dan `name="masukannama3"` untuk memberikan nama inputan, sehingga nantinya digunakan untuk mengakses nilai input di sisi server. Pada baris-baris tersebut juga terdapat penggunaan `<?php if(isset($_POST['masukannama1'])) { echo $_POST['masukannama1']; } ?>`: kode ini dipergunakan sebagai struktur kontrol PHP yang memeriksa apakah ada data yang dikirimkan melalui metode POST untuk input dengan nama `masukannama1`. Jika ada, nilai tersebut akan ditampilkan kembali ke dalam input teks. Selanjutnya pada baris-baris tersebut juga terdapat tag `</br>` atau line break, tag tersebut digunakan untuk memulai baris baru pada dokumen HTML.

Pada baris [15], syntax `input type="submit" name="submit" value="Urutkan">` penggunaannya adalah untuk mengirimkan data formulir ke server untuk diproses. Pada baris tersebut atribut `type="submit"` yang digunakan untuk menunjukkan bahwa elemen ini adalah tombol submit. Lalu atribut `name="submit"` memberikan nama untuk tombol submit tersebut. Nama ini akan digunakan untuk

mengidentifikasi tombol ini di sisi server ketika formulir dikirimkan. Selanjutnya terdapat juga atribut `value="Urutkan"` yang digunakan untuk menentukan teks yang akan ditampilkan pada tombol submit. Dalam penggunaannya, teks yang akan muncul pada tombol tau ditampilkan pada layer pengguna adalah "Urutkan".

Pada baris [16], syntax `</form></br>` penggunaan tag tersebut adalah sebagai tag penutup untuk elemen `<form>` yang berarti akhir formulir, Serta tag `
` atau break line yang memiliki break memiliki fungsi untuk memulai baris baru pada dokumen HTML.

Pada baris [18], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [19], syntax `if (isset($_POST['submit']))` {penggunaan baris ini adalah untuk kondisi yang akan memeriksa apakah tombol submit pada formulir telah ditekan. Penggunaan `isset($_POST['submit'])` akan memeriksa apakah ada data yang dikirimkan melalui metode POST dengan nama kunci 'submit'. Sehingga pada baris ini ada pengkondisian untuk melakukan pemeriksaan apakah formulir yang diisikan telah disubmit.

Pada baris [20] [21] [22], syntax

```
$masukannama1 = $_POST['masukannama1'];  
  
$masukannama2 = $_POST['masukannama2'];  
  
$masukannama3 = $_POST['masukannama3'];
```

penggunaan pada baris-baris tersebut merupakan proses dari pengambilan nilai dari formulir yang disubmit menggunakan metode POST. Setiap baris seperti `$masukannama1 = $_POST['masukannama1'];` memiliki tugas untuk mengambil nilai dari input yang sesuai dalam formulir, selanjutnya penggunaan `$_POST['masukannama1'];` adalah cara untuk mengakses nilai yang dikirim melalui formulir dengan atribut 'name' yang sesuai, yaitu 'masukannama1'. Apabila nilai-nilai

sudah diambil, maka nilai-nilai tersebut akan disimpan dalam variabel \$masukannama1, \$masukannama2, dan \$masukannama3.

Pada baris [24], syntax `if ($masukannama1 < $masukannama2 && $masukannama1 < $masukannama3) {` baris ini merupakan blok if utama. Pada kondisioanl pertama ini akan memeriksa apakah \$masukannama1 lebih kecil dari kedua \$masukannama2 dan \$masukannama3. Jika ini benar, maka blok kode di dalamnya akan dieksekusi. Jika kondisi ini terpenuhi, artinya \$masukannama1 adalah yang terkecil di antara ketiga variabel tersebut

Pada baris [25] [26] [27] [28] [29], syntax

```
    if ($masukannama2 < $masukannama3) {  
        echo    "$masukannama1    <br>    $masukannama2    <br>  
$masukannama3";  
    } else {  
        echo    "$masukannama1    <br>    $masukannama3    <br>  
$masukannama2";  
    }
```

penggunaan kode tersebut merupakan kondisional bersarang untuk memeriksa apakah \$masukannama2 lebih kecil dari \$masukannama3. Jika benar, maka urutan nama yang benar adalah \$masukannama1, \$masukannama2, \$masukannama3, maka akan dicetak menggunakan pernyataan echo. Jika tidak, maka urutan nama yang benar dan akan dicetak adalah \$masukannama1, \$masukannama3, \$masukannama2.

Pada baris [30] [31] [32] [33] [34] [35], syntax

```
} elseif ($masukannama2 < $masukannama1 && $masukannama2 <  
$masukannama3) {  
    if ($masukannama1 < $masukannama3) {
```

```

        echo "$masukannama2 <br> $masukannama1
<br> $masukannama3";

    } else {

        echo "$masukannama2 <br> $masukannama3
<br> $masukannama1";

    }

```

pada baris-baris ini merupakan kondisional yang akan dievaluasi jika kondisional sebelumnya tidak terpenuhi, misalnya \$masukannama1 bukanlah yang terkecil. Kondisional ini kemudian memeriksa apakah \$masukannama2 lebih kecil dari kedua \$masukannama1 dan \$masukannama3. Jika benar, maka \$masukannama2 adalah yang terkecil di antara ketiganya. Seperti baris sebelumnya, terdapat juga kondisional bersarang di dalamnya untuk memeriksa apakah \$masukannama1 lebih kecil dari \$masukannama3. Jika benar, urutan yang benar adalah \$masukannama2, \$masukannama1, \$masukannama3, dan akan dicetak. Jika tidak sesuai, maka urutan yang benar dan akan dicetak adalah \$masukannama2, \$masukannama3, \$masukannama1.

Pada baris [36] [37] [38] [39] [40] [41] [42], syntax

```

    } else {

        if ($masukannama1 < $masukannama2) {

            echo "$masukannama3 <br> $masukannama1
<br> $masukannama3";

        } else {

            echo "$masukannama3 <br> $masukannama2
<br> $masukannama1";

        }
    }

```


}

pada baris-baris tersebut digunakan apabila kedua kondisional sebelumnya tidak terpenuhi, maka artinya \$masukannama3 adalah yang terkecil di antara ketiganya. Di dalam blok ini, dilakukan dua kondisional kembali untuk menentukan urutan nama yang benar, berdasarkan pada hubungan antara \$masukannama1 dan \$masukannama2. Jika \$masukannama1 lebih kecil dari \$masukannama2, maka urutan yang benar adalah \$masukannama3, \$masukannama1, \$masukannama2, dan akan dicetak. Jika tidak, urutan yang benar dan akan dicetak adalah \$masukannama3, \$masukannama2, \$masukannama1.

Pada baris [43], syntax `}` merupakan penutup dari baris 19.

Pada baris [44], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [45], syntax `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 10.

Pada baris [46], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 2

Buatlah program yang menerima input nama, nim, dan jenis kelamin seperti berikut:

Nama: *

Nim: *

Jenis Kelamin : *

☐ Laki-Laki

☐ Perempuan

Jika nama atau nim atau jenis kelamin kosong ketika tombol submit diklik, maka muncul pesan error seperti berikut.

Nama: *

Nim: * nim tidak boleh kosong

Jenis Kelamin : * jenis kelamin tidak boleh kosong

☐ Laki-Laki

☐ Perempuan

Jika semua isian sudah terisi maka tampilkan cetak hasil input seperti berikut.

Nama: *

Nim: *

Jenis Kelamin : *

☒ Laki-Laki

☐ Perempuan

Output:

Muhammad Fulan
123456
Laki-laki

Simpan dengan nama file: PRAK202.php

A. Source Code

Table 7. Source Code Modul 2 Soal 2

1	<?php
2	\$masukannama = \$masukannim = \$hasil =
	\$jeniskelamin="";
3	\$nama = \$nim= \$jenisk = "";
4	
5	if (isset(\$_POST['submit'])) {
6	if (empty(\$_POST['nama'])) {
7	\$masukannama = "Nama tidak boleh kosong";
8	} else {
9	\$nama = cek_input(\$_POST['nama']);
10	}
11	if (empty(\$_POST['nim'])) {
12	\$masukannim = "Nim tidak boleh kosong";
13	} else {
14	\$nim = cek_input(\$_POST['nim']);
15	}
16	if (empty(\$_POST['jenisk'])) {
17	\$jeniskelamin = "Jenis Kelamin tidak boleh
	kosong";
18	} else {
19	\$jenisk = cek_input(\$_POST['jenisk']);
20	}
21	}
22	
23	function cek_input(\$data) {
24	\$data = trim(\$data);
25	\$data = stripslashes(\$data);
26	\$data = htmlspecialchars(\$data);

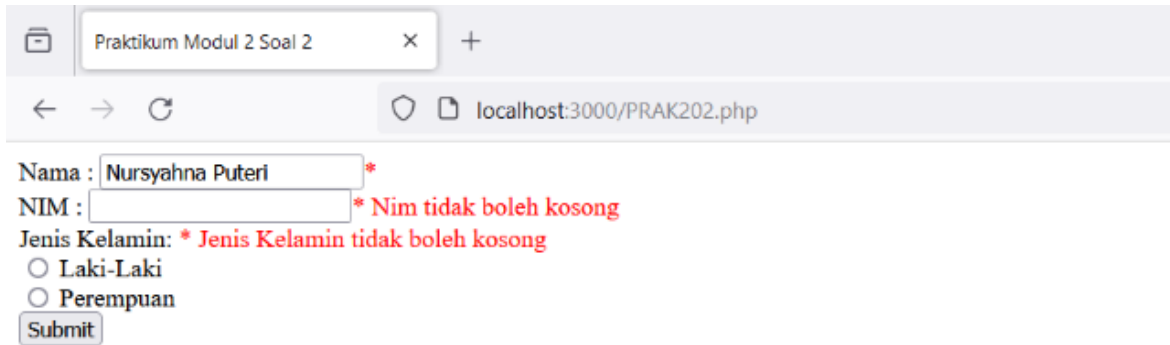
```

27         return $data;
28     }
29 ?>
30
31 <!DOCTYPE html>
32 <html lang="en">
33 <head>
34     <meta charset="UTF-8">
35     <meta http-equiv="X-UA-Compatible"
content="IE=edge">
36     <meta name="viewport" content="width=device-width,
initial-scale=1.0">
37     <title>Praktikum Modul 2 Soal 2</title>
38     <style>
39         .error {
40             color: red;
41         }
42     </style>
43 </head>
44
45 <body>
46     <form method="POST">
47         Nama           : <input type="text" name="nama"
value="<?= $nama ?>"><span class="error">*<?= $masukannama; ?></span></br>
48         NIM            : <input type="text" name="nim"
value="<?= $nim ?>"><span class="error">*<?= $masukannim; ?></span></br>
49         Jenis Kelamin: <span class="error">*<?=
$jeniskelamin; ?></span><br>

```

50	<code><input type="radio" name="jenisk" value="Laki-Laki" <?php if (isset(\$_POST["jenisk"]) && \$_POST["jenisk"] == "Laki-Laki") { echo "checked"; } ?>> Laki-Laki
</code>
51	<code><input type="radio" name="jenisk" value="Perempuan" <?php if (isset(\$_POST["jenisk"]) && \$_POST["jenisk"] == "Perempuan") { echo "checked"; } ?>> Perempuan
</code>
52	<code><input type="submit" name="submit" value="Submit"></code>
53	
54	<code></form></br></code>
55	
56	<code><?php</code>
	<code>if(!empty(\$nama) && !empty(\$nim) &&</code>
57	<code>!empty(\$jenisk)){</code>
58	<code>echo "\$nama
";</code>
59	<code>echo "\$nim
";</code>
60	<code>echo "\$jenisk
";</code>
61	<code>}</code>
62	<code>?></code>
63	<code></body></code>
64	<code></html></code>

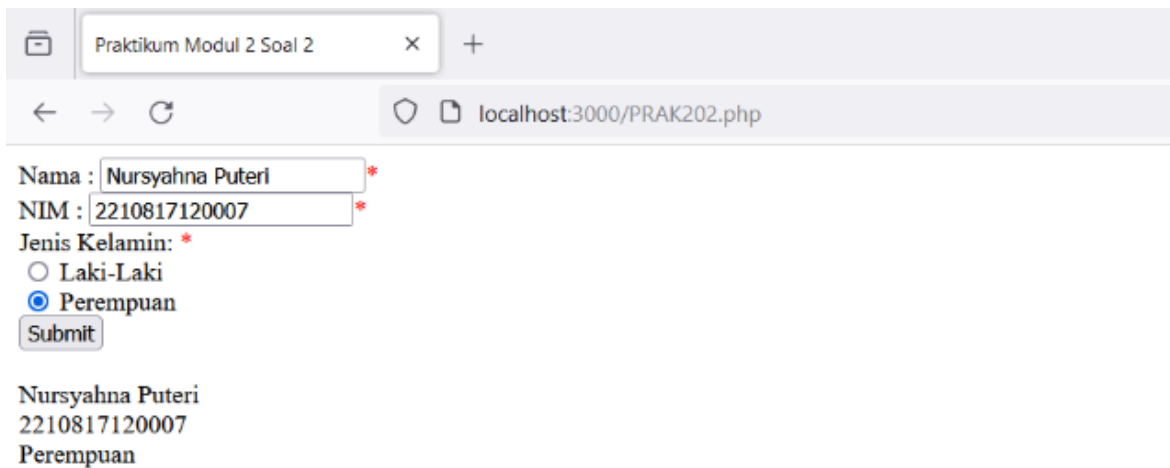
B. Output Program



A screenshot of a web browser window with the title 'Praktikum Modul 2 Soal 2'. The address bar shows 'localhost:3000/PRAK202.php'. The form contains the following fields and errors:

- Nama : Nursyahna Puteri *
- NIM : * Nim tidak boleh kosong
- Jenis Kelamin: * Jenis Kelamin tidak boleh kosong
- ☐ Laki-Laki
- ☐ Perempuan
- Submit

Gambar 7. Screenshot Hasil Jawaban Modul 2 Soal 2 (1)



A screenshot of a web browser window with the title 'Praktikum Modul 2 Soal 2'. The address bar shows 'localhost:3000/PRAK202.php'. The form contains the following fields and errors:

- Nama : Nursyahna Puteri *
- NIM : 2210817120007 *
- Jenis Kelamin: *
- ☐ Laki-Laki
- ☒ Perempuan
- Submit

Below the form, the following information is displayed:

Nursyahna Puteri
2210817120007
Perempuan

Gambar 8. Screenshot Hasil Jawaban Modul 2 Soal 2 (2)

C. Pembahasan

Pada baris [1], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [2], syntax `$masukannama = $masukannim = $hasil = $jeniskelamin=""`; empat variabel pada baris ini digunakan sebagai inisialisasi dengan nilai string kosong. Baris ini memiliki tujuan untuk menyiapkan variabel-variabel untuk menyimpan pesan kesalahan dari inputan pengguna.

Pada baris [3], syntax `$nama = $nim= $jenisk = ""`; baris ini merupakan inisialisasi dengan nilai string kosong. Ketiga variabel ini digunakan untuk menyimpan nilai yang diinputkan pengguna melalui formulir.

Pada baris [5], syntax `if (isset($_POST['submit']))` { baris ini merupakan kondisi atau blok utama yang akan memeriksa apakah tombol submit pada formulir telah ditekan. Jika tombol tersebut sudah ditekan, maka formulir telah disubmit dan proses pengolahan data harus dimulai.

Pada baris [6] [7] [8] [9] [10], syntax

```
if (empty($_POST['nama'])) {  
    $masukannama = "Nama tidak boleh kosong";  
} else {  
    $nama = cek_input($_POST['nama']);  
}
```

pada baris-baris tersebut merupakan pemeriksaan input nama dari pengguna. Pada blok pemeriksaan input ini akan dilakukan pemeriksaan apakah input untuk nama kosong atau tidak. Dalam melakukan pengecekannya maka digunakanlah `$_POST['nama']` untuk mengetahui data yang dikirimkan melalui formulir. Apabila input nama kosong, maka variabel `$masukannama` akan diisi dengan pesan kesalahan, yaitu berupa "Nama tidak boleh kosong". Pesan tersebut akan ditampilkan sebagai informasi kepada pengguna bahwa input nama tidak boleh kosong. Selanjutnya apabila input nama tidak kosong, maka nilai input tersebut akan melalui fungsi `cek_input()`. Fungsi ini juga bertanggung jawab untuk membersihkan data input dari spasi ekstra, karakter backslashes, dan karakter khusus lainnya yang tidak diinginkan. Setelah data sudah dipastikan benar, maka nilai input yang sudah ada akan disimpan dalam variabel `$nama`.

Pada baris [11] [12] [13] [14] [15], syntax

```

if (empty($_POST['nim'])) {

    $masukannim = "Nim tidak boleh kosong";

} else {

    $nim = cek_input($_POST['nim']);

}

```

pada baris-baris tersebut merupakan pemeriksaan input nim dari pengguna. Pada blok pemeriksaan input ini akan dilakukan pemeriksaan apakah input untuk NIM kosong atau tidak dengan menggunakan `$_POST['nim']`. Apabila input NIM dikosongkan oleh pengguna atau tidak terisi, maka variabel `$masukannim` akan diisi dengan pesan kesalahan, yaitu "Nim tidak boleh kosong". Pesan ini berfungsi untuk memberikan informasi kepada pengguna bahwa input NIM merupakan bagian yang wajib diisi pada formulir. Namun, jika input NIM tidak kosong, nilai input tersebut akan melewati fungsi `cek_input()`. Fungsi ini bertanggung jawab untuk membersihkan data input dari karakter-karakter yang tidak diperlukan atau tidak seharusnya diisikan. Setelah data sudah dipastikan benar, maka nilai input yang sudah ada akan disimpan dalam variabel `$nim`.

Pada baris [16] [17] [18] [19] [20] [21], syntax `if (empty($_POST['jenisk'])) {`

```

    $jeniskelamin = "Jenis Kelamin tidak boleh kosong";

} else {

    $jenisk = cek_input($_POST['jenisk']);

}

}

```

pada baris-baris tersebut merupakan pemeriksaan input jenis kelamin dari pengguna. Pada blok pemeriksaan input ini akan dilakukan pemeriksaan apakah input untuk jenis kelamin kosong atau tidak dengan menggunakan `$_POST['jenisk']`. Apabila input jenis

kelamin tidak diisi oleh pengguna, maka variabel \$jeniskelamin akan diisi dengan pesan kesalahan yang akan ditampilkan, berupa "Jenis Kelamin tidak boleh kosong". Pesan ini bertujuan untuk memberikan informasi kepada pengguna bahwa input jenis kelamin merupakan bagian yang wajib diisi pada formulir. Namun, apabila input jenis kelamin tidak kosong, nilai input tersebut akan melewati fungsi cek_input(). Fungsi ini bertanggung jawab untuk membersihkan data input dari karakter-karakter yang tidak diinginkan. Setelah data sudah dipastikan benar, maka nilai input yang sudah ada akan disimpan dalam variabel \$jenisk.

Pada baris [23], syntax `function cek_input($data) {` penggunaan baris ini adalah untuk membersihkan dan mengamankan data input yang diterima sebelum data tersebut digunakan lebih lanjut.

Pada baris [24], syntax `$data = trim($data);` pada baris ini digunakan untuk menghapus spasi kosong di awal dan di akhir string. Penggunaannya untuk memastikan bahwa tidak ada karakter yang tidak diinginkan atau spasi berlebih yang berada dalam data input.

Pada baris [25], syntax `$data = stripslashes($data);` pada baris ini digunakan untuk menghapus karakter backslashes (\) dari string. Penggunaannya untuk menghindari masalah atau error yang mungkin timbul karena karakter backslash yang terkadang dihasilkan oleh pelindung dari form HTML.

Pada baris [26], syntax `$data = htmlspecialchars($data);` pada baris ini digunakan untuk mengonversi karakter khusus HTML menjadi entitas HTML. Penggunaannya untuk dilakukan untuk mencegah serangan XSS (Cross-Site Scripting) dengan mengubah karakter-karakter yang memiliki makna tersendiri dalam HTML menjadi representasi entitas HTML, sehingga tidak akan dieksekusi oleh browser sebagai kode HTML.

Pada baris [27], syntax `return $data;` pada baris ini digunakan untuk mengembalikan data kembali ke tempat panggilan fungsi.

Pada baris [29], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [31], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [32], syntax `<html lang="en">` tag `html` merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [33], syntax `<head>` merupakan bagian kepala dari HTML. Tag `Head` berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag `Head` ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [34], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [35], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [36], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta `viewport` sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile.

Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [37], syntax `<title>Praktikum Modul 2 Soal 2</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [38] [39] [40] [41] [42], syntax

```
<style>

    .error {

        color: red;

    }

</style>
```

baris-baris tersebut merupakan tag style, dimana tag tersebut merupakan bagian HTML yang memiliki fungsi untuk mengatur gaya atau visual dari elemen-elemen HTML yang ingin ditampilkan. Pada baris tersebut tag style ditujukan untuk menetapkan gaya untuk elemen-elemen yang diberi kelas error. Penggunaan kode `.error` yang adalah selektor CSS akan menargetkan elemen-elemen yang memiliki kelas "error". Penggunaan tanda titik (.) sebelum nama kelas menunjukkan bahwa ini adalah kelas CSS. Pada aturan gaya CSS yang terdapat di dalam kurung kurawal (`{ }`), properti `color` ditetapkan ke nilai "red", yang berarti teks pada elemen-elemen yang memiliki kelas "error" akan ditampilkan dengan warna merah.

Pada baris [43], syntax `</head>` merupakan tag penutup HTML dari baris 33.

Pada baris [45], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [46], syntax `<form method="POST">` merupakan tag form yang digunakan untuk membuat formulir di dalam halaman web. Pada baris ini menggunakan atribut berupa `method="POST"` dimana metode tersebut digunakan untuk pengiriman data formulir ke server untuk diproses.

Pada baris [47], syntax Nama : `<input type="text" name="nama" value="<?=$nama ?>">*<?=$masukannama; ?></br>` penggunaan baris ini adalah tempat input teks bagi pengguna untuk memasukkan nama. Penggunaan atribut `name="nama"` adalah untuk menentukan nama yang akan digunakan ketika data dikirimkan. Nilai input untuk nama diambil dari variabel PHP `$nama`, dan akan ditampilkan kembali di dalam input jika terjadi kesalahan validasi. Pada baris ini terdapat juga penggunaan tag `` yang dipergunakan untuk menampilkan pesan kesalahan. Apabila ada yang disimpan dalam variabel PHP `$masukannama`. Selain itu juga terdapat tag `</br>` penggunaan tag ini adalah untuk membuat baris baru di dalam tampilan web.

Pada baris [48], syntax NIM : `<input type="text" name="nim" value="<?=$nim ?>">*<?=$masukannim; ?></br>` penggunaan baris ini adalah tempat input teks bagi pengguna untuk memasukkan NIM. Nilai input untuk NIM diambil dari variabel PHP `$nim` dan akan ditampilkan kembali di dalam input jika terjadi kesalahan validasi. Pada baris ini terdapat juga penggunaan tag `` yang dipergunakan untuk menampilkan pesan kesalahan. Apabila ada yang disimpan dalam variabel PHP `$masukannim`. Selain itu juga terdapat tag `</br>` penggunaan tag ini adalah untuk membuat baris baru di dalam tampilan web.

Pada baris [49] [50] [51] [52], syntax

Jenis Kelamin: `*<?=$jeniskelamin; ?>
`

`<input type="radio" name="jenisk" value="Laki-Laki" <?php if (isset($_POST["jenisk"])) &&`

```
$_POST["jenisk"] == "Laki-Laki") { echo "checked"; } ?>>
Laki-Laki<br>
```

```

☐

```

```
<input type="submit" name="submit" value="Submit">
```

penggunaan baris-baris tersebut adalah untuk menampilkan label “Jenis Kelamin”. Pada baris ini terdapat juga sebuah elemen `` yang bertugas untuk menampilkan pesan kesalahan terkait jenis kelamin, apabila ada. Pesan kesalahan ini disimpan dalam variabel PHP `$jeniskelamin` dan akan ditampilkan di samping label jenis kelamin tersebut. Pada baris tersebut terdapat dua input radio untuk memilih jenis kelamin. Input radio pertama memiliki nilai "Laki-Laki" dan input radio kedua memiliki nilai "Perempuan". Keduanya memiliki atribut `name` yang sama, yaitu "jenisk", sehingga hanya satu dari keduanya yang dapat dipilih. Apabila pengguna sudah memilih jenis kelamin sebelumnya, yaitu berdasarkan data POST yang diterima maka input radio yang sesuai akan ditandai sebagai terpilih. Lalu pada baris ini tentunya terdapat button submit yang bertugas mengirimkan data formulir ke sever saat pengguna mengkliknya. Atribut `name="submit"` memberikan nama identifikasi untuk tombol submit.

Pada baris [54], syntax `</form></br>` pada baris ini merupakan tag penutup dari baris 46.

Pada baris [56], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [57] [58] [59] [60] [61], syntax

```
if(!empty($nama) && !empty($nim) && !empty($jenisk)){
    echo "$nama <br>";
```

```

        echo "$nim <br>";

        echo "$jenisk <br>";

    }

```

penggunaan baris-baris tersebut adalah digunakan untuk memeriksa apakah variabel \$nama, \$nim, dan \$jenisk tidak kosong yaitu ditandai dengan digunakannya !empty. Jika ketiga variabel tersebut tidak kosong, maka blok kode di dalamnya akan dieksekusi. Penggunaan if(!empty(\$nama) && !empty(\$nim) && !empty(\$jenisk)){ merupakan kondisi yang digunakan untuk memeriksa apakah variabel \$nama, \$nim, dan \$jenisk tidak kosong. !empty() merupakan fungsi yang mengembalikan nilai true jika variabel yang diberikan tidak kosong. Jika semua variabel tidak kosong, maka kondisi ini akan bernilai true atau benar. echo "\$nama
"; adalah bagian dari blok kode yang akan dieksekusi jika kondisi sebelumnya terpenuhi. Baris ini mencetak isi dari variabel \$nama ke layar, diikuti dengan tag
 untuk membuat baris baru. echo "\$nim
"; akan mencetak isi dari variabel \$nim ke layar, terdapat juga tag
 untuk membuat baris baru. Serta terdapat juga echo "\$jenisk
"; yang akan mencetak isi dari variabel \$jenisk ke layar, diikuti dengan tag
 untuk membuat baris baru.

Pada baris [62], syntax ?> digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [63], syntax </body> pada baris ini merupakan tag penutup dari tag pembuka baris 45.

Pada baris [64], syntax </html> baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 3

Buatlah sebuah program untuk melakukan konversi suhu antara Celcius (C), Fahrenheit (F), Reamur (Re) dan Kelvin (K).

Output yang diinginkan:

Nilai :

Dari :

- ☒ Celcius
- ☐ Fahrenheit
- ☐ Rheamur
- ☐ Kelvin

Ke :

- ☐ Celcius
- ☒ Fahrenheit
- ☐ Rheamur
- ☐ Kelvin

Hasil Konversi: 136.4 °F

Simpan dengan nama file: PRAK203.php

A. Source Code

Table 8. Source Code Modul 2 Soal 3

1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
6	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
7	<title>Praktikum Modul 2 Soal 3</title>
8	</head>
9	<body>
10	<form action="" method="post">

	<pre> Nilai <input type="number" step="any" 11 name="suhu" value="<?php if(isset(\$_POST['suhu'])) { echo \$_POST['suhu']; } ?>"></br> Dari : </br> 12 <input type="radio" name="suhuawal" 13 value="Celcius" <?php if(isset(\$_POST['suhuawal']) && \$_POST['suhuawal'] == "Celcius") { echo "checked"; } ?>>Celcius</br> <input type="radio" name="suhuawal" 14 value="Fahrenheit" <?php if(isset(\$_POST['suhuawal']) && \$_POST['suhuawal'] == "Fahrenheit") { echo "checked"; } ?>>Fahrenheit</br> <input type="radio" name="suhuawal" 15 value="Reamur" <?php if(isset(\$_POST['suhuawal']) && \$_POST['suhuawal'] == "Reamur") { echo "checked"; } ?>>Reamur</br> <input type="radio" name="suhuawal" 16 value="Kelvin" <?php if(isset(\$_POST['suhuawal']) && \$_POST['suhuawal'] == "Kelvin") { echo "checked"; } ?>>Kelvin</br> 17 Ke : </br> 18 <input type="radio" name="konversisuhu" 19 value="Celcius" <?php if(isset(\$_POST['konversisuhu']) && \$_POST['konversisuhu'] == "Celcius") { echo "checked"; } ?>>Celcius</br> <input type="radio" name="konversisuhu" 20 value="Fahrenheit" <?php if(isset(\$_POST['konversisuhu']) && </pre>
--	--

	<code>\$_POST['konversisuhu'] == "Fahrenheit") { echo</code>
	<code>"checked"; } ?>>Fahrenheit</br></code>
21	<code> <input type="radio" name="konversisuhu"</code>
	<code>value="Reamur" <?php if(isset(\$_POST['konversisuhu'])</code>
	<code>&& \$_POST['konversisuhu'] == "Reamur") { echo</code>
	<code>"checked"; } ?>>Reamur</br></code>
22	<code> <input type="radio" name="konversisuhu"</code>
	<code>value="Kelvin" <?php if(isset(\$_POST['konversisuhu'])</code>
	<code>&& \$_POST['konversisuhu'] == "Kelvin") { echo</code>
	<code>"checked"; } ?>>Kelvin</br></code>
23	
24	<code> <input type="submit" name="submit"</code>
	<code>value="Konversi"></input></code>
25	<code></form></code>
26	
27	<code><?php</code>
28	<code>if (isset(\$_POST["submit"])) {</code>
29	<code> \$suhu = \$_POST['suhu'];</code>
30	<code> \$suhuawal = \$_POST['suhuawal'];</code>
31	<code> \$konversisuhu = \$_POST['konversisuhu'];</code>
32	
33	<code> switch (\$suhuawal) {</code>
34	<code> case "Celcius":</code>
35	<code> switch (\$konversisuhu) {</code>
36	<code> case "Celcius":</code>
37	<code> echo "<h2>Hasil Konversi : \$suhu</code>
38	<code>&deg;C</h2>";</code>
39	<code> break;</code>
	<code> case "Fahrenheit":</code>
40	

```

41             echo "<h2>Hasil Konversi : " .
42 ($suhu * 1.8 + 32) . " &deg;F</h2>";
43             break;
44             case "Reamur":
45             echo "<h2>Hasil Konversi : " .
46 ($suhu * 0.8) . " &deg;R</h2>";
47             break;
48             case "Kelvin":
49             echo "<h2>Hasil Konversi : " .
50 ($suhu + 273.15) . " &deg;K</h2>";
51             break;
52         }
53         break;
54         case "Fahrenheit":
55         switch ($konversisuhu) {
56             case "Celcius":
57             echo "<h2>Hasil Konversi : " .
58 ($suhu - 32) / 1.8 . " &deg;C</h2>";
59             break;
60             case "Fahrenheit":
61             echo "<h2>Hasil Konversi : $suhu
62 &deg;F</h2>";
63             break;
64             case "Reamur":
65             echo "<h2>Hasil Konversi : " .
66 ($suhu - 32) / 2.25 . " &deg;R</h2>";
67             break;
68             case "Kelvin":
69             echo "<h2>Hasil Konversi : " .
70 ($suhu + 459.67) / 1.8 . " &deg;K</h2>";

```

	break;
65	}
66	break;
67	case "Reamur":
68	switch (\$konversisuhu) {
69	case "Celcius":
70	echo "<h2>Hasil Konversi : " .
71	(\$suhu * 1.25) . " °C</h2>";
	break;
72	case "Fahrenheit":
73	echo "<h2>Hasil Konversi : " .
74	(\$suhu * 2.25 + 32) . " °F</h2>";
	break;
75	case "Reamur":
76	echo "<h2>Hasil Konversi : \$suhu
77	°R</h2>";
	break;
78	case "Kelvin":
79	echo "<h2>Hasil Konversi : " .
80	(\$suhu + 273.15) / 0.8 . " °K</h2>";
	break;
81	}
82	break;
83	case "Kelvin":
84	switch (\$konversisuhu) {
85	case "Celcius":
86	echo "<h2>Hasil Konversi : " .
87	(\$suhu - 273.15) . " °C</h2>";
	break;
88	case "Fahrenheit":

89	echo "<h2>Hasil Konversi : " .
90	(\$suhu * 1.8 - 459.67) . " °F</h2>";
	break;
91	case "Reamur":
92	echo "<h2>Hasil Konversi : " .
93	(\$suhu - 273.15) * 0.8 . " °R</h2>";
	case "Kelvin":
94	echo "<h2>Hasil Konversi : \$suhu
95	°R</h2>";
	}
96	}
97	}
98	?>
99	</body>
	</html>

B. Output Program

Praktikum Modul 2 Soal 3

localhost:3000/tes.php

Nilai

Dari :

☒ Celcius

☐ Fahrenheit

☐ Reamur

☐ Kelvin

Ke :

☐ Celcius

☒ Fahrenheit

☐ Reamur

☐ Kelvin

Hasil Konversi : 136.4 °F

Gambar 9. Screenshot Hasil Jawaban Modul 2 Soal 3

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [3], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [4], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris 35], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [6], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [7], syntax `<title>Praktikum Modul 2 Soal 3</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [8], syntax `</head>` merupakan tag penutup HTML dari baris 3.

Pada baris [9], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [10], syntax `<form method="POST">` merupakan tag form yang digunakan untuk membuat formulir di dalam halaman web. Pada baris ini menggunakan atribut berupa `method="POST"` dimana metode tersebut digunakan untuk pengiriman data formulir ke server untuk diproses.

Pada baris [11], syntax Nilai `<input type="number" step="any" name="suhu" value="<?php if(isset($_POST['suhu'])) { echo $_POST['suhu']; } ?>"></br>` pada baris tersebut kegunaannya adalah sebagai tempat input field atau tempat memasukkan nilai suhu dalam bentuk angka. Pada baris ini terdapat atribut `type="number"` yang menunjukkan bahwa input ini hanya menerima nilai numerik. Atribut `name="suhu"` adalah nama dari input, yang akan digunakan untuk mengidentifikasi data ketika dikirimkan ke server. Selanjutnya Atribut `value="<?php if(isset($_POST['suhu'])) { echo $_POST['suhu']; } ?>"` berfungsi untuk menampilkan kembali nilai yang dimasukkan sebelumnya oleh pengguna jika formulir sudah pernah disubmit sebelumnya.

Pada baris [12], syntax `Dari : </br>` penggunaan dari baris ini adalah adalah keterangan dari manakah konversi yang akan dilakukan, pada baris ini terdapat tag `
` yang berarti untuk memisahkan label “Dari :” dengan grup input radio yang ada nantinya.

Pada baris [13] [14] [15] [16], syntax

```
<input type="radio" name="suhuawal" value="Celcius" <?php  
if(isset($_POST['suhuawal']) && $_POST['suhuawal'] ==  
"Celcius") { echo "checked"; } ?>>Celcius</br>
```

```
        <input                type="radio"                name="suhuawal"  
value="Fahrenheit" <?php if(isset($_POST['suhuawal']) &&  
$_POST['suhuawal'] == "Fahrenheit") { echo "checked"; }  
?>>Fahrenheit</br>
```

```
        <input type="radio" name="suhuawal" value="Reamur"  
<?php if(isset($_POST['suhuawal']) && $_POST['suhuawal']  
== "Reamur") { echo "checked"; } ?>>Reamur</br>
```

```
        <input type="radio" name="suhuawal" value="Kelvin"  
<?php if(isset($_POST['suhuawal']) && $_POST['suhuawal']  
== "Kelvin") { echo "checked"; } ?>>Kelvin</br>
```

pada baris-baris tersebut merupakan elemen input dengan tipe radio dalam sebuah HTML, pada baris-baris tersebut adalah empat tombol radio yang mewakili opsi dari konversi suhu. Penggunaan tombol radio `<input type="radio" name="suhuawal" value="Celcius">Celcius</br>` menunjukkan opsi konversi suhu dari Celcius, dengan properti name adalah suhuawal, pengelompokkan semua input radio tersebut bersama-sama, membatasi pengguna untuk memilih hanya satu opsi. Selanjutnya terdapat properti value adalah Celcius, yang akan dikirimkan ke server sebagai nilai jika opsi ini dipilih. Jika kondisi yang diuji dalam tag PHP `<?php if(isset($_POST['suhuawal']) && $_POST['suhuawal'] == "Celcius") { echo "checked"; } ?>` terpenuhi, maka atribut checked akan ditambahkan, sehingga tombol radio ini akan ditandai sebagai dipilih secara default. Hal tersebut juga berlaku pada radio yang bervalue Fahrenheit, Reamur, serta Kelvin. Pada baris-baris tersebut juga terdapat tag `
` yang berarti digunakan sebagai pemisah tombol radio tersebut.

Pada baris [18], syntax Ke : `</br>` penggunaan dari baris ini adalah adalah keterangan dari manakah konversi yang akan dilakukan, pada baris ini terdapat tag `
` yang berarti untu memisahkan label “Ke :” dengan grup input radio yang ada nantinya.

Pada baris [19] [20] [21] [22], syntax

```
input type="radio" name="konversisuhu" value="Celcius"
<?php if(isset($_POST['konversisuhu']))      &&
$_POST['konversisuhu'] == "Celcius") { echo "checked"; }
?>>Celcius</br>
```

```
        <input type="radio" name="konversisuhu"
value="Fahrenheit" <?php if(isset($_POST['konversisuhu']))
&& $_POST['konversisuhu'] == "Fahrenheit") { echo
"checked"; } ?>>Fahrenheit</br>
```

```
        <input type="radio" name="konversisuhu"
value="Reamur" <?php if(isset($_POST['konversisuhu'])) &&
$_POST['konversisuhu'] == "Reamur") { echo "checked"; }
?>>Reamur</br>
```

```
        <input type="radio" name="konversisuhu"
value="Kelvin" <?php if(isset($_POST['konversisuhu'])) &&
$_POST['konversisuhu'] == "Kelvin") { echo "checked"; }
?>>Kelvin</br>
```

pada baris-baris tersebut merupakan elemen input dengan tipe radio dalam sebuah HTML, pada baris-baris tersebut adalah empat tombol radio yang mewakili opsi dari konversi suhu. Penggunaan tombol radio `<input type="radio" name="konversisuhu" value="Celcius">Celcius</br>` menunjukkan opsi konversi suhu ke Celcius, dengan properti name adalah konersisuhu, pengelompokkan semua input radio tersebut bersama-sama, membatasi pengguna untuk memilih hanya satu opsi. Selanjutnya terdapat properti value adalah Celcius, yang akan dikirimkan ke server sebagai nilai

jika opsi ini dipilih. Jika kondisi yang diuji dalam tag PHP `<?php if(isset($_POST['suhuawal']) && $_POST['suhuawal'] == "Celcius") { echo "checked"; } ?>` terpenuhi, maka atribut checked akan ditambahkan, sehingga tombol radio ini akan ditandai sebagai dipilih secara default. Hal tersebut juga berlaku pada radio yang bervalue Fahrenheit, Reamur, serta Kelvin. Pada baris-baris tersebut juga terdapat tag `
` yang berarti digunakan sebagai pemisah tombol radio tersebut.

Pada baris [24], syntax `<input type="submit" name="submit" value="Konversi"></input>` pada baris tersebut berfungsi untuk membuat tombol 'submit' untuk mengirimkan data dalam suatu formulir HTML. Pada baris tersebut terdapat atribut dengan keterangan `type="submit"`: memiliki tujuan untuk menetapkan jenis input sebagai tombol "submit". Tombol "submit" digunakan untuk mengirimkan data formulir ke server saat tombol tersebut ditekan. `name="submit"`: memberikan nama pada input yang akan digunakan untuk mengidentifikasi input tersebut saat formulir dikirimkan ke server. Pada formulir ini yaitu nama input adalah "submit". Selanjutnya digunakan juga atribut `value="Konversi"`: untuk menentukan teks atau nilai yang akan dikirimkan ke server ketika tombol tersebut ditekan. Pada kode tersebut, teks yang akan ditampilkan pada tombol submit adalah "Konversi". Sehingga ketika pengguna mengklik tombol "Konversi" maka data pada formulir akan terkirimkan ke server untuk diproses.

Pada baris [27], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [28] [29] [30] [31], syntax

```
if (isset($_POST["submit"])) {  
    $suhu = $_POST['suhu'];  
    $suhuawal = $_POST['suhuawal'];  
    $konversisuhu = $_POST['konversisuhu'];
```

penggunaan baris-baris tersebut adalah skrip PHP yang akan mengelola data formulir yang dikirimkan melalui metode POST. Pada baris tersebut terdapat penggunaan `if (isset($_POST["submit"]))`: penggunaannya ditujukan untuk sebuah kondisi yang memeriksa apakah data formulir telah dikirimkan dengan menggunakan metode POST dan tombol "submit" telah diklik. `$_POST["submit"]` akan ada nilainya jika tombol "submit" dalam formulir sebelumnya telah diklik, menandakan bahwa data formulir telah dikirimkan. Selanjutnya terdapat variabel `$suhu = $_POST['suhu'];`: penggunaan baris tersebut adalah untuk mengambil nilai yang dikirimkan melalui input dengan nama "suhu" dari formulir yang dikirimkan. Nilai ini kemudian disimpan dalam variabel `$suhu` untuk digunakan dalam proses konversi suhu. Lalu penggunaan `$suhuawal = $_POST['suhuawal'];`: kode ini bertujuan untuk menangkap nilai dari input radio dengan nama "suhuawal" dari formulir yang dikirimkan. Ini mengidentifikasi satuan suhu asal yang akan dikonversi. Dan penggunaan `$konversisuhu = $_POST['konversisuhu'];`: baris ini bertujuan untuk mengambil nilai dari input radio dengan nama "konversisuhu" dari formulir yang dikirimkan. Nilai ini menentukan satuan suhu yang diinginkan untuk hasil konversi. Sehingga dengan baris-baris tersebut data formulir yang dikirimkan oleh pengguna dapat diakses dan digunakan dalam proses berikutnya.

Pada baris [33] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49], syntax

```
switch ($suhuawal) {  
    case "Celcius":  
        switch ($konversisuhu) {  
            case "Celcius":  
                echo "<h2>Hasil Konversi : $suhu  
&deg;C</h2>";  
                break;
```

```

        case "Fahrenheit":

            echo "<h2>Hasil Konversi : " . ($suhu
* 1.8 + 32) . " &deg;F</h2>";

            break;

        case "Reamur":

            echo "<h2>Hasil Konversi : " . ($suhu
* 0.8) . " &deg;R</h2>";

            break;

        case "Kelvin":

            echo "<h2>Hasil Konversi : " . ($suhu
+ 273.15) . " &deg;K</h2>";

            break;

    }

    break;

```

penggunaan baris-baris tersebut merupakan switch-case yang digunakan untuk melakukan konversi suhu berdasarkan satuan suhu awal yang dipilih oleh pengguna adalah Celcius. Pada baris-baris tersebut terdapat pernyataan switch yang mengevaluasi variabel \$suhuawal, yang merepresentasikan satuan suhu awal yang dipilih oleh pengguna. Setelah itu, terdapat beberapa blok case yang akan memproses konversi berdasarkan satuan yang dipilih. Di dalam blok case "Celcius", terdapat switch-case kedua yang mengevaluasi variabel \$konversisuhu, yang merupakan satuan suhu yang diinginkan untuk hasil konversi. Pada blok case tersebut, terdapat beberapa case lagi yang menangani konversi ke berbagai satuan suhu, seperti Fahrenheit, Reamur, dan Kelvin. Setiap blok case akan menghitung hasil konversi sesuai dengan rumus yang sesuai dengan satuan yang dipilih oleh pengguna, selanjutnya akan menampilkan hasilnya dalam format yang ditentukan. Setelah blok case dijalankan,

eksekusi kode akan keluar dari struktur switch-case dan melanjutkan ke kode berikutnya di luar switch-case tersebut.

Pada baris [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65],
syntax

```
case "Fahrenheit":

    switch ($konversisuhu) {

        case "Celcius":

            echo "<h2>Hasil Konversi : " . ($suhu
- 32) / 1.8 . " &deg;C</h2>";

            break;

        case "Fahrenheit":

            echo  "<h2>Hasil  Konversi   :  $suhu
&deg;F</h2>";

            break;

        case "Reamur":

            echo "<h2>Hasil Konversi : " . ($suhu
- 32) / 2.25 . " &deg;R</h2>";

            break;

        case "Kelvin":

            echo "<h2>Hasil Konversi : " . ($suhu
+ 459.67) / 1.8 . " &deg;K</h2>";

            break;

    }

    break;
```

penggunaan baris-baris tersebut merupakan switch-case yang digunakan untuk melakukan konversi suhu ketika satuan suhu awal yang dipilih oleh pengguna adalah Fahrenheit. Di dalam struktur ini, terdapat beberapa blok case yang menangani konversi ke berbagai satuan suhu lainnya. Pada mulanya, terdapat blok case yang menangani kondisi di mana satuan suhu awal yang dipilih adalah Fahrenheit. Dalam blok case ini, terdapat switch-case kedua yang mengevaluasi variabel \$konversisuhu, yang merepresentasikan satuan suhu yang diinginkan untuk hasil konversi dari Fahrenheit. Di dalam switch-case kedua, terdapat beberapa blok case yang menangani konversi ke berbagai satuan suhu, seperti Celcius, Reamur, dan Kelvin. Setiap blok case ini akan menghitung hasil konversi sesuai dengan rumus yang sesuai dengan satuan yang dipilih oleh pengguna, kemudian menampilkan hasilnya dalam format yang ditentukan. Sebagai contoh, jika pengguna memilih untuk mengonversi suhu Fahrenheit ke Celcius, nilai suhu Fahrenheit akan dikonversi ke Celcius menggunakan rumus yang sesuai, yaitu $(\$suhu - 32) / 1.8$. Hasil konversi tersebut kemudian akan ditampilkan dalam format "

Hasil Konversi : " . $(\$suhu - 32) / 1.8$. " °C</h2>". Setelah blok case dalam switch-case kedua dievaluasi dan kode di dalamnya dijalankan, PHP akan keluar dari switch-case kedua dan melanjutkan eksekusi kode berikutnya setelah blok switch-case pertama.

Pada baris [66] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81], syntax

```
case "Reamur":

    switch ($konversisuhu) {

        case "Celcius":

            echo "<h2>Hasil Konversi : " . ($suhu
* 1.25) . " &deg;C</h2>";

            break;

        case "Fahrenheit":
```

```

        echo "<h2>Hasil Konversi : " . ($suhu
* 2.25 + 32) . " &deg;F</h2>";

        break;

    case "Reamur":

        echo "<h2>Hasil Konversi : $suhu
&deg;R</h2>";

        break;

    case "Kelvin":

        echo "<h2>Hasil Konversi : " . ($suhu
+ 273.15) / 0.8 . " &deg;K</h2>";

        break;

    }

    break;

```

penggunaan baris-baris tersebut merupakan switch-case yang digunakan untuk melakukan konversi suhu ketika satuan suhu awal yang dipilih oleh pengguna adalah Reamur. Di dalam struktur ini, terdapat beberapa blok case yang menangani konversi ke berbagai satuan suhu lainnya. Pertama-tama akan terdapat blok case yang menangani kondisi di mana satuan suhu awal yang dipilih adalah Reamur. Dalam blok case ini, terdapat switch-case kedua yang mengevaluasi variabel \$konversisuhu, yang merepresentasikan satuan suhu yang diinginkan untuk hasil konversi dari Reamur. Di dalam switch-case kedua, terdapat beberapa blok case yang menangani konversi ke berbagai satuan suhu, seperti Celcius, Fahrenheit, dan Kelvin. Setiap blok case ini akan menghitung hasil konversi sesuai dengan rumus yang sesuai dengan satuan yang dipilih oleh pengguna, kemudian menampilkan hasilnya dalam format yang ditentukan. Misalnya apabila pengguna memilih untuk mengonversi suhu Reamur ke Celcius, nilai suhu Reamur akan dikonversi ke Celcius menggunakan rumus yang sesuai, yaitu \$suhu

* 1.25. Hasil konversi tersebut kemudian akan ditampilkan dalam format "<h2>Hasil Konversi : " . (\$suhu * 1.25) . " °C</h2>".

Setelah blok case dalam switch-case kedua dievaluasi dan kode di dalamnya dijalankan, PHP akan keluar dari switch-case kedua dan melanjutkan eksekusi kode berikutnya setelah blok switch-case pertama. Dengan demikian, struktur switch-case ini memungkinkan program untuk melakukan konversi suhu berdasarkan pilihan yang diberikan oleh pengguna.

Pada baris [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96], syntax

```
case "Kelvin":

    switch ($konversisuhu) {

        case "Celcius":

            echo "<h2>Hasil Konversi : " . ($suhu
- 273.15) . " &deg;C</h2>";

            break;

        case "Fahrenheit":

            echo "<h2>Hasil Konversi : " . ($suhu
* 1.8 - 459.67) . " &deg;F</h2>";

            break;

        case "Reamur":

            echo "<h2>Hasil Konversi : " . ($suhu
- 273.15) * 0.8 . " &deg;R</h2>";

            case "Kelvin":
```

```

        echo "<h2>Hasil Konversi : $suhu
        &deg;R</h2>";
    }

}

}

```

penggunaan baris-baris tersebut merupakan switch-case yang digunakan untuk melakukan konversi suhu ketika satuan suhu awal yang dipilih oleh pengguna adalah Kelvin. Di dalam struktur ini, terdapat beberapa blok case yang menangani konversi ke berbagai satuan suhu lainnya. Pertama-tama terdapat blok case yang menangani kondisi di mana satuan suhu awal yang dipilih adalah Kelvin. Dalam blok case ini, terdapat switch-case kedua yang mengevaluasi variabel \$konversisuhu, yang merepresentasikan satuan suhu yang diinginkan untuk hasil konversi dari Kelvin. Di dalam switch-case kedua, terdapat beberapa blok case yang mengatur konversi ke berbagai satuan suhu, seperti Celcius, Fahrenheit, dan Reamur. Setiap blok case ini akan menghitung hasil konversi sesuai dengan rumus yang sesuai dengan satuan yang dipilih oleh pengguna, kemudian menampilkan hasilnya dalam format yang ditentukan. Misalnya apabila pengguna memilih untuk mengonversi suhu Kelvin ke Celcius, nilai suhu Kelvin akan dikonversi ke Celcius menggunakan rumus yang sesuai, yaitu \$suhu - 273.15. Hasil konversi tersebut kemudian akan ditampilkan dalam format "<h2>Hasil Konversi : " . (\$suhu - 273.15) . " °C</h2>". Setelah blok case dalam switch-case kedua dievaluasi dan kode di dalamnya dijalankan, PHP akan keluar dari switch-case kedua dan melanjutkan eksekusi kode berikutnya setelah blok switch-case pertama.

Pada baris [97], syntax ?> digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [98], </body> pada baris ini merupakan tag penutup dari tag pembuka baris 9.

Pada baris [99], `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 4

Buatlah program untuk membaca ejaan dari bilangan cacah Format Masukan :

Sebuah baris berisi satu buah bilangan cacah dan batas maksimal 999. ($a \geq 0 < 1000$)

Format Keluaran :

Sebuah baris berisi sebuah bilangan cacah yang merupakan Ejaan dari bilangan tersebut atau batas limit bilangan.

Input	Output
3	Satuan
0	Nol
100	Anda Menginput Melebihi Limit Bilangan
632	Ratusan
13	Belasan

Output yang diinginkan:

Nilai :

Hasil: belasan

Simpan dengan nama file: PRAK204.php

A. Source Code

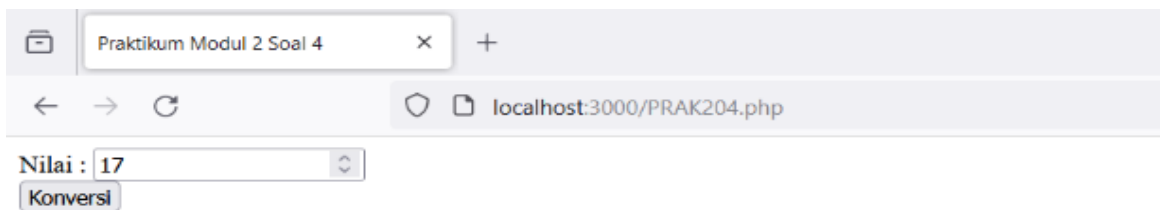
Table 9. Source Code Modul 2 Soal 4

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	

6	<pre> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Praktikum Modul 2 Soal 4</title> </head> <body> <form action="" method="POST"> Nilai : <input type="number" name="nilai" value="<?php if(isset(\$_POST['nilai'])) { echo \$_POST['nilai']; } ?>"></input></br> <input type="submit" name="submit" value="Konversi"></input> </form> <?php if (isset(\$_POST["submit"])) { \$nilai = \$_POST["nilai"]; if (\$nilai == 0){ echo "<h2>Hasil : Nol</h2>"; } elseif (\$nilai > 0 && \$nilai < 10){ echo "<h2>Hasil : Satuan</h2>"; } elseif (\$nilai >= 11 && \$nilai < 20){ echo "<h2>Hasil : Belasan</h2>"; } elseif (\$nilai >= 10 && \$nilai < 100){ echo "<h2>Hasil : Puluhan</h2>"; } elseif (\$nilai >= 100 && \$nilai < 1000){ echo "<h2>Hasil : Ratusan</h2>"; } else{ </pre>
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	

31	echo "<h2>Hasil : Anda Menginput Melebihi
	Limit Bilangan</h2>";
32	}
33	}
34	?>
35	</body>
	</html>

B. Output Program



Hasil : Belasan

Gambar 10. Screenshot Hasil Jawaban Modul 2 Soal 4

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [3], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [4], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris 35], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [6], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [7], syntax `<title>Praktikum Modul 2 Soal 4</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [8], syntax `</head>` merupakan tag penutup HTML dari baris 3.

Pada baris [10], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [11], syntax `<form method="POST">` merupakan tag form yang digunakan untuk membuat formulir di dalam halaman web. Pada baris ini menggunakan atribut berupa `method="POST"` dimana metode tersebut digunakan untuk pengiriman data formulir ke server untuk diproses.

Pada baris [12], syntax Nilai : `<input type="number" name="nilai" value="<?php if(isset($_POST['nilai'])) { echo $_POST['nilai']; } ?>"></input></br>` baris ini digunakan sebagai input field atau tempat dimana pengguna dapat memasukkan nilai numerik yang diinginkan, pengguna hanya dapat memasukan nilai numerik saja karena pada baris ini digunakan atribut `type="number"`. Pada penggunaan kode pada baris ini terdapat juga atribut `value`, yang digunakan untuk memeriksa apakah ada data yang dikirimkan melalui metode POST dengan nama "nilai". Apabila ada nilai yang dimasukkan sebelumnya maka akan ditampilkan Kembali di dalam input field, sehingga memungkinkan pengguna untuk melihat dan mengedit nilai yang telah dimasukkan sebelumnya. Selanjutnya pada baris ini terdapat tag `
` yang digunakan untuk membuat baris baru, sehingga elemen-elemen berikutnya akan dimulai di baris yang baru.

Pada baris [13], syntax `<input type="submit" name="submit" value="Konversi"></input>` penggunaan pada baris ini merupakan tempat dimana formulir akan dikirimkan. Tombol "submit" memiliki peran penting dalam formulir, ketika tombol ditekan maka formulir akan dikirimkan ke server. Atribut `type="submit"` yang menandakan bahwa tombol yang ada merupakan tombol submit, yang berarti saat di klik maka formulir akan terkirim. Selanjutnya terdapat juga `value`, yaitu dengan nilai konversi. Kegunaan `value` tersebut adalah untuk menentukan teks yang akan ditampilkan pada tombol submit. Dalam hal ini, teks yang ditampilkan adalah "Konversi, sehingga pengguna dapat menggunakan tombol ini untuk melakukan konversi nilai yang dimasukkan.

Pada baris [14], syntax `</form>` baris ini merupakan tag penutup dari baris 11.

Pada baris [16], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [17], syntax `if (isset($_POST['submit'])) {` penggunaan baris ini adalah untuk kondisi yang akan memeriksa apakah tombol submit pada formulir telah ditekan. Penggunaan `isset($_POST['submit'])` akan memeriksa apakah ada data

yang dikirimkan melalui metode POST dengan nama kunci 'submit'. Sehingga pada baris ini ada pengkondisian untuk melakukan pemeriksaan apakah formulir yang diisikan telah disubmit.

Pada baris [18], syntax `$nilai = $_POST["nilai"]`; pada baris ini merupakan variabel `$nilai` yang diinisialisasi dengan nilai yang dikirimkan melalui input dengan nama "nilai" pada formulir.

Pada baris [19] [20], syntax

```
if ($nilai == 0){  
    echo "<h2>Hasil : Nol</h2>";
```

pada baris ini merupakan kondisi pertama yang mengecek apakah nilai yang dimasukkan adalah 0. Jika iya, maka akan ditampilkan pesan "Hasil : Nol".

Pada baris [21] [22], syntax

```
} elseif ($nilai > 0 && $nilai < 10){  
    echo "<h2>Hasil : Satuan</h2>";
```

pada baris-baris tersebut merupakan kondisi kedua yang akan mengecek apakah nilai yang dimasukkan berada di antara 1 hingga 9 yang termasuk dalam kelompok belasan. Apabila benar, maka akan ditampilkan pesan "Hasil : Belasan".

Pada baris [23] [24], syntax

```
} elseif ($nilai >= 11 && $nilai < 20){  
    echo "<h2>Hasil : Belasan</h2>";
```

pada baris-baris tersebut merupakan kondisi ketiga yang mengecek apakah nilai yang dimasukkan berada di antara 11 hingga 19 yang termasuk dalam kelompok belasan. Apabila benar, maka akan ditampilkan pesan "Hasil : Belasan".

Pada baris [25] [26], syntax

```
} elseif ($nilai >= 20 && $nilai < 100){  
    echo "<h2>Hasil : Puluhan</h2>";
```

pada baris-baris tersebut merupakan kondisi keempat yang mengecek apakah nilai yang dimasukkan berada di antara 10 hingga 100 yang termasuk dalam kelompok puluhan. Apabila benar, maka akan ditampilkan pesan "Hasil : Puluhan".

Pada baris [27] [28], syntax

```
} elseif ($nilai >= 100 && $nilai < 1000){  
    echo "<h2>Hasil : Ratusan</h2>";
```

pada baris-baris tersebut merupakan kondisi kelima yang mengecek apakah nilai yang dimasukkan berada di antara 100 hingga 1000 yang termasuk dalam kelompok Ratusan. Apabila benar, maka akan ditampilkan pesan "Hasil : Ratusan".

pada baris [29] [30] [31], syntax

```
} else{  
    echo "<h2>Hasil : Anda Menginput Melebihi  
Limit Bilangan</h2>";  
}
```

pada baris-baris tersebut merupakan bagian yang menanganinya jika nilai yang dimasukkan melebihi 999. Dalam kasus ini, pesan "Hasil : Anda Menginput Melebihi Limit Bilangan" akan ditampilkan.

Pada baris [32], syntax } merupakan telah selesainya kondisioanl yang dilakukan.

Pada baris [33], syntax ?> digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [34], </body> pada baris ini merupakan tag penutup dari tag pembuka baris 10.

Pada baris [35], </html> baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

MODUL 3:

PERULANGAN DAN KONDISIONAL

SOAL 1

Buatlah sebuah program yang dapat mencetak output tergantung banyaknya jumlah input seperti yang diberikan contoh. Wajib menggunakan perulangan while dalam implementasi.

Contoh Output 1:

Jumlah Peserta :

Peserta ke-1

Peserta ke-2

Peserta ke-3

Peserta ke-4

Contoh Ouput 2:

Jumlah Peserta :

Peserta ke-1

Peserta ke-2

Peserta ke-3

Simpan dengan nama file: PRAK301.php

A. Source Code

Table 10. Source Code Modul 3 Soal 1

1	<!DOCTYPE html>
2	<html lang="en">

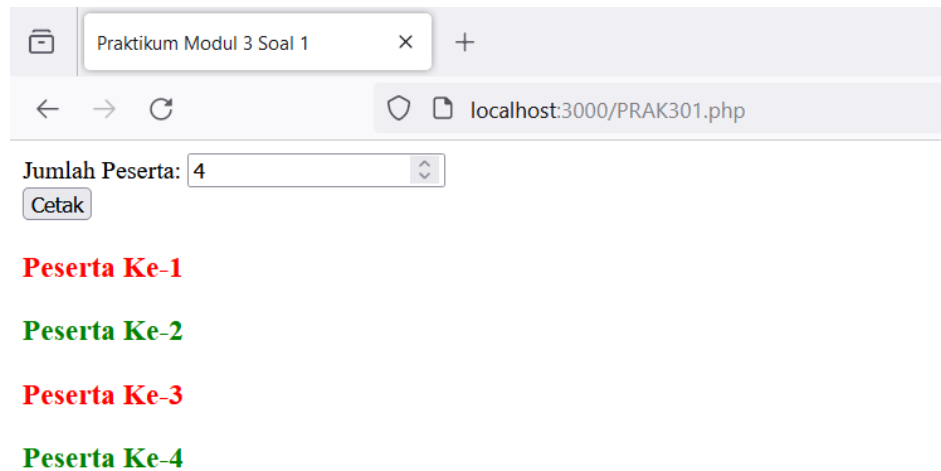
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Praktikum Modul 3 Soal 1</title>
9	</head>
10	
11	<body>
12	<form action="" method="POST">
13	Jumlah Peserta: <input type="number"
	name="nilai" value="<?php echo isset(\$_POST['nilai']) ?
	\$_POST['nilai'] : '' ; ?>">
14	<input type="submit" name="submit"
	value="Cetak">
15	</form>
16	
17	<?php
18	session_start();
19	
20	if (isset(\$_POST['submit'])) {
21	\$nilai = \$_POST['nilai'];
22	\$_SESSION['nilai'] = \$nilai;
23	\$i = 1;
24	while (\$i <= \$nilai) {
25	if (\$i % 2 == 0) {
26	echo "<h3>Peserta
	Ke-\$i</br>";

```

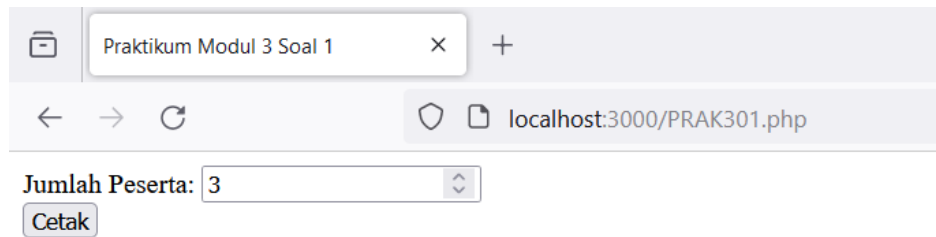
27         } else {
28             echo "<h3><font color='red'>Peserta Ke-
29             $i</font></br>";
30         }
31         $i++;
32     }
33     ?>
34 </body>
35 </html>

```

B. Output Program



Gambar 11. Screenshot Hasil Jawaban Modul 3 Soal 1 (1)



Peserta Ke-1

Peserta Ke-2

Peserta Ke-3

Gambar 12. Screenshot Hasil Jawaban Modul 3 Soal 1 (2)

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset

merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8], syntax `<title>Praktikum Modul 3 Soal 1</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [9], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [11], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [12], syntax `<form action="" method="POST">` merupakan bagian dari markup HTML yang digunakan untuk membuat sebuah formulir dalam halaman web. Dalam kode tersebut, terdapat tag pembuka `<form>` yang mendefinisikan form, lalu terdapat juga atribut `action=""` yang menentukan URL atau path ke mana data

formulir akan dikirimkan untuk diproses. Pada baris ini, nilai atribut action adalah kosong, sehingga data akan dikirimkan kembali ke halaman itu sendiri. Selanjutnya terdapat juga atribut method="POST" berguna untuk menentukan metode HTTP yang digunakan saat mengirimkan data form. Pada program ini metode yang digunakan adalah POST, sehingga data form akan dikirimkan ke server dalam badan permintaan HTTP yang tersembunyi, sehingga data yang sensitive akan aman.

Pada baris [13], syntax Jumlah Peserta: `<input type="number" name="nilai" value="<?php echo isset($_POST['nilai']) ? $_POST['nilai'] : ''; ?>">
` merupakan sebuah input field dalam bentuk bilangan pada sebuah formulir di halaman web. Input field ini memiliki atribut name yang ditetapkan sebagai "nilai", yang akan digunakan untuk mengidentifikasi input field ketika data dikirimkan ke server. Penggunaan atribut value digunakan untuk menetapkan nilai awal atau nilai default untuk input field tersebut. Dalam program ini, nilai defaultnya adalah nilai yang dikirimkan melalui form sebelumnya atau kosong jika tidak ada nilai yang dikirimkan sebelumnya. Penggunaan tag `
` digunakan untuk membuat baris baru setelah input field, sehingga elemen-elemen berikutnya akan ditampilkan di baris baru.

Pada baris [14], syntax `<input type="submit" name="submit" value="Cetak">` berguna sebagai tombol submit dalam sebuah formulir di halaman web. Tombol submit ini memiliki atribut name yang ditetapkan sebagai "submit", yang digunakan untuk mengidentifikasi tombol submit ketika formulir dikirimkan ke server. Nilai dari atribut value yang digunakan adalah "Cetak", yang berarti menentukan teks yang akan ditampilkan pada tombol tersebut di halaman web. Ketika tombol ini ditekan oleh pengguna, data dari formulir yang telah diisi akan dikirimkan ke server untuk diproses.

Pada baris [15], syntax `</form>` baris ini merupakan tag penutup dari baris 12.

Pada baris [17], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [18], syntax `session_start()`; digunakan untuk memulai sesi PHP pada suatu halaman web. Penggunaan kode tersebut memungkinkan server untuk menyimpan informasi pengguna antar permintaan halaman dalam sesi PHP. Informasi ini dapat berupa variabel-variabel yang disimpan dalam `$_SESSION` superglobal array. Dengan menggunakan `session_start()`, server akan memberikan pengenalan unik kepada klien yang digunakan untuk mengidentifikasi sesi tersebut.

Pada baris [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [31], syntax

```
if (isset($_POST['submit'])) {  
  
    $nilai = $_POST['nilai'];  
  
    $_SESSION['nilai'] = $nilai;  
  
    $i = 1;  
  
    while ($i <= $nilai) {  
  
        if ($i % 2 == 0) {  
  
            echo "<h3><font color='green'>Peserta Ke-  
$i</font></br>";  
  
        } else {  
  
            echo "<h3><font color='red'>Peserta Ke-  
$i</font></br>";  
  
        }  
  
        $i++;  
  
    }  
}
```

}

penggunaan baris-baris tersebut merupakan blok logika PHP yang akan dieksekusi ketika tombol submit dengan nama "submit" diklik pada sebuah formulir, yang selanjutnya akan diproses oleh kondisi if (isset(\$_POST['submit'])). Saat kondisi terpenuhi, langkah-langkah berikut dijalankan: nilai yang dimasukkan oleh pengguna melalui formulir disimpan dalam variabel PHP \$nilai, lalu nilai tersebut disimpan dalam variabel sesi PHP dengan nama "nilai" menggunakan \$_SESSION['nilai'], sehingga nilai tersebut diakses di halaman-halaman PHP lainnya. Selanjutnya, terdapat sebuah variabel hitungan berupa \$i diinisialisasi dengan nilai 1, selanjutnya dilakukan perulangan while sebanyak jumlah peserta yang dimasukkan oleh pengguna. Setiap peserta akan dicetak dengan nomor urut yang berbeda dan warna yang bervariasi: nomor urut genap dicetak dengan warna hijau, sementara nomor urut ganjil dicetak dengan warna merah. Seluruh proses ini bertujuan untuk memproses dan menampilkan data peserta sesuai dengan input pengguna, untuk memberikan visualisasi yang berbeda untuk nomor urut peserta yang genap dan ganjil.

Pada baris [33], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [34], syntax `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 11.

Pada baris [35], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 2

Buatlah program yang dapat menampilkan bentuk segitiga dari gambar yang dimasukkan. Input baris pertama adalah tinggi segitiga. Input berikutnya adalah 971amat gambar yang akan digunakan untuk membangun segitiga. Gunakan link berikut sebagai inputan gambar : <https://cdn0.iconfinder.com/data/icons/web-and-mobile-icons-volume-2/128/52-512.png>. Wajib menggunakan perulangan while dalam implementasinya.

Contoh Output 1:

Tinggi :

Alamat Gambar :



Contoh Output 2:

Tinggi :

Alamat Gambar :



Simpan dengan nama file: PRAK302.php

A. Source Code

Table 11. Source Code Modul 3 Soal 2

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Praktikum Modul 3 Soal 2</title>
9	<style>
10	form {
11	margin-bottom: 20px;
12	}
13	</style>
14	</head>
15	
16	<body>
17	<form action="" method="POST">
18	Tinggi : <input type="text" name="tinggi"
	value="<?php echo isset(\$_POST['tinggi']) ?
	\$_POST['tinggi'] : ''; ?>">
19	Alamat Gambar: <input type="text"
	name="alamatgambar" value="<?php echo
	isset(\$_POST['alamatgambar']) ? \$_POST['alamatgambar']
	: ''; ?>">
20	<input type="submit" name="submit"
	value="Cetak">

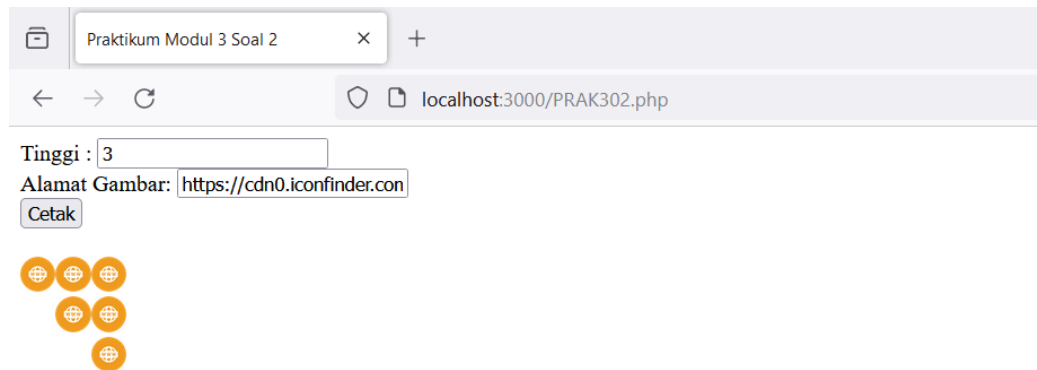
```

21     </form>
22
23     <?php
24         if (isset($_POST['submit'])) {
25             $tinggi = $_POST['tinggi'];
26             $alamatgambar = $_POST['alamatgambar'];
27             $i = 1;
28             $j = 1;
29             $k = $tinggi;
30
31             while ($i <= $tinggi) {
32                 while ($j < $i) {
33                     echo "<img style='width: 25px; opacity:
34 0;' src='$alamatgambar' alt=''>";
35                     $j = $j + 1;
36                 }
37                 while ($k >= $i) {
38                     echo "<img style='width: 25px'
39 src='$alamatgambar' alt=''>";
40                     $k = $k - 1;
41                 }
42                 echo "<br>";
43
44                 $j = 1;
45                 $k = $tinggi;
46                 $i = $i + 1;
47             }
48         }
49     }
50 }
51
52 </body>

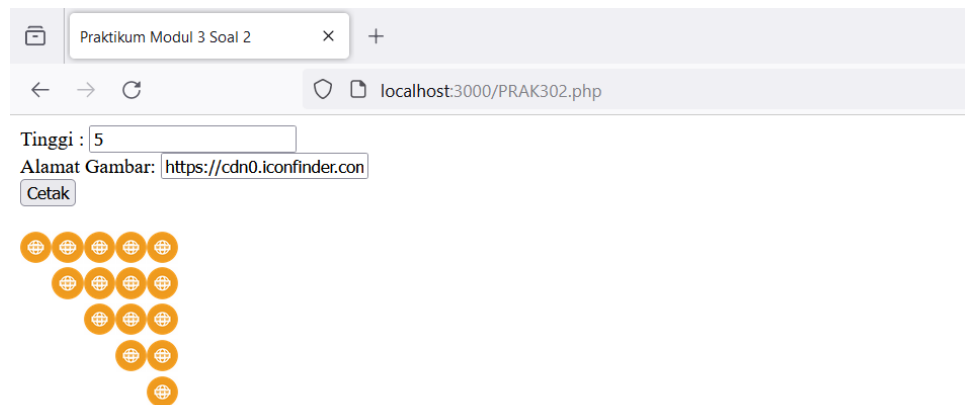
```

49	</html>
----	---------

B. Output Program



Gambar 13. Screenshot Hasil Jawaban Modul 3 Soal 2 (1)



Gambar 14. Screenshot Hasil Jawaban Modul 3 Soal 2 (2)

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8], syntax `<title>Praktikum Modul 3 Soal 1</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag

ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [9] [10] [11] [12] [13], syntax

```
<style>

    form {

        margin-bottom: 20px;

    }

</style>
```

pada baris-baris tersebut merupakan aturan CSS yang akan mendefinisikan gaya untuk elemen `<form>` pada halaman web. Penggunaan CSS tersebut akan mengatur margin bawah dari elemen `<form>` menjadi 20 piksel.

Pada baris [14], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [16], syntax `<body>` tag Body merupakan section utama dalam dokumen web.

Pada baris [17], syntax `<form action="" method="POST">` merupakan bagian dari markup HTML yang digunakan untuk membuat sebuah formulir dalam halaman web. Dalam kode tersebut, terdapat tag pembuka `<form>` yang mendefinisikan form, lalu terdapat juga atribut `action=""` yang menentukan URL atau path ke mana data formulir akan dikirimkan untuk diproses. Pada baris ini, nilai atribut `action` adalah kosong, sehingga data akan dikirimkan kembali ke halaman itu sendiri. Selanjutnya terdapat juga atribut `method="POST"` berguna untuk menentukan metode HTTP yang digunakan saat mengirimkan data form. Pada program ini metode yang digunakan adalah POST, sehingga data form akan dikirimkan ke server dalam badan permintaan HTTP yang tersembunyi, sehingga data yang sensitive akan aman.

Pada baris [18], syntax Tinggi : `<input type="text" name="tinggi" value="<?php echo isset($_POST['tinggi']) ? $_POST['tinggi'] : ''; ?>">
` merupakan sebuah input field dalam bentuk teks pada sebuah formulir di halaman web. Label "Tinggi :" yang berada di depan input field digunakan untuk memberikan keterangan kepada pengguna mengenai informasi yang diharapkan untuk dimasukkan. Input field tersebut memiliki atribut name yang ditetapkan sebagai "tinggi", yang akan digunakan untuk mengidentifikasi input field ketika data dikirimkan ke server. Selanjutnya terdapat atribut value yang digunakan agar menampilkan nilai awal atau nilai default untuk input field tersebut. Nilai ini akan ditampilkan kembali kepada pengguna jika halaman dimuat kembali setelah form dikirimkan, sehingga pengguna dapat melihat kembali nilai yang telah dimasukkan sebelumnya. Tag `
` digunakan untuk membuat baris baru setelah input field, sehingga elemen-elemen berikutnya akan ditampilkan di baris baru.

Pada baris [19], syntax Alamat Gambar: `<input type="textbox" name="alamatgambar" value="<?php echo isset($_POST['alamatgambar']) ? $_POST['alamatgambar'] : ''; ?>">
` penggunaan baris ini adalah sebuah input field dalam bentuk teks pada sebuah formulir di halaman web. Penggunaan label "Alamat Gambar:" ditempatkan di depan input field untuk memberikan keterangan kepada pengguna mengenai informasi yang diharapkan untuk dimasukkan. Sehingga nantinya pengguna bisa memasukan link beupa Alamat gambar yang diinginkan untuk diproses. Input field ini memiliki atribut name yang ditetapkan sebagai "alamatgambar", yang akan digunakan untuk mengidentifikasi input field ketika data dikirimkan ke server. Penggunaan atribut value ditetapkan agar menampilkan nilai awal atau nilai default untuk input field tersebut. Nilai ini akan ditampilkan kembali kepada pengguna jika halaman dimuat kembali setelah form dikirimkan, memungkinkan pengguna untuk melihat kembali nilai yang telah dimasukkan sebelumnya. Tag `
` digunakan untuk membuat baris baru setelah input field, sehingga elemen-elemen berikutnya akan ditampilkan di baris baru.

Pada baris [20], syntax `<input type="submit" name="submit" value="Cetak">` berguna sebagai tombol submit dalam sebuah formulir di halaman web. Tombol submit ini memiliki atribut name yang ditetapkan sebagai "submit", yang digunakan untuk mengidentifikasi tombol submit ketika formulir dikirimkan ke server. Nilai dari atribut value yang digunakan adalah "Cetak", yang berarti menentukan teks yang akan ditampilkan pada tombol tersebut di halaman web. Ketika tombol ini ditekan oleh pengguna, data dari formulir yang telah diisi akan dikirimkan ke server untuk diproses.

Pada baris [21], syntax `</form>` baris ini merupakan tag penutup dari baris 17.

Pada baris [23], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46], syntax

```
if (isset($_POST['submit'])) {  
  
    $tinggi = $_POST['tinggi'];  
  
    $alamatgambar = $_POST['alamatgambar'];  
  
    $i = 1;  
  
    $j = 1;  
  
    $k = $tinggi;  
  
  
    while ($i <= $tinggi) {  
  
        while ($j < $i) {  
  
            echo "<img style='width: 25px; opacity:  
0;' src='$alamatgambar' alt=''>";
```



```

        $j = $j + 1;
    }

    while ($k >= $i) {

        echo "<img      style='width:      25px'
src='$alamatgambar' alt=''>";

        $k = $k - 1;

    }

    echo "<br>";

    $j = 1;

    $k = $tinggi;

    $i = $i + 1;

}

}

```

penggunaan dari kode-kode tersebut merupakan blok logika dalam bahasa PHP yang akan dieksekusi jika tombol submit dengan nama "submit" diklik pada suatu formulir. Logika tersebut akan dimulai dengan memeriksa apakah tombol submit tersebut telah ditekan dan data dari formulir telah dikirimkan melalui metode POST. Apabila kondisi terpenuhi, variabel \$tinggi dan \$alamatgambar akan diinisialisasi dengan nilai yang dikirimkan melalui formulir, yang masing-masing merepresentasikan tinggi segitiga dan alamat gambar. Selanjutnya, tiga variabel penghitung, yaitu \$i, \$j, dan \$k, diatur sesuai dengan kebutuhan untuk pembentukan pola gambar segitiga. Proses pembentukan segitiga akan dilakukan dengan menggunakan dua perulangan while. Perulangan pertama, perulangan akan berjalan sebanyak nilai \$tinggi kali, yang mengatur pembentukan setiap baris dalam segitiga. Di dalamnya, terdapat dua

perulangan bersarang. Perulangan pertama digunakan untuk mencetak gambar dengan atribut `opacity: 0;`, yang membuatnya tidak terlihat, sedangkan perulangan kedua mencetak gambar yang terlihat. Dengan menggunakannya maka akan menciptakan pola segitiga dengan cara mengatur jumlah gambar yang dicetak pada setiap baris. Setelah satu baris gambar tercetak, digunakanlah tag line break `
` untuk memindahkan proses pencetakan ke baris berikutnya. Selanjutnya, pada akhir setiap iterasi, nilai variabel `$j` diatur kembali menjadi 1, nilai variabel `$k` diatur kembali menjadi `$tinggi`, dan nilai variabel `$i` diinkrementasi. Menggunakan langkah-langkah tersebut untuk memastikan bahwa perulangan dapat berjalan dengan benar dalam mencetak pola segitiga yang diinginkan.

Pada baris [47], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [48], syntax `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 16.

Pada baris [49], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 3

Buatlah sebuah program untuk mencek suatu bilangan pada deret, jika bilangan tersebut ditambah 7 apakah merupakan bilangan kelipatan 5. Input pertama adalah batas bawah deret. Input baris kedua adalah batas atas deret bilangan. Output adalah menampilkan deret bilangan dan bilangan yang diminta diganti dengan gambar bintang. Wajib menggunakan perulangan do while dalam implementasi. Source gambar bisa didownload disini: <https://www.freepnglogos.com/images/star-images-9441.html>

Contoh Output 1:

Batas Bawah :
Batas Atas :

1 2 ★ 4 5 6 7 ★ 9 10

Contoh Output 2:

Batas Bawah :
Batas Atas :

37 ★ 39 40 41 42 ★ 44 45 46 47 ★ 49 50 51 52 ★ 54 55
Simpan dengan nama file: PRAK303.php

A. Source Code

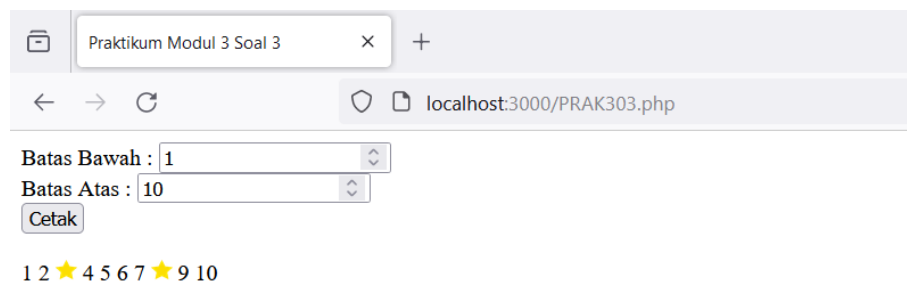
Table 12. Source Code Modul 3 Soal 3

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">

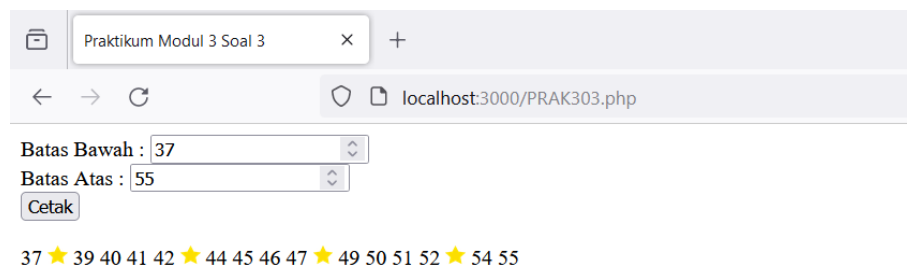
7	<code><meta name="viewport" content="width=device-width,</code>
8	<code>initial-scale=1.0"></code>
9	<code><title>Praktikum Modul 3 Soal 3</title></code>
10	<code></head></code>
11	
12	<code><body></code>
13	<code><form action="" method="POST"></code>
	<code> Batas Bawah : <input type="number" name="bawah"</code>
	<code>value="<?php echo isset(\$_POST['bawah']) ?</code>
14	<code>\$_POST['bawah'] : ''; ?>"></br></code>
	<code> Batas Atas : <input type="number" name="atas"</code>
	<code>value="<?php echo isset(\$_POST['atas']) ?</code>
15	<code>\$_POST['atas'] : ''; ?>"></br></code>
	<code> <input type="submit" name="submit"</code>
16	<code>value="Cetak"></br></code>
17	<code></form></br></code>
18	
19	<code><?php</code>
20	<code>if (isset(\$_POST['submit'])) {</code>
21	<code> \$bawah = \$_POST['bawah'];</code>
22	<code> \$atas = \$_POST['atas'];</code>
23	<code> \$i = \$bawah;</code>
24	<code> do {</code>
25	<code> if ((\$i + 7) % 5 == 0) {</code>
	<code> echo "<img src='Gambar/Bintang.png'"</code>
26	<code>width='15px' height='15px'>";</code>
27	<code> } else {</code>
28	<code> echo \$i;</code>
29	<code> }</code>
30	<code> echo "&nbsp;";</code>

31	<code>\$i++;</code>
32	<code>} while (\$i <= \$atas);</code>
33	<code>}</code>
34	<code>?></code>
35	<code></body></code>
	<code></html></code>

B. Output Program



Gambar 15. Screenshot Hasil Jawaban Modul 3 Soal 3 (1)



Gambar 16. Screenshot Hasil Jawaban Modul 3 Soal 3 (2)

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda

awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8], syntax `<title>Praktikum Modul 3 Soal 3</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau

teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [9], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [11], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [12], syntax `<form action="" method="POST">` merupakan bagian dari markup HTML yang digunakan untuk membuat sebuah formulir dalam halaman web. Dalam kode tersebut, terdapat tag pembuka `<form>` yang mendefinisikan form, lalu terdapat juga atribut `action=""` yang menentukan URL atau path ke mana data formulir akan dikirimkan untuk diproses. Pada baris ini, nilai atribut `action` adalah kosong, sehingga data akan dikirimkan kembali ke halaman itu sendiri. Selanjutnya terdapat juga atribut `method="POST"` berguna untuk menentukan metode HTTP yang digunakan saat mengirimkan data form. Pada program ini metode yang digunakan adalah POST, sehingga data form akan dikirimkan ke server dalam badan permintaan HTTP yang tersembunyi, sehingga data yang sensitive akan aman.

Pada baris [13], syntax Batas Bawah : `<input type="number" name="bawah" value="<?php echo isset($_POST['bawah']) ? $_POST['bawah'] : '' ; ?>"></br>` penggunaan baris ini adalah bagian dari formulir HTML yang bertujuan untuk memungkinkan pengguna memasukkan nilai numerik sebagai "Batas Bawah". Label "Batas Bawah :" yang ditempatkan di depan input field memberikan keterangan kepada pengguna tentang informasi yang diharapkan untuk dimasukkan. Input field ini diatur dengan tipe number, yang membatasi masukan hanya untuk nilai numerik, sehingga pengguna hanya dapat memasukkan angka. Penggunaan atribut `name` menetapkan nama atau identifikasi untuk input field ini, yang nantinya akan digunakan untuk mengidentifikasi input field ketika data dikirimkan ke server. Atribut `value` ditetapkan untuk menampilkan nilai awal atau default dalam input field tersebut. Dalam hal ini, nilai defaultnya adalah nilai

yang dikirimkan melalui form sebelumnya atau kosong jika tidak ada nilai yang dikirimkan sebelumnya. Penggunaan tag `
` digunakan untuk membuat baris baru setelah input field, sehingga elemen-elemen berikutnya akan ditampilkan di baris baru.

Pada baris [14], syntax Batas Atas : `<input type="number" name="atas" value="<?php echo isset($_POST['atas']) ? $_POST['atas'] : ''; ?>"></br>` penggunaan baris ini adalah bagian dari formulir HTML yang bertujuan untuk memungkinkan pengguna memasukkan nilai numerik sebagai "Batas Atas". Label "Batas atas :" yang ditempatkan di depan input field memberikan keterangan kepada pengguna tentang informasi yang diharapkan untuk dimasukkan. Input field ini diatur dengan tipe number, yang membatasi masukan hanya untuk nilai numerik, sehingga pengguna hanya dapat memasukkan angka. Penggunaan atribut name menetapkan nama atau identifikasi untuk input field ini, yang nantinya akan digunakan untuk mengidentifikasi input field ketika data dikirimkan ke server. Atribut value ditetapkan untuk menampilkan nilai awal atau default dalam input field tersebut. Dalam hal ini, nilai defaultnya adalah nilai yang dikirimkan melalui form sebelumnya atau kosong jika tidak ada nilai yang dikirimkan sebelumnya. Penggunaan tag `
` digunakan untuk membuat baris baru setelah input field, sehingga elemen-elemen berikutnya akan ditampilkan di baris baru.

Pada baris [15], syntax `<input type="submit" name="submit" value="Cetak">` berguna sebagai tombol submit dalam sebuah formulir di halaman web. Tombol submit ini memiliki atribut name yang ditetapkan sebagai "submit", yang digunakan untuk mengidentifikasi tombol submit ketika formulir dikirimkan ke server. Nilai dari atribut value yang digunakan adalah "Cetak", yang berarti menentukan teks yang akan ditampilkan pada tombol tersebut di halaman web. Ketika tombol ini ditekan oleh pengguna, data dari formulir yang telah diisi akan dikirimkan ke server untuk diproses.

Pada baris [16], syntax `</form></br>` baris ini merupakan tag penutup dari baris 12. Selanjutnya pada baris ini terdapat juga penggunaan tag `
` digunakan untuk

membuat baris baru setelah input field, sehingga elemen-elemen berikutnya akan ditampilkan di baris baru.

Pada baris [18], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32], syntax

```
if (isset($_POST['submit'])) {  
    $bawah = $_POST['bawah'];  
    $atas = $_POST['atas'];  
    $i = $bawah;  
    do {  
        if (($i + 7) % 5 == 0) {  
            echo "<img src='Gambar/Bintang.png'  
width='15px' height='15px'>";  
        } else {  
            echo $i;  
        }  
        echo "&nbsp;";  
        $i++;  
    } while ($i <= $atas);  
}
```

penggunaan kode-kode tersebut merupakan merupakan bagian dari blok logika PHP yang dieksekusi jika tombol submit dengan nama "submit" ditekan pada formulir yang

terkait. Langkah pertama adalah kode akan memeriksa apakah tombol submit telah ditekan dan data dari formulir telah dikirimkan melalui metode POST. Jika kondisi terpenuhi, nilai "Batas Bawah" dan "Batas Atas" yang dimasukkan oleh pengguna melalui formulir akan diambil dan disimpan dalam variabel \$bawah dan \$atas. Selanjutnya, variabel \$i diinisialisasi dengan nilai "Batas Bawah" untuk digunakan sebagai penghitung dalam perulangan. Kode kemudian menggunakan konstruksi perulangan do-while untuk melakukan iterasi dari "Batas Bawah" hingga "Batas Atas". Setiap iterasi perulangan akan memeriksa apakah nilai \$i ditambah 7 menghasilkan sisa bagi 5 yang sama dengan 0. Apabila benar maka gambar bintang akan dicetak, namun apabila salah maka nilai angka \$i akan dicetak. Dalam setiap iterasi, spasi non-breaking juga dicetak untuk memberikan jarak antara nilai-nilai yang dicetak. Sehingga, nilai \$i diinkrementasi pada setiap iterasi untuk memindahkan perulangan ke nilai selanjutnya.

Pada baris [33], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [34], syntax `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 16.

Pada baris [35], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 4

Buatlah sebuah program untuk menampilkan banyak bintang sesuai dengan jumlah bintang pada saat diinput. Output program berupa banyak gambar bintang sesuai input serta tombol “Tambah” dan “Kurang”. Saat tombol “Tambah” ditekan maka jumlah bintang akan bertambah 1, apabila tombol “Kurang” ditekan maka jumlah bintang akan berkurang 1. Source gambar bisa didownload disini: <https://www.freepnglogos.com/images/star-images-9441.html>

Contoh Output 1 (Sebelum disubmit):

Jumlah bintang

Contoh Output 2 (Sesudah disubmit):

Jumlah bintang 5



Contoh Output 2 (Ketika tombol “**Tambah**” ditekan):

Jumlah bintang 6



Simpan dengan nama file: PRAK304.php

A. Source Code

Table 13. Source Code Modul 3 Soal 4

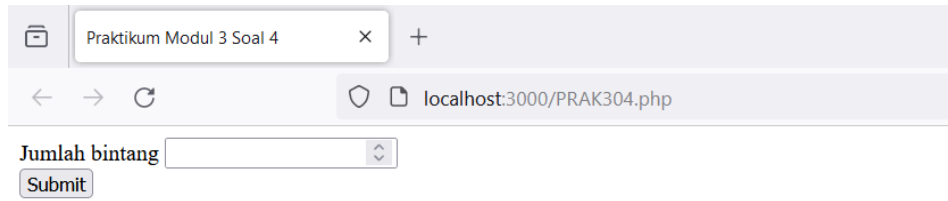
1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Praktikum Modul 3 Soal 4</title>
9	</head>
10	
11	<body>
12	<?php
13	if (\$_SERVER["REQUEST_METHOD"] == "POST") {
14	\$jumlah = \$_POST["jumlah"];
15	}
16	
17	if (isset(\$_POST['tambah'])) {
18	\$jumlah += 1;
19	}
20	
21	if (isset(\$_POST['kurang'])) {
22	\$jumlah -= 1;
23	}
24	
25	if(empty(\$jumlah)): ?>
26	<form action="" method="post">

```

27         Jumlah bintang <input type="number"
name="jumlah" min="1">
28         <br>
29         <button type="submit"
name="submit">Submit</button>
30     </form>
31     <?php
32     endif;
33     if(!empty($jumlah)):
34     ?>
35         <?php
36         echo "Jumlah bintang $jumlah<br><br><br>";
37         for ($i=0; $i < $jumlah; $i++) {
38             echo '';
39         }
40     ?>
41     <form action="" method="post">
42         <input type="number" name="jumlah" min="1"
value="<?= $jumlah ?>" hidden>
43         <button type="submit"
name="tambah">Tambah</button>
44         <button type="submit"
name="kurang">Kurang</button>
45     </form>
46     <?php endif; ?>
47 </body>
48 </html>

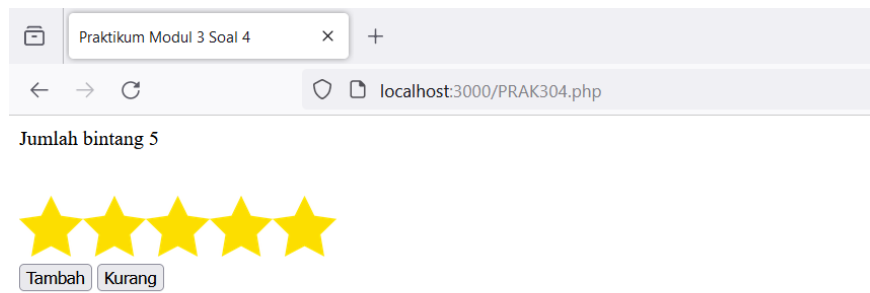
```

B. Output Program



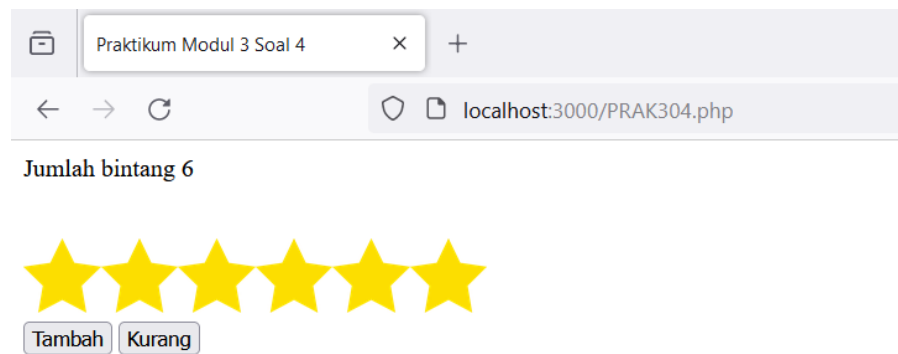
A screenshot of a web browser window. The address bar shows 'localhost:3000/PRAK304.php'. Below the address bar, there is a label 'Jumlah bintang' followed by a text input field and a 'Submit' button.

Gambar 17. Screenshot Hasil Jawaban Modul 3 Soal 4 (1)



A screenshot of a web browser window. The address bar shows 'localhost:3000/PRAK304.php'. Below the address bar, there is a label 'Jumlah bintang 5' followed by five yellow stars. Below the stars are two buttons: 'Tambah' and 'Kurang'.

Gambar 18. Screenshot Hasil Jawaban Modul 3 Soal 4 (2)



A screenshot of a web browser window. The address bar shows 'localhost:3000/PRAK304.php'. Below the address bar, there is a label 'Jumlah bintang 6' followed by six yellow stars. Below the stars are two buttons: 'Tambah' and 'Kurang'.

Gambar 19. Screenshot Hasil Jawaban Modul 3 Soal 4 (3)

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol

bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8], syntax `<title>Praktikum Modul 3 Soal 4</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [9], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [11], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [12], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [13] [14] [15], syntax

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $jumlah = $_POST["jumlah"];  
}
```

penggunaan baris tersebut merupakan bagian dari logika PHP yang memiliki peran untuk mengambil nilai yang dikirim melalui metode POST dari formulir HTML. Dengan menggunakan kondisi `$_SERVER["REQUEST_METHOD"] == "POST"`, kode tersebut akan memeriksa apakah permintaan halaman ini adalah POST request, yang mengindikasikan bahwa data telah dikirim melalui formulir HTML menggunakan metode POST. Apabila kondisi tersebut terpenuhi, langkah selanjutnya adalah mengambil nilai dari input field dengan atribut name bernilai "jumlah", yang telah

dikirim melalui formulir. Nilai tersebut kemudian akan disimpan dalam variabel \$jumlah, sehingga dapat digunakan dalam pemrosesan lanjutan.

Pada baris [17] [18] [19], syntax

```
if (isset($_POST['tambah'])) {  
    $jumlah += 1;  
}
```

penggunaan baris tersebut merupakan proses penambahan nilai variabel \$jumlah jika tombol "tambah" pada formulir HTML telah ditekan. Pertama, melalui kondisi `isset($_POST['tambah'])`, kode akan memeriksa apakah tombol tersebut telah ditekan, dengan menggunakan atribut name bernilai "tambah" sebagai penandanya. Apabila tombol "tambah" telah ditekan, nilai variabel \$jumlah akan ditambah satu menggunakan operator penambahan yaitu `+=`. Dengan adanya hal tersebut memungkinkan pengguna untuk menambah jumlah yang diinginkan melalui formulir, dan nilai \$jumlah akan diperbarui sesuai dengan jumlah yang ditambahkan.

Pada baris [21] [22] [23], syntax

```
if (isset($_POST['kurang'])) {  
    $jumlah -= 1;  
}
```

penggunaan kode tersebut merupakan proses untuk mengurangi nilai variabel \$jumlah jika tombol dengan atribut name bernilai "kurang" telah ditekan pada formulir HTML. Hal tersebut memungkinkan pengguna untuk mengurangi jumlah yang diinginkan melalui formulir, memberikan fleksibilitas atau kemudahan dalam mengatur nilai variabel tersebut. Dalam prosesnya, pertama melalui kondisi `isset($_POST['kurang'])`, kode akan memeriksa apakah tombol tersebut telah ditekan, dengan menggunakan atribut name bernilai "kurang" sebagai penandanya. Apabila tombol "kurang" telah ditekan, nilai variabel \$jumlah akan dikurangi satu menggunakan operator

pengurangan yaitu -= 1. Sehingga setiap kali tombol "kurang" ditekan, maka nilai \$jumlah akan berkurang satu poin.

Pada baris [24] [25] [26] [27] [28] [29] [30], syntax

```
if(empty($jumlah)): ?>

    <form action="" method="post">

        Jumlah      bintang      <input      type="number"
name="jumlah" min="1">

        <br>

        <button                                type="submit"
name="submit">Submit</button>

    </form>
```

penggunaan kode-kode tersebut bertujuan untuk menampilkan formulir HTML yang meminta pengguna untuk memasukkan jumlah bintang jika nilai variabel \$jumlah kosong atau belum diinisialisasi. Saat halaman dimuat, kondisi if(empty(\$jumlah)) akan memeriksa apakah variabel \$jumlah memiliki nilai atau masih kosong. Jika variabel \$jumlah kosong, maka blok kode di dalamnya akan dieksekusi. Dalam blok tersebut, sebuah formulir HTML dibuat menggunakan tag <form>. Atribut action dan method pada tag formulir menunjukkan bahwa data akan dikirimkan kembali ke halaman itu sendiri dengan metode POST. Formulir akan meminta pengguna untuk memasukkan jumlah bintang melalui input field dengan tipe bilangan. Batas minimalnya ditetapkan pada 1 untuk memastikan bahwa jumlah bintang yang dimasukkan tidak boleh kurang dari itu. Pada kode-kode tersebut terdapat juga tombol "Submit" yang dimana saat ditekan, akan mengirimkan data formulir ke halaman itu sendiri untuk diproses.

Pada baris [31] [32] [33] [34], syntax

```
<?php
```

```
endif;

if(!empty($jumlah)):

?>
```

penggunaan kode tersebut bertujuan untuk menampilkan atau menyembunyikan bagian-bagian tertentu berdasarkan nilai variabel atau kondisi yang ditentukan. Penggunaan kode `<?php` dan `?>` digunakan untuk menandai awal dan akhir dari blok logika PHP di dalam sebuah dokumen HTML. Penggunaan `endif;` berfungsi sebagai penutup dari sebuah blok `if` yang dibuka sebelumnya dengan `if(empty($jumlah));`, yang menandakan akhir dari struktur kendali `if`. Selanjutnya digunakan `if(!empty($jumlah)):` `?>` untuk membuka sebuah blok `if` baru yang menguji apakah variabel `$jumlah` tidak kosong. Jika kondisi ini terpenuhi, blok kode yang terletak di antara `if(!empty($jumlah)):` dan `endif;` (yang ditandai dengan `?>`) akan dieksekusi.

Pada baris [35] [36] [37] [38] [39] [40], syntax

```
<?php

    echo "Jumlah bintang $jumlah<br><br><br>";

    for ($i=0; $i < $jumlah; $i++) {

        echo      '';

    }

?>
```

penggunaan baris tersebut memiliki fungsi untuk menampilkan serangkaian bintang sesuai dengan jumlah yang ditentukan oleh variabel `$jumlah`. Langkah pertama adalah dengan menggunakan pernyataan `echo`, teks "Jumlah bintang" akan ditampilkan diikuti oleh nilai dari variabel `$jumlah`, yang akan memberikan informasi kepada pengguna tentang berapa jumlah bintang yang akan ditampilkan. Selanjutnya, loop `for` digunakan untuk membuat iterasi sebanyak nilai yang ditentukan oleh variabel `$jumlah`. Setiap

iterasi akan menampilkan sebuah gambar bintang menggunakan pernyataan echo, dengan sumber gambar yang telah ditentukan pada atribut src berupa file gambar "Bintang.png". Selanjutnya pada baris tersebut lebar dan tinggi gambar bintang ditetapkan masing-masing sebesar 50 piksel menggunakan atribut width dan height.

Pada baris [41] [42] [43] [44] [45], syntax

```
<form action="" method="post">

    <input type="number" name="jumlah" min="1"
value="<?= $jumlah ?>" hidden>

    <button type="submit"
name="tambah">Tambah</button>

    <button type="submit"
name="kurang">Kurang</button>

</form>
```

penggunaan baris tersebut merupakan bagian dari formulir HTML yang digunakan untuk mengontrol tindakan pengguna terhadap jumlah bintang yang ditampilkan. Pada formulir ini memiliki dua tombol, yaitu "Tambah" dan "Kurang", serta satu input field tersembunyi. Input field tersebut tersembunyi dari tampilan pengguna dan memiliki tipe number dengan nama "jumlah". Pada program ini nilai minimal yang diperbolehkan untuk dimasukkan adalah 1, sehingga memastikan bahwa jumlah bintang tidak akan kurang dari satu. Selanjutnya nilai input field diatur menggunakan variabel PHP \$jumlah, yang berfungsi sebagai nilai default. Tombol "Tambah" dan "Kurang" masing-masing memiliki tugas sebagai penanda aksi yang diambil oleh pengguna. Ketika tombol "Tambah" ditekan maka data formulir akan dikirimkan ke halaman yang ditentukan dalam atribut action menggunakan metode POST, dan nilai "tambah" akan dikirimkan bersama dengan data. Hal tersebut juga berlaku pada tombol "Kurang".

Pada baris [46], syntax `<?php endif; ?>` merupakan bagian dari struktur kendali alternatif dalam PHP yang digunakan untuk menutup blok if yang dibuka sebelumnya. Dalam struktur kendali alternatif, seperti yang terlihat dalam kode tersebut, blok if, elseif, dan else ditutup dengan `endif;`, `elseif,` dan `endif;` berturut-turut. Pengguna tersebut membantu dalam meningkatkan kejelasan dan keterbacaan kode, terutama saat ada banyak kondisi yang dilakukan.

Pada baris [47], `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 11.

Pada baris [48], `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 5

Buatlah sebuah program yang menerima masukan string kemudian mencetak string tersebut dengan masing-masing karakter pada string tercetak sebanyak panjang dari string masukan. Pada masing-masing karakter, karakter pertama harus tercetak kapital sedangkan sisanya harus tidak kapital.

Contoh program :

Contoh 1

Input:

BanDunG

Output:

BbbbbbbAaaaaaNnnnnnnDddddddUuuuuuuNnnnnnnGgggggg

Contoh 2

Input:

BANJAr

Output:

BbbbbbbAaaaaaNnnnnnJjjjjAaaaaRrrrr

Simpan dengan nama file: PRAK305.php

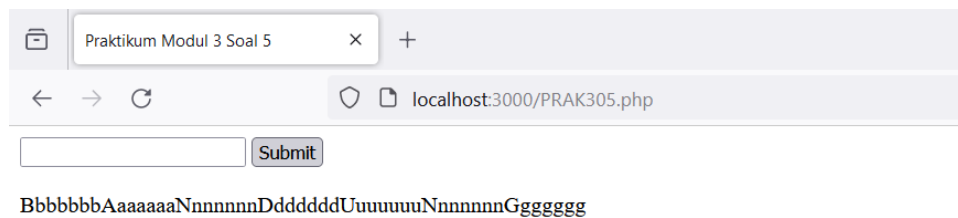
A. Source Code

Table 14. Source Code Modul 3 Soal 5

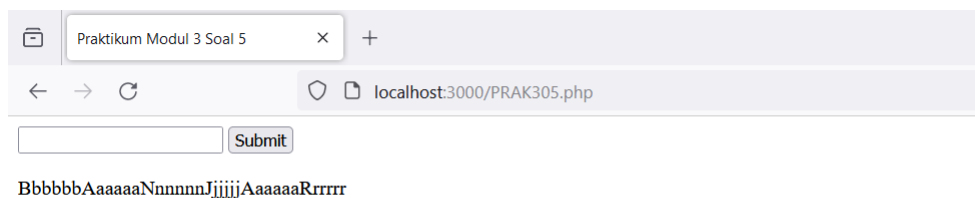
1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Praktikum Modul 3 Soal 5</title>
9	</head>
10	
11	<body>
12	<form action="" method="post">
13	<input type="text" name="teks" required>
14	<button type="submit"
	name="submit">Submit</button>
15	</form>
16	
17	
18	<?php
19	if (isset(\$_POST['submit'])) {
20	\$teks = \$_POST['teks'];
21	\$panjangteks = strlen(\$teks);
22	\$inputteks = str_split(\$teks);
23	\$j = 0;
24	\$k = 0;
25	while (\$k < \$panjangteks) {

26	echo strtoupper(\$inputteks[\$j]);
27	for (\$i = 1; \$i < \$panjangteks; \$i++) {
28	echo strtolower(\$inputteks[\$j]);
29	}
30	\$k++;
31	\$j++;
32	}
33	}
34	?>
35	</body>
36	</html>

B. Output Program



Gambar 20. Screenshot Hasil Jawaban Modul 3 Soal 5 (1)



Gambar 21. Screenshot Hasil Jawaban Modul 3 Soal 5 (2)

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol

bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8], syntax `<title>Praktikum Modul 3 Soal 5</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [9], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [11], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [12], syntax `<form action="" method="POST">` merupakan bagian dari markup HTML yang digunakan untuk membuat sebuah formulir dalam halaman web. Dalam kode tersebut, terdapat tag pembuka `<form>` yang mendefinisikan form, lalu terdapat juga atribut `action=""` yang menentukan URL atau path ke mana data formulir akan dikirimkan untuk diproses. Pada baris ini, nilai atribut `action` adalah kosong, sehingga data akan dikirimkan kembali ke halaman itu sendiri. Selanjutnya terdapat juga atribut `method="POST"` berguna untuk menentukan metode HTTP yang digunakan saat mengirimkan data form. Pada program ini metode yang digunakan adalah POST, sehingga data form akan dikirimkan ke server dalam badan permintaan HTTP yang tersembunyi, sehingga data yang sensitive akan aman.

Pada baris [13], syntax `<input type="text" name="teks" required>` kode tersebut merupakan sebuah input field dalam formulir HTML. Atribut `type="text"` menetapkan jenis input field menjadi teks, yang memungkinkan pengguna untuk memasukkan teks atau karakter apapun ke dalamnya. Dengan memberikan atribut `name="teks"`, sebuah label atau nama ditetapkan untuk mengidentifikasi nilai yang dimasukkan ke dalam input field ini. Ketika formulir disubmit, nilai dari input field ini akan dikirimkan bersama dengan nama "teks", sehingga memungkinkan server untuk

mengidentifikasi dan memproses nilai tersebut. Penggunaan atribut `required` menandakan bahwa input field ini harus diisi sebelum formulir dapat disubmit. Jika input field ini dibiarkan kosong saat formulir disubmit, browser akan menampilkan pesan kesalahan dan mencegah pengguna untuk melanjutkan proses pengiriman formulir, memastikan bahwa input yang diperlukan tidak terlewatkan.

Pada baris [14], syntax `<button type="submit" name="submit">Submit</button>` digunakan untuk membuat sebuah tombol submit dalam formulir HTML. Penggunaan atribut `type="submit"` menentukan bahwa tombol ini akan berfungsi sebagai tombol submit, yang berarti ketika tombol ini ditekan, data dari formulir akan dikirimkan ke server. Dengan menetapkan atribut `name="submit"`, tombol ini diberi label "submit", yang akan dikirimkan bersama dengan data formulir saat formulir tersebut disubmit. Sehingga memungkinkan server untuk mengidentifikasi tombol mana yang telah ditekan oleh pengguna, maka selanjutnya server dapat melakukan aksi yang sesuai, seperti memproses data atau mengarahkan pengguna ke halaman berikutnya.

Pada baris [15], syntax `</form>` merupakan tag penutup dari baris 12.

Pada baris [17], syntax `
` penggunaan tag ini digunakan untuk membuat baris baru setelah input field, sehingga elemen-elemen berikutnya akan ditampilkan di baris baru.

Pada baris [18], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33], syntax

```
if (isset($_POST['submit'])) {  
    $teks = $_POST['teks'];  
    $panjangteks = strlen($teks);
```

```

$inputteks = str_split($teks);

$j = 0;

$k = 0;

while ($k < $panjangteks) {

    echo strtoupper($inputteks[$j]);

    for ($i = 1; $i < $panjangteks; $i++) {

        echo strtolower($inputteks[$j]);

    }

    $k++;

    $j++;

}

}

```

penggunaan kode pada baris-baris tersebut merupakan bagian dari skrip PHP yang bertanggung jawab untuk memproses data yang dikirim melalui formulir HTML saat formulir tersebut disubmit. Terdapat penggunaan `if (isset($_POST['submit'])) { ... }`: dimana fungsi baris ini memeriksa apakah tombol submit dengan nama "submit" telah ditekan pada formulir. Apabila tombol tersebut telah ditekan, nilai yang dimasukkan oleh pengguna ke dalam input field dengan nama "teks" diambil dan disimpan dalam variabel `$teks`. Selanjutnya, panjang teks tersebut dihitung menggunakan fungsi `strlen()` dan teks tersebut dipecah menjadi array karakter menggunakan `str_split()`. Menggunakan dua variabel penghitung yaitu `$j` dan `$k`, diinisialisasi untuk digunakan dalam loop. Selanjutnya, sebuah loop `while` dimulai yang akan dijalankan selama `$k` kurang dari panjang teks. Pada setiap iterasi, karakter teks dari array `$inputteks` pada indeks `$j` dicetak dalam bentuk huruf kapital dengan fungsi `strtoupper()`, diikuti oleh loop `for` yang mencetak karakter teks dalam bentuk huruf kecil dengan fungsi

`strtolower()`. Iterasi loop diarahkan ke karakter teks berikutnya dengan menaikkan nilai dari variabel `$k` dan `$j`. Sehingga penggunaan kode-kode ini bertujuan untuk mengubah teks yang dimasukkan oleh pengguna menjadi urutan yang terdiri dari huruf kapital diikuti oleh huruf kecil.

Pada baris [34], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP.

Pada baris [35], `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 26.

Pada baris [36], `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

MODUL 4:

ARRAY

SOAL 1

Buatlah sebuah program yang dapat mencetak output berupa matriks tergantung banyaknya jumlah input seperti yang diberikan contoh. Input pertama adalah Panjang dari matriks, input kedua adalah lebar dari matriks, input ketiga adalah nilai dari matriks dipisahkan spasi.

Contoh Output 1:

Panjang :

Lebar :

Nilai :

34	56	41
45	36	37
51	32	46

Contoh Ouput 2:

Panjang :

Lebar :

Nilai :

Panjang nilai tidak sesuai dengan ukuran matriks

Simpan dengan nama file: PRAK401.php

A. Source Code

Table 15. Source Code Modul 4 Soal 1

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Praktikum Modul 4 Soal 1</title>
9	<style>
10	table, tr, td {
11	border: solid 1px black;
12	border-collapse: collapse;
13	padding: 10px;
14	text-align: center;
15	}
16	form {
17	margin-bottom: 20px;
18	}
19	</style>
20	</head>
21	
22	<body>
23	<form action="" method="post">
24	Panjang: <input type="text" name="panjang"
	value="<?php if(isset(\$_POST['cetak'])) {echo
	\$_POST['panjang'];}?">


```

50         }
51
52         echo "<table>";
53         for ($i = 0; $i < $panjang; $i++) {
54             echo "<tr>";
55             for ($j = 0; $j < $lebar; $j++) {
56                 echo "<td>" . $stampilnilai[$i][$j] .
57                 "</td>";
58             }
59             echo "</tr>";
60         }
61         echo "</table>";
62     } else {
63         echo "Panjang nilai tidak sesuai dengan
64         ukuran matriks";
65     }
66 }
67 ?>
</body>
</html>

```

B. Output Program

The screenshot shows a web browser window with the title 'Praktikum Modul 4 Soal 1'. The address bar shows 'localhost:3000/PRAK401.php'. The form contains three input fields: 'Panjang:' with value '3', 'Lebar:' with value '3', and 'Nilai:' with value '34 56 41 45 36 37 51 32 4'. Below these is a 'Cetak' button. At the bottom, there is a 3x3 grid of numbers:

34	56	41
45	36	37
51	32	46

Gambar 22. Screenshot Hasil Jawaban Modul 4 Soal 1

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset

merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8], syntax `<title>Praktikum Modul 4 Soal 1</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19], syntax

```
<style>
```

```
    table, tr, td {  
        border: solid 1px black;  
        border-collapse: collapse;  
        padding: 10px;
```

```

        text-align: center;

    }

    form {

        margin-bottom: 20px;

    }

</style>

```

penggunaan baris-baris tersebut merupakan aturan CSS yang akan mendefinisikan gaya untuk elemen `<form>` pada halaman web. Pada baris tersebut akan mengatur properti-properti `border`, `padding`, dan `text-align` ditetapkan untuk elemen-elemen `<table>`, `<tr>`, dan `<td>`, yang digunakan untuk pembuatan table, baris, dan sel dalam halaman. Dengan menetapkan `border` solid dengan ketebalan 1px dan warna hitam, maka garis pinggir akan jelas serta terdefinisi yang diberikan pada tabel dan sel-selnya. Selanjutnya pada baris tersebut terdapat pengaturan `border-collapse: collapse` akan menggabungkan batas sel-sel yang berdekatan dalam tabel, sehingga memberikan tampilan yang lebih rapi. Penambahan `padding` sebesar 10px akan memberikan ruang tambahan di sekitar isi sel, sementara `text-align: center` digunakan untuk membuat teks di dalam sel-sel menjadi terpusat secara horizontal. Selain itu, terdapat juga selector `form` yang digunakan untuk menetapkan margin bawah sebesar 20px pada formulir, untuk menciptakan jarak visual yang memisahkan formulir dari elemen-elemen di bawahnya, dan juga memperbaiki tata letak secara keseluruhan.

Pada baris [20], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [22], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [23], syntax `<form action="" method="POST">` merupakan bagian dari markup HTML yang digunakan untuk membuat sebuah formulir dalam halaman

web. Dalam kode tersebut, terdapat tag pembuka <form> yang mendefinisikan form, lalu terdapat juga atribut action="" yang menentukan URL atau path ke mana data formulir akan dikirimkan untuk diproses. Pada baris ini, nilai atribut action adalah kosong, sehingga data akan dikirimkan kembali ke halaman itu sendiri. Selanjutnya terdapat juga atribut method="POST" berguna untuk menentukan metode HTTP yang digunakan saat mengirimkan data form. Pada program ini metode yang digunakan adalah POST, sehingga data form akan dikirimkan ke server dalam badan permintaan HTTP yang tersembunyi, sehingga data yang sensitive akan aman.

Pada baris [24], syntax Panjang: `<input type="text" name="panjang" value="<?php if(isset($_POST['cetak'])) {echo $_POST['panjang'];}?">
` penggunaannya pada baris merupakan penerima input nilai panjang matriks dari pengguna. Penggunaan elemen input HTML ini akan menampilkan sebuah kotak input teks. Penggunaan atribut type="text" menunjukkan bahwa input ini dapat menerima teks. Nama input ditentukan oleh atribut name="panjang", yang akan digunakan untuk mengidentifikasi nilai yang dimasukkan saat formulir disubmit. Pada atribut value, kode PHP disisipkan untuk menentukan nilai awal dari input. Kondisi `if(isset($_POST['cetak']))` digunakan untuk memeriksa apakah formulir telah disubmit dengan mengecek apakah tombol "Cetak" (name="cetak") telah ditekan. Apabila formulir telah disubmit, nilai dari input dengan nama "panjang" (`$_POST['panjang']`) akan ditampilkan kembali ke dalam kotak input ini sebagai nilai awal. Selanjutnya terdapat juga penggunaan tag `
`, tag tersebut digunakan untuk membuat baris baru setelah kotak input, sehingga elemen formulir berada di baris yang berbeda.

Pada baris [25], syntax Lebar: `<input type="text" name="lebar" value="<?php if(isset($_POST['cetak'])) {echo $_POST['lebar'];}?">
` penggunaannya pada baris merupakan penerima input nilai lebar matriks dari pengguna. Penggunaan elemen input HTML ini akan menampilkan sebuah kotak input teks. Penggunaan atribut type="text" menunjukkan bahwa input ini dapat menerima teks. Nama input ditentukan oleh atribut

name="lebar", yang akan digunakan untuk mengidentifikasi nilai yang dimasukkan saat formulir disubmit. Pada atribut value, kode PHP disisipkan untuk menentukan nilai awal dari input. Kondisi if(isset(\$_POST['cetak'])) digunakan untuk memeriksa apakah formulir telah disubmit dengan mengecek apakah tombol "Cetak" (name="cetak") telah ditekan. Apabila formulir telah disubmit, nilai dari input dengan nama "lebar" (\$_POST['panjang']) akan ditampilkan kembali ke dalam kotak input ini sebagai nilai awal. Selanjutnya terdapat juga penggunaan tag
, tag tersebut digunakan untuk membuat baris baru setelah kotak input, sehingga elemen formulir berada di baris yang berbeda.

Pada baris [26], syntax Nilai: `<input type="text" name="nilai" value="<?php if(isset($_POST['cetak'])) {echo $_POST['nilai'];}?">
` penggunaannya pada baris merupakan penerima input nilai-nilai dalam matriks yang ingin dimasukkan dari pengguna. Penggunaan elemen input HTML yang menampilkan sebuah kotak input teks. Penggunaan atribut type="text" menunjukkan bahwa input ini dapat menerima teks. Nama input ditentukan oleh atribut name="nilai", yang akan digunakan untuk mengidentifikasi nilai-nilai matriks saat formulir disubmit. Pada atribut value, kode PHP disisipkan untuk menentukan nilai awal dari input. Kondisi if(isset(\$_POST['cetak'])) digunakan untuk memeriksa apakah formulir telah disubmit dengan mengecek apakah tombol "Cetak" (name="cetak") telah ditekan. Apabila formulir telah disubmit, nilai dari input dengan nama "nilai" (\$_POST['nilai']) akan ditampilkan kembali ke dalam kotak input ini sebagai nilai awal. Selanjutnya terdapat juga penggunaan tag
, tag tersebut digunakan untuk membuat baris baru setelah kotak input, sehingga elemen formulir berada di baris yang berbeda.

Pada baris [27], syntax `< input type="submit" value="Cetak" name="cetak">` berguna sebagai tombol submit dalam sebuah formulir di halaman web. Tombol submit ini memiliki atribut name yang ditetapkan sebagai "cetak", yang digunakan untuk mengidentifikasi tombol submit ketika formulir dikirimkan ke server. Nilai dari atribut value yang digunakan adalah "Cetak", yang berarti menentukan teks

yang akan ditampilkan pada tombol tersebut di halaman web. Ketika tombol ini ditekan oleh pengguna, data dari formulir yang telah diisi akan dikirimkan ke server untuk diproses.

Pada baris [28], syntax `</form>` baris ini merupakan tag penutup dari baris 23.

Pada baris [30], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [31] [32] [33], syntax

```
$panjang = "";
```

```
$lebar = "";
```

```
$nilai = "";
```

penggunaan baris-baris tersebut adalah sebagai tempat menginisialisasi variabel `$panjang`, `$lebar`, dan `$nilai` dengan nilai awal kosong (string kosong), penggunaannya agar dapat digunakan nanti untuk menyimpan data yang diterima dari pengguna melalui formulir. Menginisialisasi variabel-variabel tersebut untuk memastikan bahwa variabel memiliki nilai awal sebelum digunakan untuk menyimpan data yang diterima dari pengguna.

Pada baris [35] [36] [37] [38] [39] [40] [41], syntax

```
if (isset($_POST["cetak"])) {  
  
    $panjang = $_POST["panjang"];  
  
    $lebar = $_POST["lebar"];  
  
    $nilai = $_POST["nilai"];  
  
    $isi = explode(" ", $nilai);
```

```
$panjangNilai = count($isi);
```

penggunaan baris-baris tersebut merupakan blok logika PHP yang akan dieksekusi ketika tombol submit dengan nama "cetak" diklik pada sebuah formulir, yang selanjutnya akan diproses oleh kondisi if (isset(\$_POST['cetak'])). Saat kondisi terpenuhi, langkah-langkah berikut akan dijalankan. Langkah pertama yang dilakukan adalah baris kode if (isset(\$_POST["cetak"])) { ... } bertugas untuk melakukan pengecekan tersebut. Kemudian, dalam blok tersebut, nilai dari input field dengan nama "panjang", "lebar", dan "nilai" yang dikirimkan melalui metode POST dari formulir diambil menggunakan baris \$panjang = \$_POST["panjang"]; \$lebar = \$_POST["lebar"]; \$nilai = \$_POST["nilai"];. Selanjutnya, penggunaan baris \$isi = explode(" ", \$nilai); \$panjangNilai = count(\$isi); berfungsi untuk mengubah string yang diterima dari input field "nilai" menjadi sebuah array menggunakan fungsi explode(), dengan spasi (" ") sebagai pemisah. Setelah itu, dilakukanlah perhitungan jumlah elemen dalam array tersebut.

Pada baris [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64], syntax

```
if ($panjang * $lebar == $panjangNilai) {  
    $count = 0;  
    for ($i = 0; $i < $panjang; $i++) {  
        for ($j = 0; $j < $lebar; $j++) {  
            $stampilnilai[$i][$j] = $isi[$count];  
            $count++;  
        }  
    }  
}
```



```

        echo "<table>";

        for ($i = 0; $i < $panjang; $i++) {

            echo "<tr>";

            for ($j = 0; $j < $lebar; $j++) {

                echo "<td>" . $tampilnilai[$i][$j] .

"</td>";

            }

            echo "</tr>";

        }

        echo "</table>";

    } else {

        echo "Panjang nilai tidak sesuai dengan ukuran

matriks";

    }

}

```

penggunaan baris-baris tersebut digunakan untuk memeriksa apakah jumlah elemen yang diterima dari input field "nilai" sesuai dengan ukuran matriks yang ditentukan oleh pengguna, yang merupakan hasil perkalian dari panjang dan lebar matriks. Apabila jumlah elemen sama dengan panjang kali lebar matriks, maka matriks dapat dibentuk dan ditampilkan. Pada bagian if ($\$panjang * \$lebar == \$panjangNilai$) { ... }, dilakukan untuk pemeriksaan tersebut. Dalam blok ini, matriks dibentuk dengan menggunakan nested loop for untuk mengisi setiap elemennya dengan nilai dari input field "nilai". Variabel \$count digunakan untuk mengakses setiap elemen dari array \$isi, yang memuat nilai-nilai dari input field "nilai". Setelah matriks terbentuk dengan menggunakan echo "<table>"; langkah selanjutnya adalah dilakukan pencetakan

matriks dalam bentuk tabel HTML menggunakan tag `<table>`, `<tr>`, dan `<td>`. Apabila jumlah elemen tidak sesuai dengan ukuran matriks yang ditentukan, maka pesan "Panjang nilai tidak sesuai dengan ukuran matriks" akan dicetak.

Pada baris [65], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [66], syntax `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 22.

Pada baris [67], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 2

Diberika sebuah Kumpulan data seperti berikut:

Nama	NIM	Nilai UTS	Nilai UAS
Andi	2101001	87	65
Budi	2101002	76	79
Tono	2101003	50	41
Jessica	2101004	60	75

Implementasikan data diatas menjadi multi-dimensional associative array pada Bahasa pemrograman PHP kemudian tambahkan dua kolom baru. Kolom pertama adalah hasil perhitungan nilai akhir dari nilai UTS dan UAS dengan rumus seperti berikut: $40\% \text{ UTS} + 60\% \text{ UAS}$. Kolom kedua berisi nilai huruf dari nilai akhir berdasarkan ketentuan

HURUF	NILAI
A	≥ 80
B	70 - 79
C	60 - 69
D	50 - 59
E	< 50

berikut:

***dua kolom baru tersebut didapatkan berdasarkan baris kode logika hasil implementasi bukan dihitung manual**

Sehingga didapatn output seperti berikut

Nama	NIM	Nilai UTS	Nilai UAS	Nilai Akhir	Huruf
Andi	2101001	87	65	73.8	B
Budi	2101002	76	79	77.8	B
Tono	2101003	50	41	44.6	E
Jessica	2101004	60	75	69	C

Simpan dengan nama file: PRAK402.php

A. Source Code

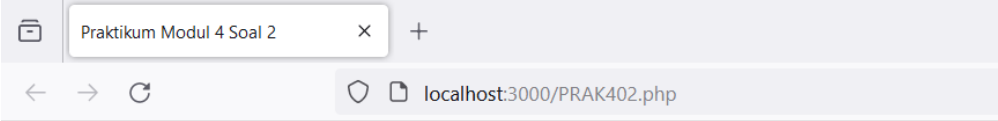
Table 16. Source Code Modul 4 Soal 2

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Praktikum Modul 4 Soal 2</title>
9	<style>
10	table,
11	tr,
12	td,
13	th {
14	border: solid 1px black;
15	border-collapse: collapse;
16	padding: 10px;
17	padding-right: 20px;
18	}
19	</style>
20	</head>
21	
22	<body>
23	<table>
24	<tr style="background-color: #D3D3D3;">
25	<th>Nama</th>
26	<th>NIM</th>

27	<th>Nilai UTS</th>
28	<th>Nilai UAS</th>
29	<th>Nilai Akhir</th>
30	<th>Huruf</th>
31	</tr>
32	
33	<?php
34	\$data = [
35	["nama" => "Andi", "nim" => "2101001",
	"uts" => 87, "uas" => 65],
36	["nama" => "Budi", "nim" => "2101002",
	"uts" => 76, "uas" => 79],
37	["nama" => "Tono", "nim" => "2101003",
	"uts" => 50, "uas" => 41],
38	["nama" => "Jessica", "nim" => "2101004",
	"uts" => 60, "uas" => 75],
39]; \$n = count(\$data);
40	
41	for (\$i=0; \$i < \$n; \$i++) {
42	\$data[\$i]["akhir"] = \$data[\$i]["uts"] * (40/100) + \$data[\$i]["uas"] * (60/100);
43	if (\$data[\$i]["akhir"] >= 80) {
44	\$data[\$i]["huruf"] = "A";
45	} elseif (\$data[\$i]["akhir"] >= 70) {
46	\$data[\$i]["huruf"] = "B";
47	} elseif (\$data[\$i]["akhir"] >= 60) {
48	\$data[\$i]["huruf"] = "C";
49	} elseif (\$data[\$i]["akhir"] >= 50) {
50	\$data[\$i]["huruf"] = "D";
51	} else {

52	\$data[\$i]["huruf"] = "E";
53	}
54	}
55	
56	for (\$i=0; \$i < \$n; \$i++) {
57	echo "<tr>";
58	echo "<td>".\$data[\$i]["nama"]."</td>";
59	echo "<td>".\$data[\$i]["nim"]."</td>";
60	echo "<td>".\$data[\$i]["uts"]."</td>";
61	echo "<td>".\$data[\$i]["uas"]."</td>";
62	echo "<td>".\$data[\$i]["akhir"]."</td>";
63	echo "<td>".\$data[\$i]["huruf"]."</td>";
64	echo "</tr>";
65	}
66	?>
67	</table>
68	</body>
69	</html>

B. Output Program



The screenshot shows a web browser window with the title 'Praktikum Modul 4 Soal 2'. The address bar displays 'localhost:3000/PRAK402.php'. Below the browser window, a table is displayed with the following data:

Nama	NIM	Nilai UTS	Nilai UAS	Nilai Akhir	Huruf
Andi	2101001	87	65	73.8	B
Budi	2101002	76	79	77.8	B
Tono	2101003	50	41	44.6	E
Jessica	2101004	60	75	69	C

Gambar 23. Screenshot Hasil Jawaban Modul 4 Soal 2

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus

ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8], syntax `<title>Praktikum Modul 4 Soal 2</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [10] [11] [12] [13] [14] [15] [16] [17] [18] [19], syntax

```
<style>

    table,

    tr,

    td,

    th {

        border: solid 1px black;
```



```

border-collapse: collapse;

padding: 10px;

padding-right: 20px;

}

</style>

```

pada baris-baris tersebut bertujuan untuk memberikan gaya pada elemen-elemen tabel dalam halaman web. Setiap penggunaan aturan gaya yang didefinisikan dalam blok kode CSS tersebut akan mengatur penampilan dan tata letak dari elemen-elemen table, misalnya baris tabel, sel tabel, dan sel kepala tabel. Pada baris tersebut mengatur border untuk menentukan garis tepi yang solid dengan ketebalan 1 piksel dan warna hitam untuk setiap elemen tabel. Selanjutnya, dengan menggunakan properti border-collapse diatur agar border dari sel-sel tabel digabungkan menjadi satu garis solid. Properti padding akan memberikan jarak sebesar 10 piksel di sekitar konten di dalam setiap elemen tabel, lalu digunakannya padding-right untuk menambahkan jarak sebesar 20 piksel di sisi kanan konten.

Pada baris [21], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [23], syntax `<body>` tag Body merupakan section utama dalam dokumen web.

Pada baris [24], syntax `<table>` digunakan untuk membuat sebuah table.

Pada baris [25] [26] [27] [28] [29] [30] [31] [32], syntax

```

<tr style="background-color: #D3D3D3;">

    <th>Nama</th>

    <th>NIM</th>

    <th>Nilai UTS</th>

    <th>Nilai UAS</th>

```

```

        <th>Nilai Akhir</th>

        <th>Huruf</th>

    </tr>

```

penggunaan baris-baris tersebut adalah untuk membuat baris tabel yang berisi header atau judul kolom. Di dalam baris tersebut, terdapat tag `<th>` yang digunakan untuk mendefinisikan sel-sel header. Setiap tag `<th>` akan menandakan sebuah judul kolom di dalam tabel. Selanjutnya digunakan atribut `style="background-color: #D3D3D3;"` diberikan pada tag `<tr>` untuk memberikan warna latar belakang pada baris header. Tujuan penggunaan warna latar belakang untuk membedakan visual antara baris header dengan baris data pada tabel.

Pada baris [34], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [35] [36] [37] [38] [39] [40], syntax

```

$data = [

    ["nama" => "Andi", "nim" => "2101001", "uts"
=> 87, "uas" => 65],

    ["nama" => "Budi", "nim" => "2101002", "uts"
=> 76, "uas" => 79],

    ["nama" => "Tono", "nim" => "2101003", "uts"
=> 50, "uas" => 41],

    ["nama" => "Jessica", "nim" => "2101004",
"uts" => 60, "uas" => 75],

]; $n = count($data);

```

penggunaan baris-baris tersebut untuk mendefinisikan sebuah array asosiatif yang disebut \$data. Setiap elemen dalam array ini adalah sebuah array lain yang akan mewakili data seorang siswa, yaitu memuat nama, NIM, nilai UTS, dan nilai UAS. Dengan menggunakan susunan data ini, informasi siswa-siswa dapat tersimpan dalam sebuah struktur yang jelas, akan memudahkan pengelolaan dan pemrosesan data. Selanjutnya, variabel \$n diinisialisasi dengan nilai jumlah elemen dalam array \$data, yang merupakan jumlah total siswa yang dimasukkan ke dalam array. Dengan penggunaan \$n = count(\$data); tersebut memungkinkan iterasi atau pengulangan melalui setiap elemen dalam array menggunakan loop, dengan batas atas iterasi sesuai dengan jumlah siswa yang ada dalam data.

Pada baris [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55], syntax

```
for ($i=0; $i < $n; $i++) {  
  
    $data[$i]["akhir"] = $data[$i]["uts"] * (40/100) +  
    $data[$i]["uas"] * (60/100);  
  
    if ($data[$i]["akhir"] >= 80) {  
  
        $data[$i]["huruf"] = "A";  
  
    } elseif ($data[$i]["akhir"] >= 70) {  
  
        $data[$i]["huruf"] = "B";  
  
    } elseif ($data[$i]["akhir"] >= 60) {  
  
        $data[$i]["huruf"] = "C";  
  
    } elseif ($data[$i]["akhir"] >= 50) {  
  
        $data[$i]["huruf"] = "D";  
  
    } else {  
  
        $data[$i]["huruf"] = "E";  
  
    }  
}
```

```
}
```

penggunaan baris-baris tersebut bertanggung jawab untuk menghitung nilai akhir dan menentukan huruf mutu untuk setiap siswa dalam array \$data. Langkah awal akan dilakukan iterasi terlebih dahulu melalui setiap elemen siswa dalam array menggunakan loop for, dimulai dari indeks 0 hingga \$n - 1, di mana \$n adalah jumlah siswa dalam array. Di dalam loop, nilai akhir untuk setiap siswa dihitung dengan rumus tertentu, yaitu 40% dari nilai UTS ditambah 60% dari nilai UAS. Nilai akhir yang sudah dihitung, akan dilakukan pengecekan kondisi menggunakan struktur if-elseif-else untuk menentukan nilai huruf mutu yang sesuai dengan nilai akhir. Apabila nilai akhir lebih besar dari atau sama dengan 80, maka huruf mutu adalah "A"; jika antara 70 dan 79, huruf mutu adalah "B"; jika antara 60 dan 69, huruf mutu adalah "C"; jika antara 50 dan 59, huruf mutu adalah "D"; dan jika kurang dari 50, huruf mutu adalah "E". Setelah huruf mutu tersebut ditentukan, maka nilai huruf tersebut disimpan ke dalam array \$data untuk setiap siswa yang ada.

Pada baris [57] [58] [59] [60] [61] [62] [63] [64] [65] [66], syntax

```
for ($i=0; $i < $n; $i++) {  
    echo "<tr>";  
    echo "<td>".$data[$i]["nama"]."</td>";  
    echo "<td>".$data[$i]["nim"]."</td>";  
    echo "<td>".$data[$i]["uts"]."</td>";  
    echo "<td>".$data[$i]["uas"]."</td>";  
    echo "<td>".$data[$i]["akhir"]."</td>";  
    echo "<td>".$data[$i]["huruf"]."</td>";  
    echo "</tr>";  
}
```

penggunaan baris-baris tersebut bertujuan untuk menampilkan data mahasiswa beserta nilai-nilainya dalam bentuk tabel HTML. Melalui penggunaan loop for, setiap elemen dalam array yang diinisialisasi dengan \$data yang berisi informasi nama, NIM, nilai UTS, serta nilai UAS akan diakses secara berurutan. Dalam setiap iterasi loop, nilai akhir setiap mahasiswa dihitung berdasarkan bobot nilai UTS dan UAS, dan kemudian akan diberikan huruf mutu sesuai dengan skala yang ditetapkan. Setiap iterasi loop mencetak satu baris dalam tabel HTML yang terdiri dari kolom nama, NIM, nilai UTS, nilai UAS, nilai akhir, dan huruf mutu. Untuk setiap elemen tersebut, baris baru (<tr>) dalam tabel akan dibuat. Selanjutnya, menggunakan perintah echo, data untuk setiap kolom dalam tabel (nama, NIM, nilai UTS, nilai UAS, nilai akhir, dan huruf mutu) dimasukkan ke dalam sel (<td>) yang sesuai. Proses ini dilakukan secara berulang hingga semua data mahasiswa telah dicetak ke dalam tabel.

Pada baris [67], `</tr>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [68], `</table>` merupakan tag penutup dari baris 24.

Pada baris [69], `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 11.

Pada baris [70], `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

SOAL 3

Buatlah Diberikan sebuah Kumpulan data seperti berikut:

No	Nama	Mata Kuliah diambil	SKS
1	Ridho	Pemrograman I	2
		Praktikum Pemrograman I	1
		Pengantar Lingkungan Lahan Basah	2
		Arsitektur Komputer	3
2	Ratna	Basis Data I	2
		Praktikum Basis Data I	1
		Kalkulus	3
3	Tono	Rekayasa Perangkat Lunak	3
		Analisis dan Perancangan Sistem	3
		Komputasi Awan	3
		Kecerdasan Bisnis	3

Implementasikan data diatas menjadi multi-dimensional associative array pada Bahasa pemrograman PHP kemudian tambahkan dua kolom baru. Kolom pertama adalah total sks dari mata kuliah yang diambil mahasiswa. Kolom kedua adalah keterangan jika total sks yang diambil kurang dari 7 maka diberi keterangan “Revisi KRS”, jika tidak maka diberi keterangan “Tidak Revisi”.

***dua kolom baru tersebut didapatkan berdasarkan baris kode logika hasil implementasi bukan dihitung manual**

Cetak hasil output seperti berikut:

No	Nama	Mata Kuliah diambil	SKS	Total SKS	Keterangan
1	Ridho	Pemrograman I	2	8	Tidak Revisi
		Praktikum Pemrograman I	1		
		Pengantar Lingkungan Lahan Basah	2		
		Arsitektur Komputer	3		
2	Ratna	Basis Data I	2	6	Revisi KRS
		Praktikum Basis Data I	1		
		Kalkulus	3		
3	Tono	Rekayasa Perangkat Lunak	3	12	Tidak Revisi
		Analisis dan Perancangan Sistem	3		
		Komputasi Awan	3		
		Kecerdasan Bisnis	3		

Simpan dengan nama file: PRAK403.php

A. Source Code

Table 17. Source Code Modul 4 Soal 3

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	
4	<code><head></code>
5	<code><meta charset="UTF-8"></code>

6	<code><meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>
7	<code><meta name="viewport" content="width=device-width,</code>
	<code>initial-scale=1.0"></code>
8	<code><title>Praktikum Modul 4 Soal 3</title></code>
9	<code><style></code>
10	<code>table, tr, td, th {</code>
11	<code>border: solid 1px black;</code>
12	<code>border-collapse: collapse;</code>
13	<code>padding: 5px;</code>
14	<code>}</code>
15	<code>table{</code>
16	<code>width: 700px;</code>
17	<code>}</code>
18	
19	<code>table tr th{</code>
20	<code>background-color: lightgray;</code>
21	<code>text-align: left;</code>
22	<code>}</code>
23	<code></style></code>
24	<code></head></code>
25	
26	<code><body></code>
27	<code><?php</code>
28	<code>\$nilai = [</code>
29	<code>["no" => 1, "nama" => "Ridho",</code>
30	<code>"matkul" => [</code>
31	<code>["namamatkul" =>"Pemrograman I", "sks"</code>
	<code>=> 2],</code>
32	

33	["namamatkul" => "Praktikum Pemrograman I", "sks" => 1],
34	["namamatkul" => "Pengantar Lingkungan Lahan Basah", "sks" => 2],
35	["namamatkul" => "Arsitektur Komputer", "sks" => 3]
36]
37],
38	["no" => 2, "nama" => "Ratna",
39	"matkul" => [
40	["namamatkul" =>"Basis Data I", "sks" => 2],
41	["namamatkul" => "Praktikum Basis Data I", "sks" => 1],
42	["namamatkul" => "Kalkulus", "sks" => 3]
43]
44],
45	["no" => 3, "nama" => "Tono",
46	"matkul" => [
47	["namamatkul" => "Rekayasa Perangkat Lunak", "sks" => 3],
48	["namamatkul" => "Analisis dan Perancangan Sistem", "sks" => 3],
49	["namamatkul" => "Komputasi Awan", "sks" => 3],
50	["namamatkul" => "Kecerdasan Bisnis", "sks" => 3]
51]
52]

53];
54	for (\$i=0; \$i < count(\$nilai); \$i++){
55	\$totalSks = 0;
	for (\$j=0; \$j <
56	count(\$nilai[\$i]["matkul"]); \$j++) {
	\$totalSks +=
57	\$nilai[\$i]["matkul"][\$j]["sks"];
58	}
59	\$nilai[\$i]["totalSks"] = \$totalSks;
60	if (\$nilai[\$i]["totalSks"] < 7) {
	\$nilai[\$i]["keterangan"] = "Revisi
61	KRS";
62	} else {
	\$nilai[\$i]["keterangan"] = "Tidak
63	Revisi";
64	}
65	}
66	?>
67	
68	<table>
69	<tr>
70	<th>No</th>
71	<th>Nama</th>
72	<th>Mata Kuliah diambil</th>
73	<th>SKS</th>
74	<th>Total SKS</th>
75	<th>Keterangan</th>
76	</tr>
77	
78	<?php

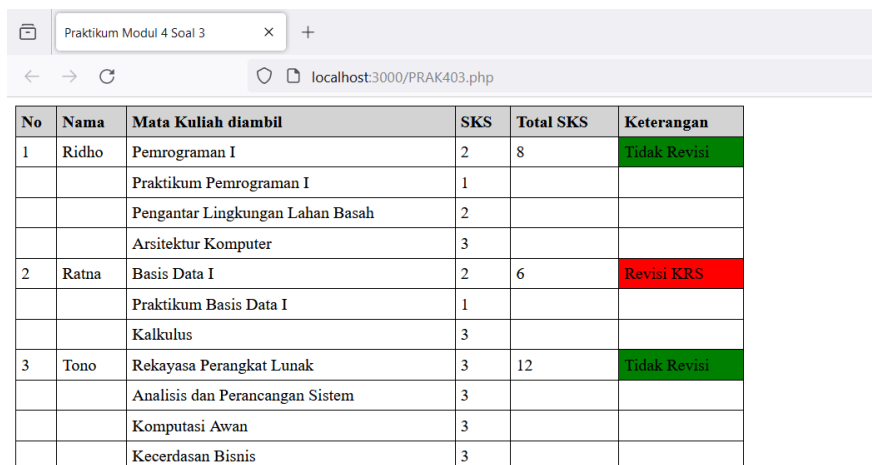
79	for (\$i=0; \$i < count(\$nilai); \$i++) {
	for (\$j=0; \$j <
80	count(\$nilai[\$i]["matkul"]); \$j++) {
81	echo "<tr>";
82	if (\$j == 0) {
	echo
83	"<td>".\$nilai[\$i]["no"]."</td>";
	echo
84	"<td>".\$nilai[\$i]["nama"]."</td>";
	echo
85	"<td>".\$nilai[\$i]["matkul"][\$j]["namamatkul"]."</td>";
	echo
86	"<td>".\$nilai[\$i]["matkul"][\$j]["sks"]."</td>";
	echo
87	"<td>".\$nilai[\$i]["totalSks"]."</td>";
	if (\$nilai[\$i]["keterangan"] ==
88	"Revisi KRS"){
	echo '<td style="background-
89	color: red;">'.\$nilai[\$i]["keterangan"]."</td>";
90	} else {
	echo '<td style="background-
91	color: green;">'.\$nilai[\$i]["keterangan"]."</td>";
92	}
93	} else {
94	echo "<td></td>";
95	echo "<td></td>";
	echo
96	"<td>".\$nilai[\$i]["matkul"][\$j]["namamatkul"]."</td>";
	echo
97	"<td>".\$nilai[\$i]["matkul"][\$j]["sks"]."</td>";

```

98         echo "<td></td>";
99         echo "<td></td>";
100     }
101     echo "</tr>";
102 }
103 }
104 ?>
105 </table>
106 </body>
</html>

```

B. Output Program



No	Nama	Mata Kuliah diambil	SKS	Total SKS	Keterangan
1	Ridho	Pemrograman I	2	8	Tidak Revisi
		Praktikum Pemrograman I	1		
		Pengantar Lingkungan Lahan Basah	2		
		Arsitektur Komputer	3		
2	Ratna	Basis Data I	2	6	Revisi KRS
		Praktikum Basis Data I	1		
		Kalkulus	3		
3	Tono	Rekayasa Perangkat Lunak	3	12	Tidak Revisi
		Analisis dan Perancangan Sistem	3		
		Komputasi Awan	3		
		Kecerdasan Bisnis	3		

Gambar 24. Screenshot Hasil Jawaban Modul 4 Soal 3

C. Pembahasan

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda

awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8], syntax `<title>Praktikum Modul 4 Soal 3</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau

teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23],
syntax

```
<style>

    table, tr, td, th {

        border: solid 1px black;

        border-collapse: collapse;

        padding: 5px;

    }

    table{

        width: 700px;

    }

    table tr th{

        background-color: lightgray;

        text-align: left;

    }

</style>
```

penggunaan kode pada baris-baris tersebut merupakan bagian dari elemen <style> dalam dokumen HTML yang bertanggung jawab untuk menetapkan tata letak dan tampilan elemen-elemen tabel dalam halaman web. Setiap baris kode memiliki peran spesifik dalam menentukan gaya dan properti yang akan diterapkan pada elemen-elemen tabel. Dengan menggunakan aturan CSS table, tr, td, th { ... } maka digunakan

untuk menetapkan properti gaya seperti border, border-collapse, dan padding untuk seluruh elemen tabel, seperti baris, sel data, dan sel header. Dengan mengatur border: solid 1px black, maka garis tepi hitam solid dengan ketebalan satu piksel diberikan pada setiap elemen tabel, sehingga menciptakan batasan yang jelas antara setiap sel dan baris. Selain itu, penggunaan pengaturan padding: 5px memberikan jarak internal lima piksel di sekitar isi setiap sel, meningkatkan keterbacaan dan estetika. Lebar tabel diatur menjadi 700 piksel, sehingga memastikan bahwa tabel memiliki lebar tetap dan tidak menyesuaikan ukuran layar pengguna. Serta pada baris tersebut terdapat pengaturan bagian sel header dalam baris tabel untuk mengatur warna latar belakang berupa abu-abu muda dan penataan teks menjadi kiri, pada sel-sel header memberikan penonjolan visual yang membedakan antara isi tabel dan judul kolom.

Pada baris [24], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [26], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [27], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [28] [29] [30] [31] [32] [33] [34] [35] [36], syntax

```
$nilai = [
    ["no" => 1, "nama" => "Ridho",
    "matkul" => [
        ["namamatkul" => "Pemrograman I", "sks" =>
2],
        ["namamatkul" => "Praktikum Pemrograman
I", "sks" => 1],
```

```

        ["namamatkul" => "Pengantar Lingkungan
Lahan Basah", "sks" => 2],

        ["namamatkul" => "Arsitektur Komputer",
"sks" => 3]

    ]

],

```

penggunaan baris-baris tersebut mendefinisikan variabel `$nilai` dan menginisialisasinya dengan sebuah array. Array ini berisikan informasi tentang beberapa mahasiswa dan mata kuliah yang diambil oleh masing-masing mahasiswa beserta jumlah SKS dari setiap mata kuliahnya. Elemen pertama dari array `$nilai` adalah `["no" => 1, "nama" => "Ridho", ...]`, yang menyimpan informasi tentang mahasiswa pertama, seperti nomor, nama, dan daftar mata kuliah yang diambil beserta jumlah SKS dari setiap mata kuliahnya. Bagian `"matkul" => [...]` merupakan sub-array dalam elemen utama yang berisi daftar mata kuliah yang diambil oleh mahasiswa bersangkutan. Setiap mata kuliah direpresentasikan sebagai array asosiatif yang memiliki kunci `"namamatkul"` untuk nama mata kuliah dan `"sks"` untuk jumlah SKS. Dengan struktur data ini, informasi tentang mahasiswa dan mata kuliah mereka tersedia dalam satu variabel yang dapat diakses dan dikelola dalam program PHP, sehingga memungkinkan untuk dilakukannya berbagai operasi seperti perhitungan total SKS atau penyimpanan data.

Pada baris [37] [38] [39] [40] [41] [42] [43], syntax

```

["no" => 2, "nama" => "Ratna",

    "matkul" => [

        ["namamatkul" => "Basis Data I", "sks" =>
2],

        ["namamatkul" => "Praktikum Basis Data I",
"sks" => 1],

```



```

        ["namamatkul" => "Kalkulus", "sks" => 3]
    ]
],

```

penggunaan baris-baris tersebut merupakan bagian dari inisialisasi variabel array yang menampung informasi tentang mahasiswa. Pada bagian ini, terdapat data untuk mahasiswa dengan nomor urut 2, bernama "Ratna". Sub-array "matkul" menyimpan informasi tentang mata kuliah yang diambil oleh mahasiswa tersebut. Setiap mata kuliah direpresentasikan sebagai array asosiatif yang memiliki dua kunci, yaitu "namamatkul" untuk nama mata kuliah dan "sks" untuk jumlah SKS (Satuan Kredit Semester) dari mata kuliah tersebut. Dengan cara ini, informasi yang berkaitan dengan mata kuliah yang diambil oleh mahasiswa dapat tersusun secara terstruktur dalam array, memudahkan dalam pengelolaan dan pengaksesan informasi tersebut dalam program.

Pada baris [44] [45] [46] [47] [48] [49] 50] [51] [52], syntax

```

["no" => 3, "nama" => "Tono",
    "matkul" => [
        ["namamatkul" => "Rekayasa Perangkat
Lunak", "sks" => 3],
        ["namamatkul" => "Analisis dan Perancangan
Sistem", "sks" => 3],
        ["namamatkul" => "Komputasi Awan", "sks"
=> 3],
        ["namamatkul" => "Kecerdasan Bisnis",
"sks" => 3]
    ]

```

```

    ]
];

```

penggunaan kode pada baris-baris tersebut digunakan untuk menginisialisasi sebuah array yang menyimpan informasi tentang seorang mahasiswa dengan nomor urut 3, yang bernama "Tono". Sub-array "matkul" menyimpan informasi tentang mata kuliah yang diambil oleh mahasiswa tersebut. Setiap mata kuliah direpresentasikan sebagai array asosiatif yang memiliki dua kunci, yaitu "namamatkul" untuk nama mata kuliah dan "sks" untuk jumlah SKS dari mata kuliah tersebut. Dengan menggunakan struktur data ini, informasi tentang mata kuliah yang diambil oleh mahasiswa dapat disimpan secara terstruktur dalam array, memungkinkan untuk pengolahan data lebih lanjut dalam program.

Pada baris [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64], syntax

```

for ($i=0; $i < count($nilai); $i++){

    $totalSks = 0;

    for ($j=0; $j < count($nilai[$i]["matkul"]);
    $j++) {

        $totalSks                                     +=
$nilai[$i]["matkul"][$j]["sks"];

    }

    $nilai[$i]["totalSks"] = $totalSks;

    if ($nilai[$i]["totalSks"] < 7) {

        $nilai[$i]["keterangan"] = "Revisi KRS";

    } else {

        $nilai[$i]["keterangan"] = "Tidak Revisi";

    }
}

```

}

pada baris-baris tersebut penggunaannya untuk proses penghitungan total SKS pada setiap mahasiswa berdasarkan mata kuliah yang diambilnya, serta memberikan keterangan apakah mahasiswa tersebut perlu melakukan revisi KRS atau tidak. Loop for pertama digunakan untuk mengiterasi melalui setiap elemen dalam array \$nilai, yang mewakili informasi tentang setiap mahasiswa. Di dalam loop tersebut, variabel \$i berperan sebagai penghitung iterasi, dimulai dari 0 hingga kurang dari jumlah elemen dalam array \$nilai. Pada setiap iterasi, variabel \$totalSks diinisialisasi dengan nilai awal 0 untuk menyimpan total SKS dari semua mata kuliah yang diambil oleh seorang mahasiswa. Kemudian, terdapat loop for bersarang yang digunakan untuk mengiterasi melalui setiap mata kuliah yang diambil oleh mahasiswa pada iterasi saat ini. Variabel \$j digunakan sebagai penghitung iterasi di dalam loop ini. Pada setiap iterasi dalam loop mata kuliah, nilai SKS dari mata kuliah saat ini ditambahkan ke variabel \$totalSks, sehingga secara bertahap total SKS dari semua mata kuliah yang diambil oleh mahasiswa tersebut dihitung. Setelah loop mata kuliah selesai, nilai total SKS yang telah dihitung disimpan dalam array asosiatif \$nilai pada indeks yang sesuai dengan mahasiswa pada iterasi saat ini. Selanjutnya, digunakan struktur kontrol if-else untuk menentukan apakah total SKS yang diambil oleh mahasiswa pada iterasi saat ini kurang dari 7 atau tidak. Apabila total SKS kurang dari 7, maka mahasiswa tersebut diberi keterangan "Revisi KRS", yang menunjukkan bahwa jumlah SKS yang diambil belum mencukupi. Namun, jika total SKS lebih dari atau sama dengan 7, maka mahasiswa tersebut diberi keterangan "Tidak Revisi", yang menandakan bahwa jumlah SKS yang diambil sudah memenuhi atau melebihi batas minimum yang ditetapkan.

Pada baris [65], syntax?> digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [67] [68] [69] [70] [71] [72] [73] [74] [75], syntax

```

<table>

    <tr>

        <th>No</th>

        <th>Nama</th>

        <th>Mata Kuliah diambil</th>

        <th>SKS</th>

        <th>Total SKS</th>

        <th>Keterangan</th>

    </tr>

```

penggunaan baris-baris tersebut merupakan bagian dari struktur HTML untuk membuat sebuah tabel. Tabel dimulai dengan tag pembuka `<table>`, ini menandakan awal dari pembuatan tabel. Setiap elemen baris dalam tabel diawali dengan tag pembuka `<tr>`, menandakan awal dari sebuah baris dalam tabel. Kemudian, setiap judul atau label kolom didefinisikan menggunakan tag pembuka `<th>` untuk elemen sel header dalam tabel. Seperti, `<th>No</th>` menandakan bahwa kolom ini akan berisi judul atau label "No". Selanjutnya penggunaan tag `<th>` untuk judul kolom adalah dikarenakan hal itu adalah bagian header dari tabel yang biasanya digunakan untuk menyajikan judul atau label. Setiap baris kode tersebut akan memberikan kontribusi dalam mendefinisikan struktur dan tampilan visual dari tabel yang akan ditampilkan dalam halaman web.

Pada baris [77], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95][96] [97] [98] [99] [100] [101] [102], syntax

```

for ($i=0; $i < count($nilai); $i++) {
    for ($j=0; $j < count($nilai[$i]["matkul"]);
    $j++) {
        echo "<tr>";
        if ($j == 0) {
            echo "<td>".$nilai[$i]["no"]."</td>";
            echo
"<td>".$nilai[$i]["nama"]."</td>";
            echo
"<td>".$nilai[$i]["matkul"][$j]["namamatkul"]."</td>";
            echo
"<td>".$nilai[$i]["matkul"][$j]["sks"]."</td>";
            echo
"<td>".$nilai[$i]["totalSks"]."</td>";
            if ($nilai[$i]["keterangan"] ==
"Revisi KRS"){
                echo '<td style="background-
color: red;">'.$nilai[$i]["keterangan"]."</td>";
            } else {
                echo '<td style="background-
color: green;">'.$nilai[$i]["keterangan"]."</td>";
            }
        } else {
            echo "<td></td>";
        }
    }
}

```

```

        echo "<td></td>";

        echo
"<td>".$nilai[$i]["matkul"][$j]["namamatkul"]."</td>";

        echo
"<td>".$nilai[$i]["matkul"][$j]["sks"]."</td>";

        echo "<td></td>";

        echo "<td></td>";

    }

    echo "</tr>";

}

}

```

pada baris-baris tersebut tujuan penggunaan kode untuk menghasilkan baris-baris tabel yang menampilkan rincian mata kuliah yang diambil oleh setiap mahasiswa, termasuk nomor mahasiswa, nama, mata kuliah, jumlah SKS, total SKS yang diambil, dan keterangan mengenai apakah perlu revisi KRS atau tidak. Proses tersebut terdapat loop pertama yang digunakan untuk mengiterasi melalui setiap elemen dalam array \$nilai, yang merepresentasikan setiap mahasiswa. Kemudian, terdapat loop kedua yang mengiterasi melalui setiap mata kuliah yang diambil oleh mahasiswa pada iterasi saat ini. Setiap iterasi dalam loop ini menghasilkan sebuah baris dalam tabel yang dimulai dengan tag <tr>. Struktur kontrol digunakan untuk memeriksa apakah iterasi saat ini adalah iterasi pertama dalam loop mata kuliah atau bukan. Jika iterasi pertama, informasi tambahan tentang mahasiswa, seperti nomor, nama, total SKS, dan keterangan, ditampilkan. Apabila tidak, hanya informasi mata kuliah yang ditampilkan. Setelah itu penggunaan tag <td> digunakan untuk menambahkan sel kosong jika ini bukan iterasi pertama dalam loop mata kuliah untuk mempertahankan struktur tabel. Selanjutnya yang terakhir, terdapat struktur kontrol yang menentukan warna latar belakang sel keterangan berdasarkan keterangan yang diberikan, baik "Revisi KRS"

atau "Tidak Revisi". Setelah semua detail untuk sebuah mata kuliah ditampilkan, tag penutup `</tr>` digunakan untuk menandai akhir dari baris dalam tabel.

Pada baris [103], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [104], syntax `</table>` merupakan tag penutup dari tag 67.

Pada baris [105], syntax `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 16.

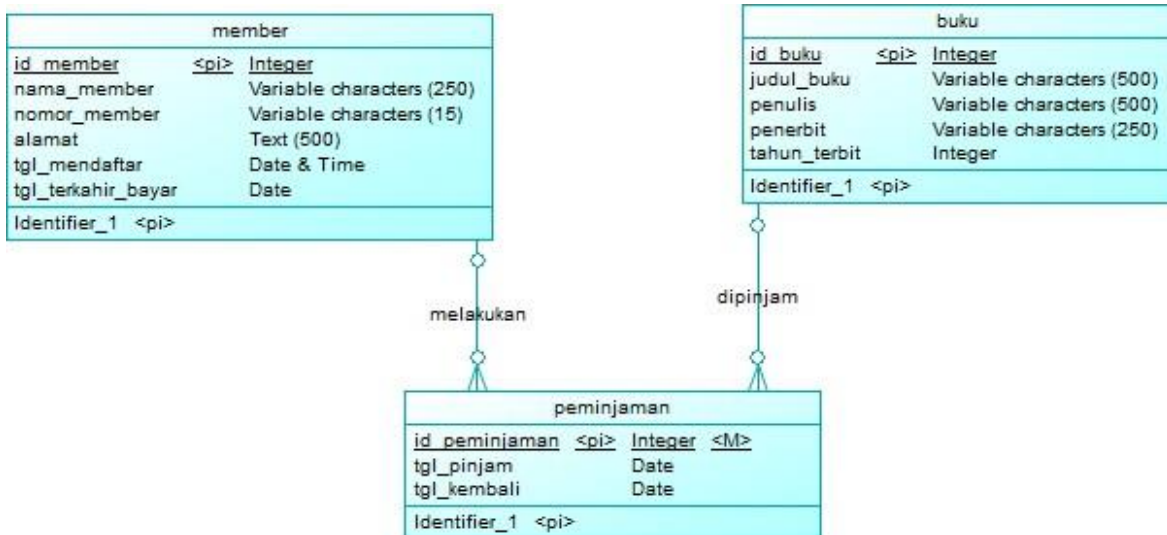
Pada baris [106], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

MODUL 5:

FUNCTION DAN DATABASE

SOAL PRAKTIKUM

Diberikan desain konseptual sebuah basis data seperti berikut.



Buatlah sebuah aplikasi berbasis web sederhana yang dapat melakukan operasi CRUD (Create, Read, Update, Delete) dari hasil implementasi desain basis data yang diberikan. Adapun ketentuan pembuatannya sebagai berikut:

Adapun ketentuan pembuatannya sebagai berikut:

1. Koneksi database dibuat menjadi satu file sendiri yaitu Koneksi.php, kemudian gunakan fungsi *require* ketika ingin melakukan operasi ke basis data.
2. Operasi data seperti Insert, Update, Delete, Get Data dibuat menjadi fungsi sendiri masing- masing dan disimpan di dalam satu file khusus yaitu Model.php
3. Minimal terdapat file-file berikut di dalam proyek

Nama File	Keterangan
Koneksi.php	Berisi fungsi-fungsi untuk melakukan koneksi ke basis data
Model.php	Berisi fungsi-fungsi untuk melakukan operasi data seperti insert, update, delete, get data untuk semua tabel
Member.php	Menampilkan semua data member dalam bentuk tabel dan terdapat tombol untuk melakukan tambah, edit, dan hapus data member.
FormMember.php	Menampilkan formulir isian untuk menambah atau mengubah data member.
Buku.php	Menampilkan semua data buku dalam bentuk tabel dan terdapat tombol untuk melakukan tambah, edit dan hapus data buku.
FormBuku.php	Menampilkan formulir isian untuk menambah atau mengubah data buku.
Peminjaman.php	Menampilkan semua data peminjaman dalam bentuk tabel dan terdapat tombol untuk melakukan tambah, edit, dan hapus data peminjaman.
FormPeminjaman.php	Menampilkan formulir isian untuk menambah atau mengubah data peminjaman.

Simpan dengan nama projek di Github: PRAK501

A. Source Code Koneksi.php

Table 18. Source Code Modul 5 Koneksi.php

1	<?php
2	\$DBHOST = "localhost";
3	\$DBNAME = "praktikum501";
4	\$USERNAME = "root";

5	\$PASSWORD = "";
6	try {
7	\$conn = new PDO(
8	"mysql:host=\$DBHOST;dbname=\$DBNAME",
9	\$USERNAME,
10	\$PASSWORD
11);
12	} catch (\Throwable \$e) {
13	echo "Koneksi Gagal, " . \$e->getMessage();
14	}
15	?>

B. Pembahasan Koneksi.php

Pada baris [1], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [2], syntax `$DBHOST = "localhost";` digunakan untuk mendeklarasikan variabel `$DBHOST` dan mengisinya dengan string "localhost". Baris ini menunjuk ke server database yang dijalankan di lokal.

Pada baris [3], syntax `$DBNAME = "praktikum501";` digunakan untuk mendeklarasikan variabel `$DBNAME` dan mengisinya dengan nama database "praktikum501", atau mengatakan bahwa database yang dibuat bernama "praktikum501".

Pada baris [4], syntax `$USERNAME = "root";` digunakan untuk mendeklarasikan variabel `$USERNAME` dan mengisinya dengan string "root", yang merupakan nama pengguna dari default untuk MySQL.

Pada baris [5], syntax `$PASSWORD = ""`; digunakan untuk mendeklarasikan variabel `$PASSWORD` dan mengisinya dengan string kosong `""`, yang artinya tidak ada kata sandi yang digunakan untuk mengakses database.

Pada baris [6] [7] [8] [9] [10] [11], syntax

```
try {  
  
    $conn = new PDO(  
  
        "mysql:host=$DBHOST;dbname=$DBNAME",  
  
        $USERNAME,  
  
        $PASSWORD  
  
    );
```

penggunaan baris-baris tersebut adalah pembuka blok dengan pernyataan `try {` yang digunakan untuk memulai blok `try`, blok ini digunakan untuk menangkap kesalahan yang mungkin terjadi selama eksekusi blok kode yang dibuat. Selanjutnya, dalam blok `try` terdapat perintah `$conn = new PDO("mysql:host=$DBHOST;dbname=$DBNAME", $USERNAME, $PASSWORD)`; yang berfungsi untuk mencoba membuat koneksi baru ke database dengan menggunakan kelas `PDO` atau `PHP Data Objects`. Pernyataan `new PDO("mysql:host=$DBHOST;dbname=$DBNAME", $USERNAME, $PASSWORD)`; ini, membuat objek `PDO` baru dengan menggunakan `DSN` atau `Data Source Name` yang terdiri dari tipe database yaitu `mysql`, host (`$DBHOST`), dan nama database (`$DBNAME`). Selanjutnya variabel `$USERNAME` dan `$PASSWORD` digunakan untuk autentikasi ke database. Jika koneksi berhasil, maka objek `PDO` akan disimpan dalam variabel `$conn`.

Pada baris [12] [13] [14], syntax

```
} catch (\Throwable $e) {  
  
    echo "Koneksi Gagal, " . $e->getMessage();
```

```
}
```

penggunaan baris-baris tersebut untuk menangani kesalahan dengan menggunakan pernyataan `} catch (\Throwable $e) {`, yang berfungsi untuk menangkap semua kesalahan yang terjadi dalam blok `try`. Pada program ini, `\Throwable` adalah kelas dasar untuk semua kesalahan dan exceptions di PHP, apabila ada kesalahan yang terjadi selama eksekusi blok `try`, maka program akan memasuki blok `catch`. Di dalam blok `catch`, terdapat juga perintah `echo "Koneksi Gagal, " . $e->getMessage();` dimana fungsinya untuk menampilkan pesan kesalahan. Pesan "Koneksi Gagal" akan ditampilkan diikuti oleh pesan error yang dikembalikan dari objek exception `$e`. Lalu Metode `getMessage()` dari objek exception `$e` digunakan untuk mengembalikan pesan kesalahan yang lebih jelas, tujuannya untuk membantu dalam memahami dan memperbaiki kesalahan yang terjadi.

Pada baris [15], syntax `?>` digunakan sebagai tag akhir atau tag penutup dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web

C. Source Code Model.php

Table 19. Source Code Modul 5 Model.php

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	
4	<code><head></code>
5	<code> <meta charset="UTF-8"></code>
6	<code> <meta http-equiv="X-UA-Compatible"</code> <code>content="IE=edge"></code>
7	<code> <meta name="viewport" content="width=device-</code> <code>width, initial-scale=1.0"></code>
8	

9	<pre> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384- 10 1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq 11 yl2QvZ6jIW3" crossorigin="anonymous"> 12 <title>Model</title> 13 </head> 14 15 <body> 16 </body> 17 18 </html> 19 <?php 20 include_once("Koneksi.php"); 21 22 function readData(\$nama_tabel) 23 { 24 require "Koneksi.php"; 25 \$stmt = \$conn->prepare("SELECT * FROM 26 \$nama_tabel"); 27 \$stmt->execute(); 28 \$result = \$stmt->fetchAll(); 29 30 if (!empty(\$result)) { 31 if (\$nama_tabel == "member") { 32 foreach (\$result as \$hasil) { 33 echo "<tr>"; 34 echo " <td class='text-center'>" . 35 \$hasil['id_member'] . "</td>"; </pre>
---	---

33	echo "<td>" . \$hasil['nama_member'] . "</td>";
34	echo "<td>" . \$hasil['nomor_member'] . "</td>";
35	echo "<td>" . \$hasil['alamat'] . "</td>";
36	echo "<td>" . \$hasil["tgl_mendaftar"] . "</td>";
37	echo "<td>" . \$hasil["tgl_terakhir_bayar"] . "</td>";
38	echo "<td>"; echo "Edit";
39	echo " "; echo "<a class='btn btn-danger' href='Member.php?id_member=" . \$hasil['id_member'] . " onclick=\"return confirm('Yakin Ingin 40 41 42 43 44 45 46 47 48 49
	Hapus?')\">Hapus"; echo "</td>"; echo "</tr>"; } } elseif (\$nama_tabel == "buku") { foreach (\$result as \$hasil) { echo "<tr>"; echo "<td class='text-center'>" . \$hasil['id_buku'] . "</td>"; echo "<td>" . \$hasil['judul_buku'] . "</td>";

50	echo "<td>" . \$hasil['penulis'] . "</td>";
51	echo "<td>" . \$hasil['penerbit'] . "</td>";
52	echo "<td>" . \$hasil["tahun_terbit"] .
53	"</td>"; echo "<td>"; echo "Edit"; echo " "; echo "Hapus"; echo "</td>"; echo "</tr>";
60	}
61	} elseif (\$nama_tabel == "peminjaman") { require "Koneksi.php"; if (isset(\$_GET['id_peminjaman'])) { 62 63 deletePeminjaman(\$_GET['id_peminjaman']); 64 } foreach (\$result as \$hasil) { 65 66 echo "<tr>"; echo "<td>" . \$hasil["id_peminjaman"] 67 . "</td>"; echo "<td>" . \$hasil["tgl_pinjam"] . 68 "</td>";

	echo "<td>" . \$hasil["tgl_kembali"] .
69	"</td>";
70	foreach (\$dataBuku as \$temp) {
	if (\$baris['id_buku'] ==
71	\$temp['id_buku']) {
	echo "<td>" .
72	\$temp['judul_buku'] . "</td>";
73	}
74	}
75	foreach (\$dataMember as \$temp) {
	if (\$baris['id_member'] ==
76	\$temp['id_member']) {
	echo "<td>" .
77	\$temp['nama_member'] . "</td>";
78	}
79	}
80	echo "<td>";
	echo "<a class='btn btn-primary'
	href='FormPeminjaman.php?id_peminjaman=" .
81	\$hasil['id_peminjaman'] . "'>Edit";
82	echo " ";
	echo "<a class='btn btn-danger'
	href='Peminjaman.php?id_peminjaman=" .
	\$hasil['id_peminjaman'] . "' onclick=\"return
83	confirm('Yakin Ingin Dihapus?')\">Hapus";
84	echo "</td>";
85	echo "</tr>";
86	}
87	}
88	}

89	}
90	
	function insertDataMember(\$nama_member,
	\$nomor_member, \$alamat, \$tgl_mendaftar,
91	\$tgl_terakhir_bayar)
92	{
	\$sql = "INSERT INTO `member` (`nama_member`,
	`nomor_member`, `alamat`, `tgl_mendaftar`,
	`tgl_terakhir_bayar`) VALUES
	(:nama_member,:nomor_member,:alamat,:tgl_mendaftar,:t
93	gl_terakhir_bayar)";
94	require "Koneksi.php";
95	\$stmt = \$conn->prepare(\$sql);
	\$result = \$stmt->execute(array(':nama_member' =>
	\$nama_member, ':nomor_member' => \$nomor_member,
	':alamat' => \$alamat, ':tgl_mendaftar' =>
	\$tgl_mendaftar, ':tgl_terakhir_bayar' =>
96	\$tgl_terakhir_bayar));
97	if (!empty(\$result)) {
98	header('location:Member.php');
99	}
100	}
101	
	function insertDataBuku(\$id_buku, \$judul_buku,
102	\$penulis, \$penerbit, \$tahun_terbit)
103	{
	\$sql = "INSERT INTO `buku` (`judul_buku`,
	`penulis`, `penerbit`, `tahun_terbit`) VALUES
104	(:judul_buku,:penulis,:penerbit,:tahun_terbit)";
105	require "Koneksi.php";

106	\$stmt = \$conn->prepare(\$sql);
	\$result = \$stmt->execute(array(':judul_buku' =>
	\$judul_buku, ':penulis' => \$penulis, ':penerbit' =>
107	\$penerbit, ':tahun_terbit' => \$tahun_terbit));
108	if (!empty(\$result)) {
109	header('location:Buku.php');
110	}
111	}
112	
	function insertDataPeminjaman(\$tgl_pinjam,
113	\$tgl_kembali, \$id_buku, \$id_member)
114	{
	\$sql = "INSERT INTO `peminjaman` (`tgl_pinjam`,
	`tgl_kembali`, `id_buku`, `id_member`) VALUES
115	(:tgl_pinjam,:tgl_kembali,:id_buku,:id_member)";
116	require "Koneksi.php";
117	\$stmt = \$conn->prepare(\$sql);
	\$result = \$stmt->execute(array(':tgl_pinjam' =>
	\$tgl_pinjam, ':tgl_kembali' => \$tgl_kembali,
118	':id_buku' => \$id_buku, ':id_member' => \$id_member));
119	if (!empty(\$result)) {
120	header('location:Peminjaman.php');
121	}
122	}
123	
124	//Edit
125	function editMember()
126	{
127	require "Koneksi.php";

```

128     $stmt = $conn->prepare("SELECT * FROM member where
129 id_member=" . $_GET["id_member"]);
130     $stmt->execute();
131     $GLOBALS['result'] = $stmt->fetchAll();
132 }
133
134 function editBuku()
135 {
136     require "Koneksi.php";
137     $stmt = $conn->prepare("SELECT * FROM buku where
138 id_buku=" . $_GET["id_buku"]);
139     $stmt->execute();
140     $GLOBALS['result'] = $stmt->fetchAll();
141 }
142
143 function editPeminjaman()
144 {
145     require "Koneksi.php";
146     $stmt = $conn->prepare("SELECT * FROM peminjaman
147 WHERE id_peminjaman =" . $_GET['id_peminjaman']);
148     $stmt->execute();
149     $GLOBALS['result'] = $stmt->fetchAll();
150 }
151
152 //Update
153 function updateMember($id_member, $nama_member,
154 $nomor_member, $alamat, $tgl_mendaftar,
155 $tgl_terakhir_bayar)
156 {
157     require "Koneksi.php";

```

	<pre> \$pdo_statement = \$conn->prepare("UPDATE member SET nama_member='" . \$nama_member . "', nomor_member='" . \$nomor_member . "', alamat='" . \$alamat . "', tgl_mendaftar='" . \$tgl_mendaftar . "', tgl_terakhir_bayar='" . 154 \$tgl_terakhir_bayar . "' where id_member=" . 155 \$id_member 156); 157 \$result = \$pdo_statement->execute(); 158 if (\$result) { 159 header('location:Member.php'); 160 } 161 } 162 function updateBuku(\$id_buku, \$judul_buku, \$penulis, 163 \$penerbit, \$tahun_terbit) 164 { 165 require "Koneksi.php"; \$pdo_statement = \$conn->prepare("UPDATE buku SET judul_buku='" . \$judul_buku . "', penulis='" . \$penulis . "', penerbit='" . \$penerbit 166 . "', tahun_terbit='" . \$tahun_terbit . "' where 167 id_buku=" . \$id_buku 168); 169 \$result = \$pdo_statement->execute(); 170 if (\$result) { 171 header('location:Buku.php'); 172 } 173 } </pre>
--	---

```

174 function updatePeminjaman($id_peminjaman,
175 $tgl_pinjam, $tgl_kembali, $id_buku, $id_member)
176 {
177     require "Koneksi.php";
178     $pdo_statement = $conn->prepare(
179         "UPDATE peminjaman SET tgl_pinjam='" .
180         $tgl_pinjam . "', tgl_kembali='" . $tgl_kembali . "',
181         id_buku='" . $id_buku . "', id_member='" . $id_member
182         . "' WHERE id_peminjaman = " . $id_peminjaman
183     );
184     $result = $pdo_statement->execute();
185     if ($result) {
186         header('location:Peminjaman.php');
187     }
188 }
189 // Hapus
190 function deleteMember($id_member)
191 {
192     require "Koneksi.php";
193     $stmt = $conn->prepare("DELETE FROM member where
194     id_member=" . $id_member);
195     $result = $stmt->execute();
196     if ($result) {
197         header('location:Member.php');
198     }
199 }
200 function deleteBuku($id_buku)
201 {

```

	require "Koneksi.php";
200	\$stmt = \$conn->prepare("DELETE FROM buku where
201	id_buku=" . \$id_buku);
202	\$result = \$stmt->execute();
203	if (\$result) {
204	header('location:Buku.php');
205	}
206	}
207	function deletePeminjaman(\$id_peminjaman)
208	{
	require "Koneksi.php";
209	\$stmt = \$conn->prepare("DELETE FROM peminjaman
210	WHERE id_peminjaman=" . \$id_peminjaman);
211	\$result = \$stmt->execute();
212	if (\$result) {
213	header('location:Peminjaman.php');
214	}
215	}
216	
217	function getMember(\$conn)
218	{
219	\$query = \$conn->prepare("SELECT * from member");
220	\$query->execute();
221	\$hasil = \$query->fetchAll(PDO::FETCH_ASSOC);
222	return \$hasil;
223	}
224	
225	function getBuku(\$conn)
226	{
227	\$query = \$conn->prepare("SELECT * from buku");

228	\$query->execute();
229	\$hasil = \$query->fetchAll(PDO::FETCH_ASSOC);
230	return \$hasil;
231	}
232	
233	function getPeminjaman(\$conn)
	{
234	\$query = \$conn->prepare("SELECT * from
235	peminjaman");
236	\$query->execute();
237	\$hasil = \$query->fetchAll(PDO::FETCH_ASSOC);
	return \$hasil;
	}

D. Pembahasan Model.php

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [4], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [5], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [6], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [7], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [8] [9], syntax

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/c
ss/bootstrap.min.css" rel="stylesheet"

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2Q
vZ6jIW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS. Terdapat penggunaan atribut penting dalam tag ini yaitu, `href=https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css` digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini

mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag `<link>` adalah `rel`, yang dengan nilai `rel="stylesheet"` menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut `integrity` seperti dalam `integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"` berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka akan dilakukan penghitungan hash dari file yang diunduh dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut `crossorigin` dengan nilai `crossorigin="anonymous"` digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan. Penggunaan, `anonymous` berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

Pada baris [10], syntax `<title>Model</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [11], syntax `</head>` merupakan tag penutup HTML dari baris 4.

Pada baris [13], syntax `<body>` tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [14], syntax `</body>` pada baris ini merupakan tag penutup dari tag pembuka baris 13.

Pada baris [16], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

Pada baris [17], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [18], syntax `include_once("Koneksi.php");` baris ini adalah konstruksi bahasa dalam PHP yang digunakan untuk menyertakan dan mengevaluasi file yang ditentukan selama eksekusi skrip. `include_once` memastikan bahwa file hanya disertakan satu kali. Apabila file sudah disertakan sebelumnya, maka `include_once` akan mengabaikannya panggilan yang dilakukan lagi. Pada program ini, file yang akan disertakan adalah `Koneksi.php`, yang berisi kode untuk mengatur koneksi ke database. Dengan menggunakan `include_once`, akan memastikan bahwa kode dalam `Koneksi.php` hanya dieksekusi sekali, bahkan jika `include_once` dipanggil beberapa kali dalam skrip yang sama. Tujuan dari penggunaannya adalah untuk mencegah potensi masalah seperti pengaturan ulang koneksi database atau definisi ulang fungsi dan variabel yang sudah ada.

Pada baris [20] [21], syntax

```
function readData($nama_tabel)
{
```

penggunaan baris tersebut adalah untuk membaca data dari tabel database yang berdasarkan nama tabel yang diberikan sebagai argumen.

Pada baris [22] [23] [24] [25], syntax

```
require "Koneksi.php";

$stmt = $conn->prepare("SELECT * FROM $nama_tabel");

$stmt->execute();
```

```
$result = $stmt->fetchAll();
```

penggunaan baris tersebut adalah sebagai perintah require "Koneksi.php"; yang digunakan untuk menyertakan file Koneksi.php ke dalam skrip saat ini. File Koneksi.php berisi definisi koneksi ke database, seperti nama host, nama database, username, password, dan konfigurasi lainnya yang diperlukan untuk mengakses database. Dengan menyertakan file ini, maka dapat menggunakan koneksi database yang telah ditetapkan di berbagai bagian skrip tanpa perlu menuliskan ulang kode koneksi setiap kali dibutuhkan. Selanjutnya, pada baris `$stmt = $conn->prepare("SELECT * FROM $nama_tabel");`, merupakan sebuah pernyataan SQL disiapkan untuk dieksekusi. Pernyataan SQL yang dipersiapkan ini adalah untuk melakukan seleksi atau SELECT dari tabel yang dinamis, di mana nama tabelnya ditentukan oleh nilai yang diberikan pada parameter `$nama_tabel`. Proses persiapan pernyataan ini menggunakan objek PDO (`$conn`) yang telah dibuat sebelumnya di file Koneksi.php. Selanjutnya, digunakannya baris `$stmt->execute();`, pernyataan SQL yang telah dipersiapkan sebelumnya dijalankan. Lalu penggunaan baris `$result = $stmt->fetchAll();` digunakan untuk mengambil semua baris hasil dari eksekusi pernyataan SQL sebelumnya dan menyimpannya dalam variabel `$result`. Fungsi `fetchAll()` digunakan untuk mengambil semua baris hasil sebagai array asosiatif yang kemudian dapat digunakan untuk pemrosesan lebih lanjut atau untuk menampilkan data sesuai kebutuhan aplikasi.

Pada baris [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43], syntax

```
if (!empty($result)) {  
    if ($nama_tabel == "member") {  
        foreach ($result as $hasil) {  
            echo "<tr>";
```

```

        echo " <td class='text-center'>" .
$hasil['id_member'] . "</td>";

        echo "<td>" . $hasil['nama_member'] .
"</td>";

        echo "<td>" . $hasil['nomor_member'] .
"</td>";

        echo "<td>" . $hasil['alamat'] . "</td>";
        echo "<td>" . $hasil["tgl_mendaftar"] .
"</td>";

        echo "<td>" . $hasil["tgl_terakhir_bayar"]
. "</td>";

        echo "<td>";

        echo "<a class='btn btn-primary'
href='FormMember.php?id_member=" . $hasil['id_member'] .
"'>Edit</a>";

        echo " ";

        echo "<a class='btn btn-danger'
href='Member.php?id_member=" . $hasil['id_member'] . "
onclick=\"return confirm('Yakin Ingin
Dihapus?')\">Hapus</a>";

        echo "</td>";

        echo "</tr>";

    }

```

penggunaan baris-baris tersebut bertanggung jawab untuk memproses dan menampilkan data yang diambil dari tabel database, khususnya tabel "member".

Penggunaan blok kode `if (!empty($result))` { digunakan untuk mengecek apakah hasil dari query tidak kosong. Fungsi `fetchAll()` mengembalikan array, dan kondisi `!empty($result)` memastikan bahwa array tersebut tidak kosong. Kemudian, `if ($nama_tabel == "member")` { digunakan untuk mengecek apakah nama tabel yang diminta adalah "member". Selanjutnya, `foreach ($result as $hasil)` { melakukan iterasi melalui setiap baris hasil query yang disimpan dalam array `$result`. Setiap elemen dari array tersebut adalah satu baris dari tabel. Penggunaan `echo "<tr>";` membuka baris baru dalam tabel HTML. Baris-baris selanjutnya menggunakan `echo` untuk menambahkan sel tabel yang berisi data anggota: `echo "<td class='text-center'>" . $hasil['id_member'] . "</td>";` menambahkan sel tabel yang berisi ID anggota (`id_member`) dengan teks yang diatur agar berada di tengah (`text-center`), `echo "<td>" . $hasil['nama_member'] . "</td>";` menambahkan sel tabel yang berisi nama anggota (`nama_member`), `echo "<td>" . $hasil['nomor_member'] . "</td>";` menambahkan sel tabel yang berisi nomor anggota (`nomor_member`), `echo "<td>" . $hasil['alamat'] . "</td>";` menambahkan sel tabel yang berisi alamat anggota (`alamat`), `echo "<td>" . $hasil['tgl_mendaftar'] . "</td>";` menambahkan sel tabel yang berisi tanggal pendaftaran anggota (`tgl_mendaftar`), dan `echo "<td>" . $hasil['tgl_terakhir_bayar'] . "</td>";` menambahkan sel tabel yang berisi tanggal terakhir pembayaran anggota (`tgl_terakhir_bayar`). Selanjutnya penggunaan `echo "<td>";` membuka sel tabel baru untuk menampung tombol-tombol aksi, yaitu Edit dan Hapus. Tombol Edit ditambahkan dengan kode `echo "Edit";`, yang mengarah ke `FormMember.php` dengan parameter `id_member` yang sesuai dengan ID anggota saat ini. Tombol ini menggunakan kelas Bootstrap `btn btn-primary` untuk styling. Spasi antara tombol Edit dan tombol Hapus ditambahkan dengan `echo " ";`. Tombol Hapus ditambahkan dengan kode `echo "Hapus";`, yang mengarah ke `Member.php` dengan parameter `id_member` yang sesuai dengan ID anggota saat ini. Tombol ini menggunakan kelas Bootstrap `btn btn-danger` untuk styling dan memiliki atribut

onclick yang menampilkan pesan konfirmasi saat tombol ditekan untuk memastikan pengguna ingin menghapus data tersebut. Selanjutnya, echo "</td>"; menutup sel tabel yang berisi tombol-tombol aksi, dan echo "</tr>"; menutup baris tabel HTML.

Pada baris [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58], syntax

```
} elseif ($nama_tabel == "buku") {  
    foreach ($result as $hasil) {  
        echo "<tr>";  
        echo "<td class='text-center'" .  
$hasil['id_buku'] . "</td>";  
        echo "<td>" . $hasil['judul_buku'] .  
"</td>";  
        echo "<td>" . $hasil['penulis'] . "</td>";  
        echo "<td>" . $hasil['penerbit'] .  
"</td>";  
        echo "<td>" . $hasil["tahun_terbit"] .  
"</td>";  
        echo "<td>";  
        echo "<a class='btn btn-primary'  
href='FormBuku.php?id_buku=" . $hasil['id_buku'] .  
'>Edit</a>";  
        echo " ";  
        echo "<a class='btn btn-danger'  
href='Buku.php?id_buku=" . $hasil['id_buku'] . "'>";
```

```

onclick=\"return          confirm('Yakin          Ingin
Dihapus?')\">Hapus</a>\";

        echo "</td>\";

        echo "</tr>\";

    }

```

penggunaan baris-baris tersebut masih dalam bagian dari fungsi `readData($nama_tabel)`, yang digunakan untuk membaca data dari berbagai tabel dalam database dan menampilkan hasilnya dalam bentuk tabel HTML. Penggunaan blok kode `} elseif ($nama_tabel == "buku") {` mengecek apakah nama tabel yang diminta adalah "buku". Apabila ya, maka blok kode ini akan dijalankan. Selanjutnya, `foreach ($result as $hasil) {` melakukan iterasi melalui setiap baris hasil query yang disimpan dalam array `$result`. Setiap elemen dari array tersebut adalah satu baris dari tabel "buku". Untuk setiap baris data, `echo "<tr>\";` membuka baris baru dalam tabel HTML. Baris-baris selanjutnya menggunakan `echo` untuk menambahkan sel tabel yang berisi data buku: `echo "<td class='text-center'>\" . $hasil['id_buku'] . "</td>\";` menambahkan sel tabel yang berisi ID buku (`id_buku`) dengan teks yang diatur agar berada di tengah (`text-center`), `echo "<td>\" . $hasil['judul_buku'] . "</td>\";` menambahkan sel tabel yang berisi judul buku (`judul_buku`), `echo "<td>\" . $hasil['penulis'] . "</td>\";` menambahkan sel tabel yang berisi nama penulis buku (`penulis`), `echo "<td>\" . $hasil['penerbit'] . "</td>\";` menambahkan sel tabel yang berisi nama penerbit buku (`penerbit`), dan `echo "<td>\" . $hasil["tahun_terbit"] . "</td>\";` menambahkan sel tabel yang berisi tahun terbit buku (`tahun_terbit`). Penggunaan `echo "<td>\";` membuka sel tabel baru untuk menampung tombol-tombol aksi (Edit dan Hapus). Tombol Edit ditambahkan dengan kode `echo "Edit\";`, yang mengarah ke `FormBuku.php` dengan parameter `id_buku` yang sesuai dengan ID buku saat ini. Tombol ini menggunakan kelas Bootstrap `btn btn-primary` untuk styling. Spasi antara tombol Edit dan tombol Hapus ditambahkan dengan `echo " \";`. Tombol Hapus

ditambahkan dengan kode echo "Hapus";, yang mengarah ke Buku.php dengan parameter id_buku yang sesuai dengan ID buku saat ini. Tombol ini menggunakan kelas Bootstrap btn btn-danger untuk styling dan memiliki atribut onclick yang menampilkan pesan konfirmasi saat tombol ditekan untuk memastikan pengguna ingin menghapus data tersebut. Selanjutnya, echo "</td>"; menutup sel tabel yang berisi tombol-tombol aksi, dan echo "</tr>"; menutup baris tabel HTML. Sehingga bagian kode ini bertujuan untuk menampilkan data dari tabel "buku"

pada baris [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88], syntax

```

} elseif ($nama_tabel == "peminjaman") {

    require "Koneksi.php";

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

        deletePeminjaman($_GET['id_peminjaman']);

    }

    foreach ($result as $hasil) {

        echo "<tr>";

        echo "<td>" . $hasil["id_peminjaman"] .
"</td>";

        echo "<td>" . $hasil["tgl_pinjam"] .
"</td>";

        echo "<td>" . $hasil["tgl_kembali"] .
"</td>";

        foreach ($dataBuku as $temp) {

```



```

                if ($baris['id_buku'] ==
$temp['id_buku']) {

                    echo "<td>" . $temp['judul_buku']

. "</td>";

                }

            }

            foreach ($dataMember as $temp) {

                if ($baris['id_member'] ==
$temp['id_member']) {

                    echo "<td>" . $temp['nama_member']

. "</td>";

                }

            }

            echo "<td>";

            echo "<a class='btn btn-primary'
href='FormPeminjaman.php?id_peminjaman="
$hasil['id_peminjaman'] . "'>Edit</a>";

            echo " ";

            echo "<a class='btn btn-danger'
href='Peminjaman.php?id_peminjaman="
$hasil['id_peminjaman'] . "' onclick=\"return
confirm('Yakin Ingin Dihapus?')\">Hapus</a>";

            echo "</td>";

            echo "</tr>";

```

```

        }

    }

}

}

```

penggunaan baris-baris tersebut digunakan untuk menangani tabel "peminjaman". Digunakannya, kode `} elseif ($nama_tabel == "peminjaman") {` mengecek apakah nama tabel yang diminta adalah "peminjaman". Apabila ya, maka blok kode ini akan dijalankan. Selanjutnya, `require "Koneksi.php";` memastikan bahwa file Koneksi.php dimasukkan kembali untuk menyediakan koneksi ke database. Kode `if (isset($_GET['id_peminjaman'])) {` mengecek apakah parameter `id_peminjaman` ada di URL (menggunakan metode GET). Jika ya, maka blok kode di dalamnya akan dijalankan, termasuk `deletePeminjaman($_GET['id_peminjaman']);` yang memanggil fungsi `deletePeminjaman` dengan parameter `id_peminjaman` dari URL untuk menghapus data peminjaman dari database. Kemudian, `foreach ($result as $hasil) {` melakukan iterasi melalui setiap baris hasil query yang disimpan dalam array `$result`, di mana setiap elemen dari array tersebut adalah satu baris dari tabel "peminjaman". Untuk setiap baris data, `echo "<tr>";` membuka baris baru dalam tabel HTML. Baris-baris berikutnya menggunakan `echo` untuk menambahkan sel tabel yang berisi data peminjaman: `echo "<td>" . $hasil["id_peminjaman"] . "</td>";` menambahkan sel tabel yang berisi ID peminjaman (`id_peminjaman`), `echo "<td>" . $hasil["tgl_pinjam"] . "</td>";` menambahkan sel tabel yang berisi tanggal peminjaman (`tgl_pinjam`), dan `echo "<td>" . $hasil["tgl_kembali"] . "</td>";` menambahkan sel tabel yang berisi tanggal pengembalian (`tgl_kembali`). Selanjutnya, penggunaan `foreach ($dataBuku as $temp) {` untuk melakukan iterasi melalui setiap elemen dalam array `$dataBuku` (yang berisi data buku). Kode `if ($baris['id_buku'] == $temp['id_buku']) {` mengecek apakah `id_buku` pada baris peminjaman cocok dengan `id_buku` pada baris buku. Apabila ya, maka `echo "<td>" . $temp['judul_buku'] . "</td>";` menambahkan sel tabel yang berisi judul buku (`judul_buku`). Begitu pula dengan penggunaan `foreach ($dataMember as`

\$temp) { melakukan iterasi melalui setiap elemen dalam array \$dataMember (yang berisi data member). Kode if (\$baris['id_member'] == \$temp['id_member']) { mengecek apakah id_member pada baris peminjaman cocok dengan id_member pada baris member. Jika ya, maka echo "<td>" . \$temp['nama_member'] . "</td>"; menambahkan sel tabel yang berisi nama member (nama_member). Lalu, penggunaan echo "<td>"; membuka sel tabel baru untuk menampung tombol-tombol aksi (Edit dan Hapus). Tombol Edit ditambahkan dengan echo "Edit";, yang mengarah ke FormPeminjaman.php dengan parameter id_peminjaman yang sesuai dengan ID peminjaman saat ini. Tombol ini menggunakan kelas Bootstrap btn btn-primary untuk styling. Spasi antara tombol Edit dan tombol Hapus ditambahkan dengan echo " ";. Tombol Hapus ditambahkan dengan echo "Hapus";, yang mengarah ke Peminjaman.php dengan parameter id_peminjaman yang sesuai dengan ID peminjaman saat ini. Tombol ini menggunakan kelas Bootstrap btn btn-danger untuk styling dan memiliki atribut onclick yang menampilkan pesan konfirmasi saat tombol ditekan untuk memastikan pengguna ingin menghapus data tersebut. Selanjutnya, echo "</td>"; menutup sel tabel yang berisi tombol-tombol aksi, dan echo "</tr>"; menutup baris tabel HTML.

Pada baris [90] [91] [92] [93] [94] [95] [96] [97] [98] [99], syntax

```
function insertDataMember($nama_member, $nomor_member,
$alamat, $tgl_mendaftar, $tgl_terakhir_bayar)
{
    $sql = "INSERT INTO `member` (`nama_member`,
`nomor_member`, `alamat`, `tgl_mendaftar`,
`tgl_terakhir_bayar`) VALUES
```

```

(:nama_member,:nomor_member,:alamat,:tgl_mendaftar,:tgl_t
erakhir_bayar)";

    require "Koneksi.php";

    $stmt = $conn->prepare($sql);

    $result = $stmt->execute(array(':nama_member' =>
$nama_member, ':nomor_member' => $nomor_member, ':alamat'
=> $alamat, ':tgl_mendaftar' => $tgl_mendaftar,
':tgl_terakhir_bayar' => $tgl_terakhir_bayar));

    if (!empty($result)) {

        header('location:Member.php');

    }

}

```

penggunaan baris-baris tersebut bertujuan untuk menambahkan data baru ke tabel member di dalam database. Fungsi ini didefinisikan dengan beberapa parameter yaitu \$nama_member, \$nomor_member, \$alamat, \$tgl_mendaftar, dan \$tgl_terakhir_bayar. Pertama, string SQL untuk melakukan operasi INSERT ke tabel member didefinisikan dengan nilai-nilai yang akan dimasukkan ke dalam kolom-kolom tabel diwakili oleh placeholder (penanda tempat) yang diawali dengan titik dua, seperti :nama_member. Kemudian, file Koneksi.php diimpor untuk memastikan koneksi ke database tersedia, yang asumsinya mendefinisikan variabel \$conn sebagai representasi koneksi ke database. Selanjutnya, menggunakan metode prepare dari objek PDO (\$conn), statement SQL dipersiapkan untuk mencegah serangan SQL injection. Setelah statement dipersiapkan, metode execute dijalankan dengan memberikan nilai-nilai sebenarnya untuk setiap placeholder dalam bentuk array asosiatif, di mana kunci adalah nama placeholder dan nilai adalah data yang akan dimasukkan ke dalam kolom tabel. Hasil eksekusi statement SQL ini disimpan dalam variabel \$result. Apabila hasil eksekusi statement tidak kosong, yang berarti operasi INSERT berhasil, browser

diarahkan ke halaman Member.php menggunakan fungsi header untuk melakukan redirect.

pada baris [101] [102] [103] [104] [105] [106] [107] [108] [109] [110], syntax

```
function insertDataBuku($id_buku, $judul_buku, $penulis,
    $penerbit, $tahun_terbit)
{
    $sql = "INSERT INTO `buku` (`judul_buku`, `penulis`,
    `penerbit`, `tahun_terbit`) VALUES
    (:judul_buku,:penulis,:penerbit,:tahun_terbit)";

    require "Koneksi.php";

    $stmt = $conn->prepare($sql);

    $result = $stmt->execute(array(':judul_buku' =>
    $judul_buku, ':penulis' => $penulis, ':penerbit' =>
    $penerbit, ':tahun_terbit' => $tahun_terbit));

    if (!empty($result)) {

        header('location:Buku.php');

    }
}
```

penggunaan kode-kode tersebut memiliki fungsi yang sama seperti insertDataMember, dimana pada baris ini insertDataBuku digunakan untuk memasukkan data buku baru ke dalam tabel buku di database. Hal tersebut dilakukan dengan menyusun pernyataan SQL INSERT dengan placeholder untuk setiap nilai yang akan dimasukkan, kemudian mengeksekusi pernyataan tersebut menggunakan koneksi database yang telah tersedia. Setelah itu, jika operasi berhasil, pengguna akan diarahkan ke halaman Buku.php.

Pada baris [112] [113] [114] [115] [116] [117] [118] [119] [120] [121], syntax

```

function insertDataPeminjaman($tgl_pinjam, $tgl_kembali,
$id_buku, $id_member)

{

    $sql = "INSERT INTO `peminjaman` (`tgl_pinjam`,
`tgl_kembali`, `id_buku`, `id_member`) VALUES
(:tgl_pinjam,:tgl_kembali,:id_buku,:id_member)";

    require "Koneksi.php";

    $stmt = $conn->prepare($sql);

    $result = $stmt->execute(array(':tgl_pinjam' =>
$tgl_pinjam, ':tgl_kembali' => $tgl_kembali, ':id_buku' =>
$id_buku, ':id_member' => $id_member));

    if (!empty($result)) {

        header('location:Peminjaman.php');

    }

}

```

penggunaan kode-kode tersebut memiliki fungsi yang sama seperti insertDataMember, dimana pada baris ini insertDataPeminjaman digunakan untuk menambahkan data peminjaman baru ke dalam tabel peminjaman di database. Hal tersebut dilakukan dengan menyusun pernyataan SQL INSERT dengan placeholder untuk setiap nilai yang akan dimasukkan, kemudian mengeksekusi pernyataan tersebut menggunakan koneksi database yang telah tersedia. Jika operasi berhasil, pengguna akan diarahkan ke halaman Peminjaman.php.

pada baris [124] [125] [126] [127] [128] [129] [130], syntax

```

function editMember()

{

```

```

        require "Koneksi.php";

        $stmt = $conn->prepare("SELECT * FROM member where
id_member=" . $_GET["id_member"]);

        $stmt->execute();

        $GLOBALS['result'] = $stmt->fetchAll();

    }

```

penggunaan baris-baris tersebut yaitu editMember digunakan untuk mengambil data anggota (member) berdasarkan ID anggota yang diberikan dalam URL (`$_GET["id_member"]`). Dengan menggunakan pernyataan `require` untuk menyertakan file `Koneksi.php`, yang diharapkan berisi kode untuk menghubungkan ke database. Ini memastikan bahwa koneksi ke database tersedia untuk eksekusi pernyataan SQL selanjutnya. Lalu, dilakukan persiapan pernyataan SQL untuk memilih semua kolom dari tabel member di mana `id_member` sama dengan nilai yang diberikan di URL. Langkah tersebut dilakukan dengan tujuan untuk mengambil data anggota yang sesuai dengan ID yang diberikan, sehingga data dapat ditampilkan atau diedit. Pernyataan SQL dieksekusi menggunakan metode `execute()`. Ini mengirimkan pernyataan SQL ke server database untuk dieksekusi dan menghasilkan hasil yang relevan. Hasil dari eksekusi pernyataan SQL tersebut diambil dengan menggunakan metode `fetchAll()` dan disimpan dalam variabel global `$GLOBALS['result']`. Dengan ini, data anggota yang sesuai dengan ID yang diberikan dapat diakses dan disimpan dalam variabel global ini.

Pada baris [132] [133] [134] [135] [136] [137] [138], syntax function editBuku()

```

{

    require "Koneksi.php";

    $stmt = $conn->prepare("SELECT * FROM buku where
id_buku=" . $_GET["id_buku"]);

    $stmt->execute();

```

```

        $GLOBALS['result'] = $stmt->fetchAll();
    }

```

penggunaan baris-baris tersebut sama dengan fungsi edit member. Fungsi editBuku digunakan untuk mengambil data buku berdasarkan ID buku yang diberikan dalam URL. Penggunaannya dilakukan dengan melakukan query SQL untuk memilih buku yang sesuai dengan ID yang diberikan, kemudian menyimpan hasilnya ke dalam variabel global \$GLOBALS['result']. Hasil tersebut kemudian dapat digunakan untuk menampilkan data buku pada halaman edit buku.

Pada baris [140] [141] [142] [143] [144] [145] [146], syntax

```

function editPeminjaman()
{
    require "Koneksi.php";

    $stmt = $conn->prepare("SELECT * FROM peminjaman WHERE
id_peminjaman =" . $_GET['id_peminjaman']);

    $stmt->execute();

    $GLOBALS['result'] = $stmt->fetchAll();
}

```

penggunaan baris-baris tersebut sama dengan fungsi edit member. Fungsi editPeminjaman digunakan untuk mengambil data peminjaman berdasarkan ID peminjaman yang diberikan dalam URL. Penggunaannya dilakukan dengan menyusun query SQL untuk memilih data peminjaman yang sesuai dengan ID yang diberikan, kemudian hasilnya disimpan dalam variabel global \$GLOBALS['result']. Hasil tersebut kemudian dapat digunakan untuk menampilkan data pada halaman edit peminjaman.

Pada baris [149] [150] [151] [152] [153] [154] [155] [156] [157] [158] [159], syntax


```

function      updateMember($id_member,      $nama_member,
$nomor_member,      $alamat,      $tgl_mendaftar,
$tgl_terakhir_bayar)

{

    require "Koneksi.php";

    $pdo_statement = $conn->prepare(

        "UPDATE member SET nama_member='" . $nama_member .
        "', nomor_member='" . $nomor_member . "', alamat='" .
        $alamat . "', tgl_mendaftar='" . $tgl_mendaftar . "',
        tgl_terakhir_bayar='" . $tgl_terakhir_bayar . "' where
        id_member=" . $id_member

    );

    $result = $pdo_statement->execute();

    if ($result) {

        header('location:Member.php');

    }

}

```

penggunaan baris-baris tersebut merupakan Fungsi updateMember yang bertujuan untuk memperbaiki data anggota (member) dalam database dengan nilai-nilai yang baru. Digunakan pernyataan require untuk menyertakan file Koneksi.php, yang diharapkan berisi kode untuk menghubungkan ke database. Hal ini diperlukan untuk memastikan koneksi ke database tersedia sehingga operasi perbaruan data dapat dilakukan. Selanjutnya, pernyataan SQL UPDATE dipersiapkan untuk mengubah data anggota dengan menggunakan nilai baru yang diberikan. Pada tahap ini, nilai baru untuk setiap kolom yang ingin diperbarui telah ditentukan. Kemudian, pernyataan SQL dieksekusi menggunakan objek PDO yang telah dipersiapkan sebelumnya. Proses

eksekusi ini akan mengirimkan pernyataan SQL ke server database untuk dieksekusi. Apabila pernyataan SQL berhasil dieksekusi dan mengembalikan nilai true, ini menandakan bahwa proses perbaruan data anggota telah berhasil. Sebagai langkah lanjutannya, pengguna akan diarahkan kembali ke halaman Member.php.

Pada baris [161] [162] [163] [164] [165] [166] [167] [168] [169] [170] [171], syntax

```
function updateBuku($id_buku, $judul_buku, $penulis,
    $penerbit, $tahun_terbit)
{
    require "Koneksi.php";

    $pdo_statement = $conn->prepare(
        "UPDATE buku SET judul_buku='" . $judul_buku . "',
        penulis='" . $penulis . "', penerbit='" . $penerbit . "',
        tahun_terbit='" . $tahun_terbit . "' where id_buku=" .
        $id_buku
    );

    $result = $pdo_statement->execute();

    if ($result) {
        header('location:Buku.php');
    }
}
```

penggunaan baris-baris tersebut sama dengan penggunaan fungsi update member. Fungsi updateBuku mengubah data buku dalam database dengan nilai baru yang diberikan. Hal ini dilakukan dengan menyusun pernyataan SQL UPDATE untuk mengubah entri buku berdasarkan ID buku yang diberikan. Setelah pernyataan SQL

dieksekusi, pengguna diarahkan kembali ke halaman Buku.php jika pembaruan berhasil.

Pada baris [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183], syntax

```
function updatePeminjaman($id_peminjaman, $tgl_pinjam,
$ttl_kembali, $id_buku, $id_member)
{
    require "Koneksi.php";

    $pdo_statement = $conn->prepare(
        "UPDATE peminjaman SET tgl_pinjam='" . $tgl_pinjam
        . "', tgl_kembali='" . $ttl_kembali . "', id_buku='" .
        $id_buku . "', id_member='" . $id_member . "' WHERE
        id_peminjaman = " . $id_peminjaman
    );

    $result = $pdo_statement->execute();

    if ($result) {
        header('location:Peminjaman.php');
    }
}
```

penggunaan dari baris-baris tersebut sama dengan penggunaan fungsi update member. Fungsi updatePeminjaman digunakan untuk mengubah data peminjaman dalam database sesuai dengan nilai-nilai yang baru diberikan. Hal ini dilakukan dengan menyusun pernyataan SQL UPDATE untuk mengubah entri peminjaman berdasarkan ID peminjaman yang diberikan. Setelah pernyataan SQL dieksekusi, pengguna diarahkan kembali ke halaman Peminjaman.php jika pembaruan berhasil.

Pada baris [186] [187] [188] [189] [190] [191] [192] [193] [194] [195], syntax

```

function deleteMember($id_member)
{
    require "Koneksi.php";

    $stmt = $conn->prepare("DELETE FROM member where
id_member=" . $id_member);

    $result = $stmt->execute();

    if ($result) {
        header('location:Member.php');
    }
}

```

penggunaan baris-baris tersebut merupakan Fungsi deleteMember berperan dalam menghapus data anggota (member) dari database berdasarkan ID member yang diberikan. Fungsi menggunakan pernyataan require untuk menyertakan file Koneksi.php, yang berisi kode untuk menghubungkan ke database. Hal ini diperlukan agar fungsi dapat berinteraksi dengan database untuk menghapus data anggota. Selanjutnya, pernyataan SQL DELETE dipersiapkan untuk menghapus data anggota berdasarkan ID member yang diberikan. Proses ini memastikan bahwa hanya data anggota dengan ID yang sesuai yang akan dihapus dari database. Setelah pernyataan SQL dipersiapkan, langkah selanjutnya adalah mengeksekusi pernyataan tersebut menggunakan objek PDO yang telah dipersiapkan sebelumnya. Proses eksekusi ini akan mengirimkan pernyataan SQL ke server database untuk dieksekusi. Apabila pernyataan SQL berhasil dieksekusi dan mengembalikan nilai true, ini menandakan bahwa proses penghapusan data anggota telah berhasil. Sebagai langkah berikutnya, pengguna akan diarahkan kembali ke halaman Member.php.

pada baris [196] [197] [198] [199] [200] [201] [202] [203] [204], syntax

```

function deleteBuku($id_buku)

```

```

{
    require "Koneksi.php";

    $stmt = $conn->prepare("DELETE FROM buku where
id_buku=" . $id_buku);

    $result = $stmt->execute();

    if ($result) {

        header('location:Buku.php');

    }

}

```

penggunaan dari baris-baris tersebut sama dengan penggunaan fungsi delete member. Fungsi deleteBuku bertujuan untuk menghapus data buku dari database berdasarkan ID buku yang diberikan. Langkah-langkahnya termasuk persiapan pernyataan SQL DELETE untuk menghapus entri buku yang sesuai dengan ID buku yang diberikan. Setelah penghapusan berhasil dieksekusi, pengguna akan diarahkan kembali ke halaman Buku.php. Dengan demikian, fungsi ini memungkinkan pengguna untuk menghapus data buku dari database.

Pada baris [205] [206] [207] [208] [209] [210] [211] [212] [213], syntax

```

function deletePeminjaman($id_peminjaman)
{
    require "Koneksi.php";

    $stmt = $conn->prepare("DELETE FROM peminjaman WHERE
id_peminjaman=" . $id_peminjaman);

    $result = $stmt->execute();

    if ($result) {

```

```

        header('location:Peminjaman.php');
    }
}

```

Penggunaan baris-baris tersebut sama dengan fungsi delete member. Fungsi deletePeminjaman menghapus data peminjaman dari database berdasarkan ID peminjaman yang diberikan, kemudian mengarahkan pengguna kembali ke halaman Peminjaman.php setelah penghapusan berhasil dieksekusi.

Pada baris [215] [216] [217] [218] [219] [220] [221],
syntax

```

function getMember($conn)
{
    $query = $conn->prepare("SELECT * from member");
    $query->execute();
    $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
    return $hasil;
}

```

penggunaan baris-baris tersebut memiliki Fungsi getMember yang memiliki tujuan untuk mengambil semua data anggota dari database. Fungsi mempersiapkan pernyataan SQL untuk memilih semua data dari tabel member. Hal ini dilakukan agar data lengkap dari semua anggota dapat diambil dari database. Setelah pernyataan SQL dipersiapkan, langkah selanjutnya adalah mengeksekusi pernyataan tersebut menggunakan koneksi database yang diberikan sebagai parameter \$conn. Proses eksekusi ini akan mengirimkan pernyataan SQL ke server database untuk dieksekusi. Apabila pernyataan dieksekusi, hasil dari eksekusi pernyataan SQL diambil menggunakan metode fetchAll dari objek statement PDO. Metode ini mengambil

semua baris hasil dan mengembalikannya dalam bentuk array. Selanjutnya hasil yang telah diambil dari database dikembalikan oleh fungsi dalam bentuk array asosiatif menggunakan mode PDO::FETCH_ASSOC. Dalam array ini, setiap elemen mewakili satu baris data dari tabel member, di mana kunci array adalah nama kolom dan nilai adalah nilai yang sesuai.

Pada baris [223] [224] [225] [226] [227] [228] [229], syntax

```
function getBuku($conn)
{
    $query = $conn->prepare("SELECT * from buku");
    $query->execute();
    $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
    return $hasil;
}
```

penggunaan baris-baris tersebut memiliki kesamaan dengan fungsi get member. Fungsi getBuku digunakan untuk mengambil semua data buku dari database dan mengembalikan hasil dalam bentuk array asosiatif. Ini memungkinkan pengguna untuk mengakses dan menggunakan data buku dalam kode mereka.

Pada baris [231] [232] [233] [234] [235] [236] [237], syntax

```
function getPeminjaman($conn)
{
    $query = $conn->prepare("SELECT * from peminjaman");
    $query->execute();
    $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
    return $hasil;
}
```

}

penggunaan baris-baris tersebut memiliki kesamaan dengan fungsi `get member`. Fungsi `getPeminjaman` digunakan untuk mengambil semua data peminjaman dari database dan mengembalikan hasil dalam bentuk array asosiatif. Hal ini memungkinkan pengguna untuk mengakses dan menggunakan data peminjaman dalam kode mereka.

E. Source Code Index.php

Table 20. Source Code Modul 5 Index.php

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	<code> <meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>
6	<code> <meta name="viewport" content="width=device-width,</code>
	<code>initial-scale=1.0"></code>
7	<code> <title>Sistem Informasi Perpustakaan</code>
	<code>Mahasiswa</title></code>
8	<code> <style></code>
9	<code> body {</code>
10	<code> background-image: url('library_page.jpg');</code>
11	<code> background-size: cover;</code>
12	<code> background-position: center;</code>
13	<code> color: #000000;</code>
14	<code> font-family: Arial, sans-serif;</code>
15	<code> margin: 0;</code>
16	<code> display: flex;</code>
17	<code> flex-direction: column;</code>
18	<code> align-items: center;</code>
19	<code> justify-content: center;</code>

20	height: 100vh;
21	overflow: hidden;
22	}
23	
24	h2 {
25	text-align: center;
26	animation: fadeInDown 1s ease-in-out;
27	}
28	
29	h1 {
30	font-size: 48px;
31	margin-bottom: 20px;
32	color: #FFF5EE;
33	text-transform: uppercase;
34	letter-spacing: 2px;
35	font-weight: bold;
36	text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
37	animation: scaleIn 1s ease-in-out;
38	}
39	
40	.grid-container {
41	display: grid;
42	grid-template-columns: repeat(3, 1fr);
43	gap: 20px;
44	width: 80%;
45	margin-top: 20px;
46	}
47	
48	.grid-item {
49	background: #FFF8DC;

50	color: #8B4513;
51	border-radius: 21px;
52	padding: 20px;
53	text-align: center;
54	animation: scaleIn 1s ease-in-out;
55	transition: transform 0.3s;
56	}
57	
58	.grid-item:hover {
59	transform: scale(1.05);
60	}
61	
62	.grid-item img {
63	transition: transform 0.3s;
64	max-width: 100%;
65	}
66	
67	.grid-item img:hover {
68	transform: scale(1.1);
69	}
70	
71	.grid-item h3 {
72	margin-bottom: 10px;
73	transition: color 0.3s;
74	}
75	
76	.grid-item:hover h3 {
77	color: #B22222;
78	}
79	

```

80     @keyframes fadeInDown {
81         0% {
82             opacity: 0;
83             transform: translateY(-20px);
84         }
85         100% {
86             opacity: 1;
87             transform: translateY(0);
88         }
89     }
90
91     @keyframes scaleIn {
92         0% {
93             opacity: 0;
94             transform: scale(0.8);
95         }
96         100% {
97             opacity: 1;
98             transform: scale(1);
99         }
100    }
101    </style>
102 </head>
103
104 <body>
105     <h1>SISTEM INFORMASI PERPUSTAKAAN MAHASISWA</h1>
106     <div class="grid-container">
107         <div class="grid-item">
108             <h3>Buku</h3>
109

```

	<pre> 111 </div> 112 <div class="grid-item"> 113 <h3>Member</h3> 115 </div> 116 <div class="grid-item"> 117 <h3>Peminjaman</h3> 119 </div> 120 </div> 121 </body> </html> </pre>
--	--

F. Output Program Index.php



Gambar 25. Screenshot Hasil Jawaban Modul 5 *Index.php*

G. Pembahasan Index.php

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [3], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [4], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [5], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [6], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [7], syntax `<title> Sistem Informasi Perpustakaan Mahasiswa</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [8], syntax `<style>` merupakan tag HTML yang digunakan untuk menentukan aturan gaya (CSS) secara internal dalam dokumen HTML.

Pada baris [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22], syntax

```
body {
```

```
background-image: url('library_page.jpg');
```

```

background-size: cover;

background-position: center;

color: #000000;

font-family: Arial, sans-serif;

margin: 0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

overflow: hidden;

}

```

penggunaan baris-baris tersebut merupakan kode css yang mengatur tampilan elemen <body> pada halaman web dengan menggunakan gambar latar belakang "library_page.jpg" dan memastikan gambar mencakup seluruh area konten dengan background-size: cover;. Background-position: center; menempatkan gambar di tengah horizontal dan vertikal. Warna teks diatur menjadi hitam (#000000) dengan font Arial atau sans-serif sebagai cadangan. Margin halaman dihapus dengan margin: 0;. Elemen body menjadi flex container dengan display: flex; dan konten ditata dalam satu kolom dengan flex-direction: column;. Align-items: center; dan justify-content: center; sehingga memastikan konten terpusat secara horizontal dan vertikal. Tinggi elemen body setara dengan 100% tinggi layar dengan height: 100vh; dan overflow: hidden; menghilangkan overflow konten. Hal ini memastikan tampilan halaman web yang teratur dengan gambar latar belakang menarik dan konten terpusat.

Pada baris [24] [25] [26] [27], syntax

```
h2 {
    text-align: center;
    animation: fadeInDown 1s ease-in-out;
}
```

penggunaan kode CSS tersebut digunakan untuk mengatur gaya tampilan teks dalam elemen <h2>. Teks tersebut akan terpusat secara horizontal dan memiliki efek animasi muncul dengan memudar dan bergeser ke bawah saat ditampilkan dengan durasi 1 detik.

Pada baris [29] [30] [31] [32] [33] [34] [35] [36] [37] [38], syntax

```
h1 {
    font-size: 48px;
    margin-bottom: 20px;
    color: #FFF5EE;
    text-transform: uppercase;
    letter-spacing: 2px;
    font-weight: bold;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
    animation: scaleIn 1s ease-in-out;
}
```

penggunaan kode CSS ini berfungsi untuk mengatur tampilan teks dalam elemen <h1>, dengan setiap properti memiliki perannya masing-masing. Baris ini menggunakan properti font-size menetapkan ukuran font menjadi 48 piksel, menciptakan teks yang besar dan menonjol. Kemudian, margin-bottom menambahkan margin bawah sebesar 20 piksel, memberikan ruang di bagian bawah elemen <h1> untuk memisahkannya dari

elemen lain. Selanjutnya, properti color mengubah warna teks menjadi putih kekuningan (#FFF5EE), menyesuaikan tampilan dengan latar belakang atau desain keseluruhan. Teks juga diberi transformasi menjadi huruf kapital dengan properti text-transform, memastikan konsistensi gaya teks. Properti letter-spacing menambahkan jarak antar huruf sebesar 2 piksel, membuat teks lebih mudah dibaca dan menarik perhatian. Untuk menekankan teks, properti font-weight mengatur ketebalan font menjadi tebal (bold). Efek bayangan teks diberikan oleh properti text-shadow dengan blur 2 piksel dan bayangan berwarna abu-abu kehitaman (rgba(0, 0, 0, 0.5)), menciptakan tampilan teks yang lebih menonjol. Selanjutnya penggunaan properti animation menambahkan animasi bernama scaleIn dengan durasi 1 detik dan efek easing-in-out, memberikan sentuhan dinamis pada teks dengan muncul secara perlahan-lahan dan mulus.

Pada baris [40] [41] [42] [43] [44] [45] [46], syntax

```
.grid-container {  
  
    display: grid;  
  
    grid-template-columns: repeat(3, 1fr);  
  
    gap: 20px;  
  
    width: 80%;  
  
    margin-top: 20px;  
  
}
```

Penggunaan kode CSS ini memiliki tujuan untuk mengatur tata letak dari kontainer grid (.grid-container). Penggunaan properti display: grid;, elemen tersebut diubah menjadi grid container, yang berarti kontennya akan diatur dalam grid. Selanjutnya, properti grid-template-columns: repeat(3, 1fr); menetapkan jumlah kolom grid menjadi 3 dengan lebar yang sama untuk setiap kolom. Nilai 1fr menunjukkan bahwa setiap kolom akan memiliki lebar yang sama dan memenuhi ruang yang tersedia. Kemudian, properti gap: 20px; menetapkan jarak antara setiap item dalam grid sebesar 20 piksel,

menciptakan ruang yang konsisten di antara elemen-elemen. Untuk menyesuaikan ukuran grid dengan tata letak halaman, lebar grid container diatur menggunakan properti `width: 80%;`, sehingga mencakup 80% dari lebar kontainer induknya. Terakhir, `margin-top: 20px;` ditetapkan untuk memberikan margin atas sebesar 20 piksel, memberikan ruang antara grid container dengan elemen lain di halaman, meningkatkan estetika dan keterbacaan tata letak secara keseluruhan.

Pada baris [48] [49] [50] [51] [52] [53] [54] [55] [56], syntax

```
.grid-item {  
  
    background: #FFF8DC;  
  
    color: #8B4513;  
  
    border-radius: 21px;  
  
    padding: 20px;  
  
    text-align: center;  
  
    animation: scaleIn 1s ease-in-out;  
  
    transition: transform 0.3s;  
  
}
```

penggunaan kode CSS tersebut mengatur tampilan setiap item dengan beberapa aturan gaya yang spesifik. Pertama, `background: #FFF8DC;` menetapkan latar belakang item menjadi warna putih kekuningan. Selanjutnya, `color: #8B4513;` menetapkan warna teks menjadi coklat tua. Efek estetik tambahan diberikan dengan `border-radius: 21px;` yang mengubah sudut elemen menjadi melengkung dengan jari-jari 21 piksel. Untuk memberikan ruang di dalam setiap item, `padding: 20px;` digunakan, menjaga konten dari tepi elemen. Selain itu, `text-align: center;` membuat teks di setiap item rata tengah secara horizontal, menciptakan tampilan yang teratur. Animasi hadir melalui `animation: scaleIn 1s ease-in-out;`, yang membuat setiap item muncul dengan efek perbesaran selama 1 detik dengan transisi yang mulus. Terakhir, `transition: transform`

0.3s; menambahkan efek animasi halus ketika item dihover dengan mengatur durasi transisi untuk perubahan transformasi selama 0.3 detik. Dengan kombinasi aturan-aturan ini, tampilan item-item tersebut dapat disesuaikan dengan presisi dan memberikan pengalaman visual yang menarik.

Pada baris [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78], syntax

```
.grid-item:hover {  
    transform: scale(1.05);  
}  
  
.grid-item img {  
    transition: transform 0.3s;  
    max-width: 100%;  
}  
  
.grid-item img:hover {  
    transform: scale(1.1);  
}  
  
.grid-item h3 {  
    margin-bottom: 10px;  
    transition: color 0.3s;  
}
```

```
.grid-item:hover h3 {
    color: #B22222;
}
```

penggunaan kode CSS ini untuk mengatur respons saat pengguna mengarahkan kursor ke atas elemen grid (.grid-item) dan gambar (img) di dalamnya. Ketika kursor diarahkan ke atas item grid, aturan .grid-item:hover { transform: scale(1.05); } membuat item tersebut sedikit membesar sebesar 5% dari ukuran normalnya. Efek transisi diterapkan pada transformasi gambar di dalam item grid dengan aturan .grid-item img { transition: transform 0.3s; max-width: 100%; }, memastikan perubahan ukuran gambar terjadi secara halus dalam waktu 0.3 detik tanpa melebihi lebar maksimum kontainer. Selain itu, ketika kursor mengarah ke atas gambar di dalam item grid, aturan .grid-item img:hover { transform: scale(1.1); } membuat gambar membesar menjadi 110% dari ukuran normalnya. Untuk elemen judul (h3) dalam item grid, aturan .grid-item h3 { margin-bottom: 10px; transition: color 0.3s; } menetapkan margin bawah sebesar 10 piksel dan memberikan efek transisi pada perubahan warna teks selama 0.3 detik. Lalu, aturan .grid-item:hover h3 { color: #B22222; } membuat warna teks judul berubah menjadi merah tua (#B22222) dengan transisi warna yang halus saat kursor diarahkan ke atasnya. Pengaturan ini memberikan pengalaman interaktif yang lebih dinamis bagi pengguna.

Pada baris [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100], syntax

```
@keyframes fadeInDown {
    0% {
        opacity: 0;
        transform: translateY(-20px);
```

```

    }

    100% {

        opacity: 1;

        transform: translateY(0);

    }

}

@keyframes scaleIn {

    0% {

        opacity: 0;

        transform: scale(0.8);

    }

    100% {

        opacity: 1;

        transform: scale(1);

    }

}

```

penggunaan kode-kode di atas menguraikan dua animasi CSS, yaitu `fadeInDown` dan `scaleIn`, yang didefinisikan melalui aturan `@keyframes`. Animasi `fadeInDown` menghasilkan efek muncul dari atas ke bawah dengan mengubah `opacity` (ketidaktransparanan) dan melakukan translasi (pemindahan posisi vertikal). Pada titik awal (0% dari animasi), elemen memiliki `opacity 0` dan translasi -20px ke atas (dimulai dari posisi 20 piksel di atas posisi akhirnya). Pada titik akhir (100% dari animasi), `opacity` elemen mencapai 1 (tidak transparan) dan translasi kembali ke posisi

normalnya (0). Animasi `scaleIn`, di sisi lain, menghasilkan efek muncul dari ukuran yang lebih kecil menjadi ukuran normal dengan mengubah `opacity` dan melakukan transformasi skala (perubahan ukuran). Pada titik awal (0% dari animasi), elemen memiliki `opacity 0` (sepenuhnya transparan) dan skala 0.8 (80% dari ukuran normal). Pada titik akhir (100% dari animasi), `opacity` elemen mencapai 1 (tidak transparan) dan skala kembali ke ukuran normal (1).

Pada baris [101], syntax `</style>` merupakan tag penutup di baris 8.

Pad baris [102], syntax `</head>` merupakan tag penutup dari baris 3.

Pada baris [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118][119] [120], syntax

```
<body>

    <h1>SISTEM INFORMASI PERPUSTAKAAN MAHASISWA</h1>

    <div class="grid-container">

        <div class="grid-item">

            <h3>Buku</h3>

            <a      href="Buku.php"></a>

        </div>

        <div class="grid-item">

            <h3>Member</h3>

            <a      href="Member.php"></a>

        </div>

        <div class="grid-item">
```

```

        <h3>Peminjaman</h3>

        <a                                href="Peminjaman.php"></a>

    </div>

</div>

</body>

```

penggunaan baris-baris tersebut digunakan untuk menyusun struktur dasar sebuah halaman web. Elemen `<body>` berfungsi sebagai wadah untuk semua konten yang akan ditampilkan. Judul utama halaman, `<h1>`, memuat teks "SISTEM INFORMASI PERPUSTAKAAN MAHASISWA". Selanjutnya, terdapat sebuah div dengan kelas "grid-container" yang digunakan untuk mengatur tata letak elemen-elemen di dalamnya menggunakan grid CSS. Di dalam grid container tersebut, terdapat dua div dengan kelas "grid-item" yang menampilkan informasi tentang "Buku" dan "Member". Setiap elemen `<div class="grid-item">` diikuti dengan judul level 3, `<h3>`, yang berisi teks "Buku", "Member", dan "Peminjaman" sebagai label untuk setiap item dalam grid. Elemen anchor, ``, mengarahkan pengguna ke halaman "Buku.php" saat diklik. Terakhir, terdapat elemen gambar, ``, yang menampilkan gambar "book.svg" dengan teks alternatif "Buku" dan tinggi 170 piksel.

Pada baris [121], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

H. Source Code Member.php

Table 21. Source Code Modul 5 *Member.php*

1	<code><?php</code>
2	<code>include_once("Koneksi.php");</code>
3	

4	function deleteMember(\$id_member)
5	{
6	global \$conn;
7	\$stmt = \$conn->prepare("DELETE FROM member WHERE id_member = ?");
8	\$stmt->execute([\$id_member]);
9	header("Location: Member.php");
10	exit();
11	}
12	
13	function readData(\$nama_tabel)
14	{
15	global \$conn;
16	\$stmt = \$conn->prepare("SELECT * FROM \$nama_tabel");
17	\$stmt->execute();
18	\$result = \$stmt->fetchAll();
19	
20	if (!empty(\$result)) {
21	if (\$nama_tabel == "member") {
22	foreach (\$result as \$hasil) {
23	echo "<tr>";
24	echo " <td class='text-center'>" . \$hasil['id_member'] . "</td>";
25	echo "<td>" . \$hasil['nama_member'] . "</td>";
26	echo "<td>" . \$hasil['nomor_member'] . "</td>";
27	echo " <td>" . \$hasil['alamat'] . "</td>";
28	

29	<pre> echo "<td>" . \$hasil["tgl_mendaftar"] . "</td>"; echo "<td>" \$hasil["tgl_terakhir_bayar"] . "</td>"; echo "<td>"; echo "Edit"; echo " "; echo "Hapus"; echo "</td>"; echo "</tr>"; } } } } ?> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> </pre>
----	---

	<link
49	href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
	integrity="sha384-
50	1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy
51	l2QvZ6jIW3" crossorigin="anonymous">
52	<title>Data Member</title>
53	<style>
54	body {
55	background-image: url('library_page.jpg');
56	background-size: cover;
57	background-position: center;
58	color: #ffffff;
59	font-family: Arial, sans-serif;
60	margin: 0;
61	display: flex;
62	flex-direction: column;
63	align-items: center;
64	justify-content: center;
65	overflow: auto;
66	}
67	table,
68	tr,
69	td {
70	border-collapse: collapse;
71	border-radius: 21px;
72	padding: 20px 40px;
73	font-size: 18px;
74	}
75	table {

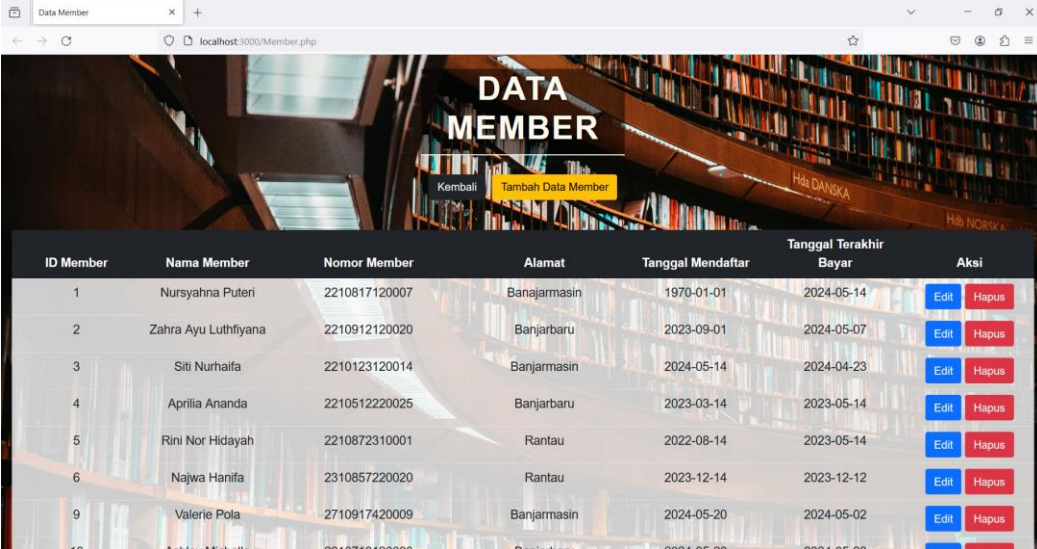
76	width: max-content;
77	border-radius: 21px;
78	background: rgba(255, 255, 255, 0.8);
79	color: black;
80	}
81	td {
82	width: 100px;
83	height: 10px;
84	text-align: center;
85	}
86	.buttons-container {
87	display: flex;
88	justify-content: flex-end;
89	margin-bottom: 20px;
90	}
91	.buttons-container a:first-child {
92	margin-right: 10px;
93	}
94	button {
95	border-radius: 10px;
96	border: 1px solid #03045e;
97	position: relative;
98	overflow: hidden;
99	transition: all 0.5s ease-in;
100	z-index: 1;
101	}
102	button::before,
103	button::after {
104	content: '';
105	position: absolute;

106	top: 0;
107	width: 0;
108	height: 100%;
109	transform: skew(15deg);
110	transition: all 0.5s;
111	overflow: hidden;
112	z-index: -1;
113	}
114	button::before {
115	left: -10px;
116	background: #537188;
117	}
118	button::after {
119	right: -10px;
120	background: #537188;
121	}
122	button:hover::before,
123	button:hover::after {
124	width: 70%;
125	}
126	button:hover span {
127	color: #e0aaff;
128	transition: 0.3s;
129	}
130	button span {
131	color: #03045e;
132	font-size: 18px;
133	transition: all 0.3s ease-in;
134	}
135	h1 {

136	font-size: 48px;
137	margin: 40px auto 20px;
138	color: #FFFAF0;
139	text-transform: uppercase;
140	letter-spacing: 2px;
141	font-weight: bold;
	text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
142	animation: fadeInDown 1s ease-in-out,
143	scaleIn 1s ease-in-out;
144	border-bottom: 2px solid #FFFAF0;
145	padding-bottom: 10px;
146	background-color: rgba(0, 0, 0, 0.5);
147	width: 80%;
148	text-align: center;
149	}
150	</style>
151	</head>
152	<body class="p-3" style="background-color: #E1D4BB;">
153	<?php
154	if (isset(\$_GET['id_member'])) {
155	deleteMember(\$_GET['id_member']);
156	}
157	?>
158	<h2>
159	<center><h1>Data Member</h1></center>
160	</h2>
	<div class="buttons-container">
161	<button class="btn btn-dark
	mb-4">Kembali</button>
162	

163	<code><button class="btn</code>
164	<code>btn-warning mb-1">Tambah Data Member</button></code>
165	<code></div></code>
166	<code><table class="table"></code>
167	<code><thead class="table-dark"></code>
	<code><tr></code>
168	<code><th class="text-center">ID Member</th></code>
	<code><th class="text-center">Nama</code>
169	<code>Member</th></code>
170	<code><th class="text-center">Nomor</code>
	<code>Member</th></code>
171	<code><th class="text-center">Alamat</th></code>
	<code><th class="text-center">Tanggal</code>
172	<code>Mendaftar</th></code>
173	<code><th class="text-center">Tanggal</code>
174	<code>Terakhir Bayar</th></code>
175	<code><th class="text-center">Aksi</th></code>
176	<code></tr></code>
177	<code></thead></code>
178	<code><tbody></code>
179	<code><?php readData("member"); ?></code>
180	<code></tbody></code>
	<code></table></code>
	<code></body></code>
	<code></html></code>

I. Output Program Member.php



ID Member	Nama Member	Nomor Member	Alamat	Tanggal Mendaftar	Tanggal Terakhir Bayar	Aksi
1	Nursyahna Puteri	2210817120007	Banjarmasin	1970-01-01	2024-05-14	Edit Hapus
2	Zahra Ayu Luthfiyana	2210912120020	Banjarbaru	2023-09-01	2024-05-07	Edit Hapus
3	Siti Nurhaifa	2210123120014	Banjarmasin	2024-05-14	2024-04-23	Edit Hapus
4	Aprilia Ananda	2210512220025	Banjarbaru	2023-03-14	2023-05-14	Edit Hapus
5	Rini Nor Hidayah	2210872310001	Rantau	2022-08-14	2023-05-14	Edit Hapus
6	Najwa Hanifa	2310857220020	Rantau	2023-12-14	2023-12-12	Edit Hapus
9	Valerie Pola	2710917420009	Banjarmasin	2024-05-20	2024-05-02	Edit Hapus
10	Ashley Michelle	2210718120000	Banjarbaru	2024-05-20	2024-05-22	Edit Hapus

Gambar 26. Screenshot Hasil Jawaban Modul 5 *Member.php*

J. Pembahasan Member.php

Pada baris [1], syntax `<?php` digunakan sebagai tag awal atau tag pembuka dari blok kode PHP. Semua kode PHP haruslah berada di antara tag ini agar dapat dieksekusi oleh server web.

Pada baris [2], syntax `include_once("Koneksi.php")`; baris ini adalah konstruksi bahasa dalam PHP yang digunakan untuk menyertakan dan mengevaluasi file yang ditentukan selama eksekusi skrip. `include_once` memastikan bahwa file hanya disertakan satu kali. Apabila file sudah disertakan sebelumnya, maka `include_once` akan mengabaikannya panggilan yang dilakukan lagi. Pada program ini, file yang akan disertakan adalah `Koneksi.php`, yang berisi kode untuk mengatur koneksi ke database. Dengan menggunakan `include_once`, akan memastikan bahwa kode dalam `Koneksi.php` hanya dieksekusi sekali, bahkan jika `include_once` dipanggil beberapa kali dalam skrip yang

sama. Tujuan dari penggunaannya adalah untuk mencegah potensi masalah seperti pengaturan ulang koneksi database atau definisi ulang fungsi dan variabel yang sudah ada.

Pada baris [4] [5] [6] [7] [8] [9] [10] [11], syntax

```
function deleteMember($id_member)

{

    global $conn;

    $stmt = $conn->prepare("DELETE FROM member WHERE
id_member = ?");

    $stmt->execute([$id_member]);

    header("Location: Member.php");

    exit();

}
```

penggunaan kode-kode di atas merupakan sebuah fungsi PHP yang bertujuan untuk menghapus data member dari database. Fungsi ini dideklarasikan dengan nama deleteMember yang menerima parameter \$id_member sebagai identifikasi unik untuk data member yang akan dihapus. Dalam fungsi tersebut, global \$conn; digunakan untuk mengakses variabel \$conn yang merupakan koneksi ke database yang sudah didefinisikan di tempat lain dalam program. Pernyataan SQL yang disiapkan (prepared statement) "\$stmt = \$conn->prepare("DELETE FROM member WHERE id_member = ?");" bertujuan untuk menghapus data member dari tabel "member" berdasarkan \$id_member yang diberikan. Setelah pernyataan SQL tersebut disiapkan, eksekusi dilakukan dengan "\$stmt->execute([\$id_member]);", di mana nilai dari \$id_member disertakan untuk dihapus. Setelah proses penghapusan selesai, pengguna diarahkan kembali ke halaman "Member.php" menggunakan fungsi header("Location: Member.php");. Lalu, "exit();" digunakan untuk menghentikan eksekusi script PHP

selanjutnya setelah melakukan pengalihan halaman, sehingga proses di sini berhenti dan tidak ada kode tambahan yang dijalankan setelahnya.

Pada baris [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40], syntax

```
function readData($nama_tabel)
{
    global $conn;

    $stmt = $conn->prepare("SELECT * FROM $nama_tabel");
    $stmt->execute();

    $result = $stmt->fetchAll();

    if (!empty($result)) {
        if ($nama_tabel == "member") {
            foreach ($result as $hasil) {
                echo "<tr>";
                echo "    <td    class='text-center'>" .
$hasil['id_member'] . "</td>";
                echo "    <td>" . $hasil['nama_member'] .
"</td>";
                echo "    <td>" . $hasil['nomor_member'] .
"</td>";
                echo "    <td>" . $hasil['alamat'] . "</td>";
                echo "    <td>" . $hasil["tgl_mendaftar"] .
"</td>";
            }
        }
    }
}
```

```

        echo "<td>" . $hasil["tgl_terakhir_bayar"]
        . "</td>";

        echo "<td>";

        echo "    <a      class='btn      btn-primary'
href='FormMember.php?id_member=" . $hasil['id_member'] .
">Edit</a>";

        echo " ";

        echo "    <a      class='btn      btn-danger'
href='Member.php?id_member=" . $hasil['id_member'] . "
onclick=\"return          confirm('Yakin          Ingin
Dihapus?')\">Hapus</a>";

        echo "</td>";

        echo "</tr>";

    }

}

}

}

?>

```

penggunaan kode-kode tersebut merupakan Fungsi `readData`, dimana berguna untuk membaca data dari sebuah tabel yang disebutkan dalam parameter `$nama_tabel`. Proses ini dimulai dengan akses global terhadap variabel `$conn`, yang merupakan koneksi ke database yang telah didefinisikan sebelumnya dalam program. Selanjutnya, sebuah pernyataan SQL yang telah disiapkan ("`SELECT * FROM $nama_tabel`") digunakan untuk memilih semua kolom dari tabel yang ditentukan. Eksekusi dari pernyataan SQL ini dilakukan dengan "`$stmt->execute()`". Hasil dari query tersebut kemudian diambil dan disimpan dalam sebuah array menggunakan fungsi "`$result = $stmt->fetchAll()`".

Sebelum melakukan iterasi, dilakukan pengujian untuk memastikan bahwa ada hasil yang ditemukan dengan menggunakan `if (!empty($result)) {...}`. Setelah itu, dilakukan iterasi melalui hasil query yang ditemukan menggunakan loop `foreach`. Setiap baris hasil dari query diakses dalam loop `foreach` dengan nama `$hasil`. Dalam setiap iterasi, data dari setiap baris hasil ditampilkan dalam bentuk baris tabel HTML. Selain itu, juga ditampilkan link untuk mengedit dan menghapus data, yang mengarahkan pengguna ke halaman `FormMember.php` untuk proses edit dan ke halaman `Member.php` untuk proses hapus. Dengan fungsi ini, memungkinkan untuk menampilkan data dari tabel yang spesifik dalam bentuk tabel HTML di halaman web.

Pada baris [42], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisikan informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [43], syntax `<html lang="en">` tag `html` merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [44], syntax `<head>` merupakan bagian kepala dari HTML. Tag `Head` berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag `Head` ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [45], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [46], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [47], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [48] [49], syntax

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/c
ss/bootstrap.min.css" rel="stylesheet"

integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2Q
vZ6jIW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS. Terdapat penggunaan atribut penting dalam tag ini yaitu, `href=https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css` digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag `<link>` adalah `rel`, yang dengan nilai `rel="stylesheet"` menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut `integrity` seperti dalam `integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"` berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka akan dilakukan penghitungan hash dari file yang diunduh

dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut `crossorigin` dengan nilai `crossorigin="anonymous"` digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan. Penggunaan, `anonymous` berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

Pada baris [50], syntax `<title>Data Member</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [51], syntax `<style>` merupakan tag HTML yang digunakan untuk menentukan aturan gaya (CSS) secara internal dalam dokumen HTML.

Pada baris [52] [53] [54] [55] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64], syntax

```
body {
```

```
    background-image: url('library_page.jpg');  
    background-size: cover;  
    background-position: center;  
    color: #ffffff;  
    font-family: Arial, sans-serif;  
    margin: 0;  
    display: flex;  
    flex-direction: column;  
    align-items: center;
```

```

        justify-content: center;

        overflow: auto;

    }

```

penggunaan baris-baris tersebut merupakan kode css yang mengatur tampilan elemen <body> pada halaman web dengan menggunakan gambar latar belakang "library_page.jpg" dan memastikan gambar mencakup seluruh area konten dengan background-size: cover;. Background-position: center; menempatkan gambar di tengah horizontal dan vertikal. Warna teks pada halaman web ditetapkan menjadi putih (#ffffff) melalui properti color: #ffffff;. Font teks ditentukan sebagai Arial atau font tanpa serifikasi dengan properti font-family: Arial, sans-serif;. Selain itu, margin: 0; menghapus margin yang ada secara default di sekitar elemen <body>. Dengan display: flex; digunakan untuk menerapkan model tata letak flexbox, posisi dan tata letak elemen-elemen dalam <body> diatur. flex-direction: column; mengatur arah tata letak flexbox menjadi kolom, sehingga elemen-elemen anak disusun secara vertikal. Properti align-items: center; dan justify-content: center; menetapkan penempatan horizontal dan vertikal elemen-elemen anak di tengah <body>. Lalu, overflow: auto; menentukan perilaku tampilan overflow untuk elemen <body>, yang berarti ketika konten lebih besar dari layar, scrollbar akan muncul untuk memungkinkan pengguna menggulir konten.

Pada baris [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75][76] [77] [78] [79][80] [81] [82] [83], syntax

```

table,

    tr,

    td {

        border-collapse: collapse;

        border-radius: 21px;
    }

```

```

padding: 20px 40px;

font-size: 18px;
}

table {

width: max-content;

border-radius: 21px;

background: rgba(255, 255, 255, 0.8);

color: black;

}

td {

width: 100px;

height: 10px;

text-align: center;

}

```

penggunaan kode CSS tersebut bertujuan untuk menetapkan gaya elemen-elemen HTML `<table>`, `<tr>`, dan `<td>`. Setiap properti CSS memiliki peran spesifik dalam menentukan penampilan dan perilaku dari elemen-elemen tersebut. Penggunaan `border-collapse: collapse;` menggabungkan batas sel-sel tabel agar terlihat seperti satu batas tunggal. Properti `border-radius: 21px;` memberikan sudut yang dibulatkan pada elemen tabel, baris, dan sel, dengan sudut dibulatkan menjadi 21 piksel. `Padding: 20px 40px;` menambahkan jarak antara isi sel dan batas sel, dengan angka pertama mengatur padding atas dan bawah, dan angka kedua mengatur padding kanan dan kiri. Ukuran font teks di dalam sel tabel ditentukan dengan `font-size: 18px;`. Properti `width: max-content;` menyesuaikan lebar tabel dengan ukuran konten maksimum di dalamnya, memungkinkan tabel memperluas sejauh yang diperlukan untuk menampung konten

tanpa memotongnya. Warna latar belakang tabel ditentukan dengan `background: rgba(255, 255, 255, 0.8);` menggunakan format RGBA, dengan nilai transparansi 0.8 untuk memberikan efek transparansi. Warna teks di dalam sel tabel ditetapkan menjadi hitam dengan `color: black;`. Lebar dan tinggi setiap sel tabel ditetapkan secara khusus dengan `width: 100px;` dan `height: 10px;`, masing-masing menjadi 100 piksel dan 10 piksel. Lalu, `text-align: center;` mengatur teks di dalam sel tabel menjadi rata tengah secara horizontal.

Pada baris [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132], syntax

```
.buttons-container {  
  
    display: flex;  
  
    justify-content: flex-end;  
  
    margin-bottom: 20px;  
  
}  
  
.buttons-container a:first-child {  
  
    margin-right: 10px;  
  
}  
  
button {  
  
    border-radius: 10px;  
  
    border: 1px solid #03045e;  
  
    position: relative;  
  
    overflow: hidden;
```



```

        transition: all 0.5s ease-in;
        z-index: 1;
    }
    button::before,
    button::after {
        content: '';
        position: absolute;
        top: 0;
        width: 0;
        height: 100%;
        transform: skew(15deg);
        transition: all 0.5s;
        overflow: hidden;
        z-index: -1;
    }
    button::before {
        left: -10px;
        background: #537188;
    }
    button::after {
        right: -10px;
        background: #537188;
    }

```

```

    }

    button:hover::before,

    button:hover::after {

        width: 70%;

    }

    button:hover span {

        color: #e0aaff;

        transition: 0.3s;

    }

    button span {

        color: #03045e;

        font-size: 18px;

        transition: all 0.3s ease-in;

    }

```

penggunaan kode-kode CSS tersebut digunakan untuk menetapkan gaya khusus pada elemen HTML yang memiliki kelas tertentu. Pada, kelas `.buttons-container` bertujuan untuk mengatur gaya sebuah kontainer yang berisi tombol-tombol. Dengan penggunaan properti `display: flex;`, kontainer tersebut diubah menjadi flex container, sehingga penataan fleksibel terhadap tombol-tombol di dalamnya. Properti `justify-content: flex-end;` digunakan untuk menempatkan tombol-tombol di ujung kanan kontainer. Selector `.buttons-container a:first-child` mengarahkan pada link pertama di dalam kontainer tombol. Properti `margin-right: 10px;` memberikan jarak 10 piksel di sebelah kanan link pertama. Gaya dari tombol itu sendiri ditentukan oleh properti `button`, termasuk `border-radius: 10px;` yang memberikan sudut yang dibulatkan pada tombol, serta `border: 1px solid #03045e;` yang menetapkan tepi tombol menjadi garis

solid dengan warna tertentu. Pseudoelemen `button::before` dan `button::after` menambahkan elemen sebelum dan sesudah isi tombol, dengan properti yang mengatur penampilannya seperti warna dan lebar. Properti-properti pada `button:hover::before` dan `button:hover::after` mengontrol penampilan elemen sebelum dan sesudah tombol saat tombol dihover. Properti `button:hover span` mengatur penampilan teks di dalam tombol saat dihover, sedangkan `button span` mengatur penampilan teks di dalam tombol secara umum, termasuk ukuran font dan efek transisi.

Pada baris [133] [134] [135] [136] [137] [138] [139] [140] [141] [142] [143] [144] [145] [146] [147], syntax

```
h1 {  
  
    font-size: 48px;  
  
    margin: 40px auto 20px;  
  
    color: #FFFAF0;  
  
    text-transform: uppercase;  
  
    letter-spacing: 2px;  
  
    font-weight: bold;  
  
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);  
  
    animation: fadeInDown 1s ease-in-out, scaleIn  
1s ease-in-out;  
  
    border-bottom: 2px solid #FFFAF0;  
  
    padding-bottom: 10px;  
  
    background-color: rgba(0, 0, 0, 0.5);  
  
    width: 80%;  
  
    text-align: center;
```

}

penggunaan kode-kode CSS tersebut bertujuan untuk menetapkan gaya khusus pada elemen `<h1>` (judul utama) dalam sebuah halaman web. Tiap properti CSS memiliki peran tertentu dalam menentukan penampilan dan perilaku dari elemen tersebut. Pertama, `font-size: 48px;` mengatur ukuran font teks menjadi 48 piksel. Properti `margin: 40px auto 20px;` menetapkan margin sebesar 40 piksel di sisi atas dan bawah, dengan margin otomatis di sisi kiri dan kanan. Warna teks ditentukan menjadi putih dengan `color: #FFFAF0;`. `Text-transform: uppercase;` mengubah semua huruf teks menjadi huruf kapital, sementara `letter-spacing: 2px;` menambahkan jarak antara setiap huruf sebesar 2 piksel. `Font-weight: bold;` memberikan efek tebal pada teks. Properti `text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);` menambahkan bayangan pada teks dengan pergeseran 2 piksel ke kanan dan 2 piksel ke bawah, radius bayangan 4 piksel, dan warna bayangan yang ditentukan oleh nilai RGBA (hitam dengan opasitas 50%). `Animation: fadeInDown 1s ease-in-out, scaleIn 1s ease-in-out;` mengatur animasi dengan efek `fadeInDown` (perubahan opasitas dari 0 menjadi 1 sambil turun) selama 1 detik dengan gerakan yang halus (`ease-in-out`), dan `scaleIn` (perbesaran elemen) selama 1 detik dengan gerakan yang halus (`ease-in-out`). Properti `border-bottom: 2px solid #FFFAF0;` menambahkan garis bawah dengan ketebalan 2 piksel dan warna putih, sementara `padding-bottom: 10px;` menambahkan ruang tambahan sebesar 10 piksel di bagian bawah elemen. `Background-color: rgba(0, 0, 0, 0.5);` mengatur warna latar belakang menjadi hitam dengan opasitas 50%. Lebar elemen ditentukan menjadi 80% dari lebar kontainer induk dengan `width: 80%;`, dan `text-align: center;` digunakan untuk memusatkan teks secara horizontal di tengah elemen.

Pada baris [148], syntax `</style>` merupakan tag penutup dari baris 51.

Pada baris [149], syntax `</head>` merupan tag penutup dari baris 44.

Pada baris [150] [151] [152] [153] [154] [155] [156] [157] [158] [159] [160] [161] [162] [163] [164] [165] [166] [167] [168] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178], syntax

```

<body class="p-3" style="background-color: #E1D4BB;">

    <?php

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

            deleteMember($_GET['id_member']);

        }

    ?>

    <h2>

        <center><h1>Data Member</h1></center>

    </h2>

    <div class="buttons-container">

        <a href="index.php"><button class="btn btn-dark
mb-4">Kembali</button></a>

        <a href="FormMember.php"><button class="btn btn-
warning mb-1">Tambah Data Member</button></a>

    </div>

    <table class="table">

        <thead class="table-dark">

            <tr>

                <th class="text-center">ID Member</th>

                <th class="text-center">Nama Member</th>

                <th class="text-center">Nomor Member</th>

                <th class="text-center">Alamat</th>

```

```
 Tanggal Mendaftar</th>   Tanggal Terakhir Bayar</th>   Aksi</th>  </tr>  </thead>  <tbody>  <?php readData("member"); ?>  </tbody>  </table> | | |
```

penggunaan kode tersebut bertujuan menampilkan data member dalam bentuk table. Penggunaan atribut kelas Bootstrap p-3 diterapkan pada elemen <body> untuk memberikan padding sebesar 3 unit (0.75 rem) di sekitar konten. Selain itu, properti inline style digunakan untuk menetapkan warna latar belakang halaman dengan kode warna #E1D4BB, menciptakan tampilan yang estetik dan terstruktur. Kemudian, dilakukan pengecekan apakah terdapat parameter id_member dalam URL. Jika parameter tersebut ditemukan, fungsi deleteMember() akan dipanggil untuk menghapus data member sesuai dengan id yang disediakan, menunjukkan fungsionalitas untuk penghapusan data yang terintegrasi dengan halaman. Judul "Data Member" ditampilkan di tengah halaman menggunakan elemen <h1> yang diletakkan di dalam elemen <center>, memberikan fokus visual yang jelas pada konten utama halaman. Selanjutnya, sebuah div dengan kelas buttons-container dibuat untuk menampung dua tombol aksi: "Kembali" dan "Tambah Data Member". Tombol-tombol tersebut mengarahkan pengguna ke halaman index.php dan FormMember.php masing-masingnya, memastikan navigasi yang mudah dan intuitif bagi pengguna. Tabel dibuat dengan menggunakan kelas Bootstrap table untuk menampilkan data member dengan

struktur yang teratur. Kolom-kolom seperti ID Member, Nama Member, Nomor Member, Alamat, Tanggal Mendaftar, Tanggal Terakhir Bayar, dan Aksi disertakan, memberikan informasi yang komprehensif kepada pengguna. Terakhir, fungsi `readData()` dipanggil untuk membaca data member dari database dan menampilkan data tersebut dalam bentuk baris-baris tabel. Setiap baris data ditampilkan menggunakan fungsi `echo` untuk setiap baris data yang diperoleh dari panggilan SQL, memastikan bahwa data ditampilkan dengan benar sesuai dengan yang ada di dalam database.

Pada baris [179], syntax `</body>` merupakan tag penutup dari baris 150.

Pada baris [180], syntax `</html>` merupakan tag penutup dari baris 43.

K. Source Code FormMember.php

Table 22. Source Code Modul 5 *FormMember.php*

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	<code> <meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>
6	<code> <meta name="viewport" content="width=device-width,</code>
	<code>initial-scale=1.0"></code>
7	<code> <link</code>
	<code>href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dis</code>
	<code>t/css/bootstrap.min.css" rel="stylesheet"</code>
8	<code> integrity="sha384-</code>
	<code>1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy</code>
	<code>l2QvZ6jIW3" crossorigin="anonymous"></code>
9	<code> <title>Form Member</title></code>
10	<code><style></code>

11	body {
12	background-image: url('library_page.jpg');
13	background-size: cover;
14	background-position: center;
15	color: #ffffff;
16	font-family: Arial, sans-serif;
17	margin: 0;
18	display: flex;
19	justify-content: center;
20	align-items: center;
21	height: 100vh;
22	overflow: auto;
23	}
24	.form-container {
25	background-color: rgba(225, 212, 187, 0.5);
26	padding: 20px;
27	border-radius: 10px;
28	}
29	input, textarea {
30	font-weight: 500;
31	font-size: 1rem;
32	color: #fff;
33	background-color: rgb(28, 28, 30);
34	box-shadow: 0 0 0.4rem rgba(0, 0, 0, 0.5),
	0 0 0 0.15rem transparent;
35	border-radius: 0.4rem;
36	border: none;
37	outline: none;
38	padding: 0.4rem;
39	transition: 0.4s;

40	}
41	input: hover, textarea: hover {
42	box-shadow: 0 0 0 0.15rem rgba(135, 207,
	235, 0.186);
43	}
44	input: focus, textarea: focus {
45	box-shadow: 0 0 0 0.15rem skyblue;
46	}
47	button {
48	display: inline-block;
49	border-radius: 10px;
50	border: 1px solid #03045e;
51	position: relative;
52	overflow: hidden;
53	transition: all 0.5s ease-in;
54	z-index: 1;
55	}
56	button::before, button::after {
57	content: '';
58	position: absolute;
59	top: 0;
60	width: 0;
61	height: 100%;
62	transform: skew(15deg);
63	transition: all 0.5s;
64	overflow: hidden;
65	z-index: -1;
66	}
67	button::before {
68	left: -10px;

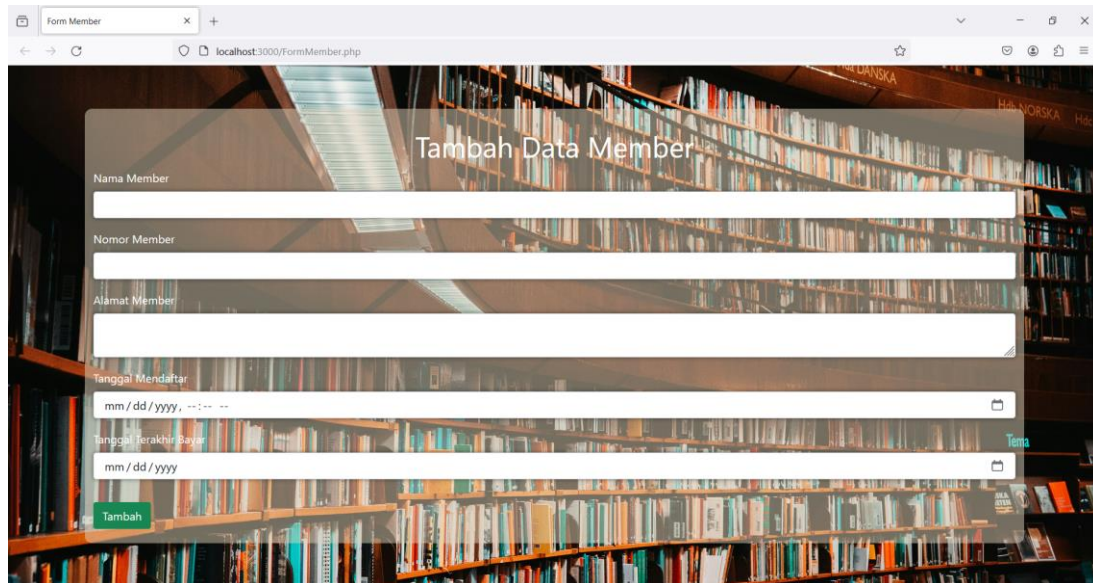
69	background: #537188;
70	}
71	button::after {
72	right: -10px;
73	background: #537188;
74	}
75	button:hover::before, button:hover::after {
76	width: 70%;
77	}
78	button:hover span {
79	color: #e0aaff;
80	transition: 0.3s;
81	}
82	button span {
83	color: #03045e;
84	font-size: 1rem;
85	transition: all 0.3s ease-in;
86	}
87	h1 {
88	text-align: center;
89	margin-top: 20px;
90	color: #fff;
91	}
92	</style>
93	</head>
94	
95	<body>
96	<?php
97	ob_start();
98	include_once('Model.php');

99	if (isset(\$_GET['id_member'])) {
100	editMember();
101	}
102	?>
103	<div class="container form-container">
104	<div class="row">
105	<form action="" method="post">
106	<h1 class="mt-2">Tambah Data
	Member</h1>
107	
108	<label for="nama_member" class="form-
	label" style="color: #fff;">Nama Member</label>
109	<input type="text" name="nama_member"
	id="nama_member" class="form-control mb-3"
110	value="<?php if
	(isset(\$_GET['id_member'])) echo
	\$result[0]['nama_member']; ?>" required>
111	
112	<label for="nomor_member" class="form-
	label" style="color: #fff;">Nomor Member</label>
113	<input type="text" name="nomor_member"
	id="nomor_member" class="form-control mb-3"
114	value="<?php if
	(isset(\$_GET['id_member'])) echo
	\$result[0]['nomor_member']; ?>" required>
115	
116	<label for="alamat" class="form-label"
	style="color: #fff;">Alamat Member</label>
117	<textarea name="alamat" id="alamat"
	class="form-control mb-3" required><?php if

	(isset(\$_GET['id_member'])) echo \$result[0]['alamat']; ?></textarea>
118	
119	<label for="tgl_mendaftar" class="form-label" style="color: #fff;">Tanggal Mendaftar</label>
120	<input type="datetime-local" name="tgl_mendaftar" id="tgl_mendaftar" class="form-control mb-3"
121	value="<?php if (isset(\$_GET['id_member'])) echo date('Y-m-d\TH:i', strtotime(\$result[0]['tgl_mendaftar'])); ?>" required>
122	
123	<label for="tgl_terakhir_bayar" class="form-label" style="color: #fff;">Tanggal Terakhir Bayar</label>
124	<input type="date" name="tgl_terakhir_bayar" id="tgl_terakhir_bayar" class="form-control mb-3"
125	value="<?php if (isset(\$_GET['id_member'])) echo \$result[0]['tgl_terakhir_bayar']; ?>" required>
126	<?php
127	ob_start();
128	if (isset(\$_GET['id_member'])) {
	echo '<button type="submit"
129	class="btn btn-success mt-3" name="update">Update</button>';
	} else {
130	
131	

	<pre> echo ' <button type="submit" class="btn btn-success mt-3" name="tambah">Tambah</button>'; } ?> </form> </div> </div> <?php ob_start(); if (isset(\$_POST['update'])) { \$tgl_mendaftar = date('Y-m-d H:i:s', 132 strtotime(\$_POST['tgl_mendaftar'])); updateMember(\$_GET['id_member'], \$_POST['nama_member'], \$_POST['nomor_member'], \$_POST['alamat'], \$tgl_mendaftar, 141 \$_POST['tgl_terakhir_bayar']); } 143 144 145 if (isset(\$_POST['tambah'])) { \$tgl_mendaftar = date('Y-m-d H:i:s', 146 strtotime(\$_POST['tgl_mendaftar'])); insertDataMember(\$_POST['nama_member'], \$_POST['nomor_member'], \$_POST['alamat'], 147 \$tgl_mendaftar, \$_POST['tgl_terakhir_bayar']); 148 } 149 ?> 150 </body> </html> </pre>
--	--

L. Output Program FormMember.php



Gambar 27. Screenshot Hasil Jawaban Modul 5 *FormMember.php*

M. Pembahasan FormMember.php

Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [3], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [4], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset

merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [5], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [6], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [7] [8], syntax

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/c
ss/bootstrap.min.css" rel="stylesheet"

integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2Q
vZ6jIW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS. Terdapat penggunaan atribut penting dalam tag ini yaitu, `href=https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css` digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag `<link>` adalah

rel, yang dengan nilai rel="stylesheet" menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut integrity seperti dalam integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka akan dilakukan penghitungan hash dari file yang diunduh dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut crossorigin dengan nilai crossorigin="anonymous" digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan. Penggunaan, anonymous berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

Pada baris [9], syntax <title>Form Member</title> tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [10], syntax <style> merupakan tag HTML yang digunakan untuk menentukan aturan gaya (CSS) secara internal dalam dokumen HTML.

Pada baris [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23], syntax

```
body {  
  
    background-image: url('library_page.jpg');  
  
    background-size: cover;  
  
    background-position: center;  
  
    color: #ffffff;
```



```

        font-family: Arial, sans-serif;

        margin: 0;

        display: flex;

        justify-content: center;

        align-items: center;

        height: 100vh;

        overflow: auto;

    }

```

penggunaan baris-baris tersebut merupakan kode css yang mengatur tampilan elemen `<body>` pada halaman web dengan menggunakan gambar latar belakang "library_page.jpg" dan memastikan gambar mencakup seluruh area konten dengan `background-size: cover;`. `Background-position: center;` menempatkan gambar di tengah horizontal dan vertikal. Warna teks pada halaman web ditetapkan menjadi putih (`#ffffff`) melalui properti `color: #ffffff;`. Font teks ditentukan sebagai Arial atau font tanpa serifikasi dengan properti `font-family: Arial, sans-serif;`. Selain itu, `margin: 0;` menghapus margin yang ada secara default di sekitar elemen `<body>`. Dengan `display: flex;` digunakan untuk mengatur elemen body menggunakan model tata letak flexbox, memudahkan pengaturan penempatan elemen di dalamnya. `justify-content: center;` dan `align-items: center;` bertujuan untuk membuat konten di tengah secara horizontal dan vertikal. Tinggi elemen body ditetapkan sebagai 100% dari tinggi viewport dengan `height: 100vh;`, memastikan halaman selalu memiliki tinggi yang sama dengan tinggi layar pengguna. Terakhir, `overflow: auto;` digunakan untuk mengatur perilaku overflow jika konten melebihi ukuran viewport, sehingga scroll vertikal akan muncul jika konten tidak muat dalam satu layar.

Pada baris [24] [25][26] [27], syntax

```
.form-container {
```

```
background-color: rgba(225, 212, 187, 0.5);  
  
padding: 20px;  
  
border-radius: 10px;
```

penggunaan kode-kode CSS tersebut bertujuan untuk menentukan tampilan dari kontainer formulir pada halaman web. Setiap properti CSS yang digunakan memiliki peran penting dalam menentukan penampilan visual dari kontainer formulir tersebut. Penggunaan, background-color ditetapkan dengan nilai RGBA (Red, Green, Blue, Alpha) yang menghasilkan warna latar belakang untuk kontainer formulir. Dalam kasus ini, warna yang ditentukan adalah kombinasi merah (225), hijau (212), dan biru (187), dengan tingkat transparansi sebesar 0.5. Penggunaan transparansi memungkinkan latar belakang halaman untuk sedikit terlihat melalui kontainer formulir, memberikan tampilan yang lebih menarik secara visual. Selanjutnya, padding ditetapkan sebesar 20 piksel, yang menambahkan ruang pola di sekitar konten dalam kontainer formulir. Hal ini membantu menjaga jarak antara konten formulir dan tepi kontainer, memberikan tampilan yang lebih teratur dan mudah dibaca bagi pengguna. Selanjutnya, border-radius digunakan untuk mengatur sudut lengkung kontainer formulir, sehingga sudut-sudutnya menjadi lebih bulat. Dalam contoh ini, sudut-sudut kontainer formulir diberi lengkungan sebesar 10 piksel.

Pada baris [29] [30] [31] [32] [33] [34] [35][36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46], syntax

```
input, textarea {  
  
    font-weight: 500;  
  
    font-size: 1rem;  
  
    color: #fff;  
  
    background-color: rgb(28, 28, 30);
```

```

        box-shadow: 0 0 0.4rem rgba(0, 0, 0, 0.5), 0 0
0 0.15rem transparent;

        border-radius: 0.4rem;

        border: none;

        outline: none;

        padding: 0.4rem;

        transition: 0.4s;
    }

    input:hover, textarea:hover {

        box-shadow: 0 0 0 0.15rem rgba(135, 207, 235,
0.186);
    }

    input:focus, textarea:focus {

        box-shadow: 0 0 0 0.15rem skyblue;
    }

```

Penggunaan kode-kode CSS tersebut untuk mengatur tampilan dan perilaku elemen input dan textarea dalam halaman web. Pertama-tama, selektor CSS input dan textarea digunakan untuk menargetkan semua elemen input dan textarea di halaman HTML. Selanjutnya, properti-properti seperti font-weight, font-size, color, dan background-color digunakan untuk menyesuaikan tampilan teks dan latar belakang elemen. Efek bayangan diterapkan menggunakan properti box-shadow, yang memberikan tampilan tiga dimensi pada elemen dengan bayangan yang terdefinisi dengan baik. Properti-properti seperti border-radius, border, outline, dan padding digunakan untuk menyesuaikan tampilan dan ukuran elemen, sementara properti transition menambahkan efek transisi yang halus ketika ada perubahan dalam tampilan elemen.

Pada saat kursor berada di atas elemen, properti hover ditetapkan untuk memberikan efek visual tambahan, seperti bayangan highlight. Ketika elemen menerima fokus, properti focus ditetapkan untuk menyoroti elemen dengan bayangan yang berbeda, menunjukkan bahwa elemen sedang aktif untuk menerima input dari pengguna.

Pada baris [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62][63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86], syntax

```
button {  
  
    display: inline-block;  
  
    border-radius: 10px;  
  
    border: 1px solid #03045e;  
  
    position: relative;  
  
    overflow: hidden;  
  
    transition: all 0.5s ease-in;  
  
    z-index: 1;  
  
}  
  
button::before, button::after {  
  
    content: '';  
  
    position: absolute;  
  
    top: 0;  
  
    width: 0;  
  
    height: 100%;  
  
    transform: skew(15deg);
```

```

        transition: all 0.5s;

        overflow: hidden;

        z-index: -1;
    }

    button::before {

        left: -10px;

        background: #537188;
    }

    button::after {

        right: -10px;

        background: #537188;
    }

    button:hover::before, button:hover::after {

        width: 70%;
    }

    button:hover span {

        color: #e0aaff;

        transition: 0.3s;
    }

    button span {

        color: #03045e;

        font-size: 1rem;
    }

```

```
transition: all 0.3s ease-in;

}
```

penggunaan kode-kode CSS di atas adalah definisi gaya untuk tombol (button) pada halaman web. Pertama-tama, properti display diterapkan dengan nilai inline-block, memungkinkan tombol memiliki lebar dan tinggi yang didefinisikan serta dapat disusun secara horizontal. Selanjutnya, border-radius digunakan untuk memberikan sudut bulatan pada tombol dengan efek melengkung sebesar 10 piksel, menciptakan tampilan yang lebih estetik. Border solid dengan ketebalan 1 piksel dan warna biru tua diterapkan pada tombol, memberikan garis tepi yang terlihat jelas. Properti position diatur sebagai relative untuk memungkinkan penyesuaian posisi relatif terhadap elemen lain di halaman web. Overflow tombol diatur menjadi hidden, yang berarti konten di luar batas tombol tidak akan ditampilkan. Ini berguna untuk menjaga tampilan tombol tetap rapi dan terkendali. Selanjutnya, efek transisi diterapkan pada semua perubahan properti dengan durasi 0.5 detik dan fungsi timing ease-in. Z-index ditetapkan sebagai 1 untuk mengontrol urutan tumpukan dalam lapisan tata letak, memastikan tombol berada di atas elemen lain jika diperlukan. Pseudo-elements ::before dan ::after dibuat untuk tombol dengan konten kosong dan posisi absolut. Lebar pseudo-elements diatur sebagai 0, sehingga tidak terlihat, sedangkan tinggi diatur sebagai 100% agar memiliki tinggi yang sama dengan tombol. Transformasi skew sebesar 15 derajat diterapkan pada pseudo-elements, memberikan efek miring yang menarik pada tombol. Pseudo-elements ::before dan ::after ditempatkan masing-masing di sebelah kiri dan kanan tombol. Saat tombol dihover, lebar pseudo-elements diperbesar menjadi 70% untuk memberikan efek miring yang lebih terlihat, dan warna teks tombol diubah menjadi ungu muda. Selain itu, efek transisi diterapkan pada perubahan properti dengan durasi 0.3 detik saat tombol dihover, memberikan interaksi yang responsif dan halus. Lalu, properti untuk konten teks tombol ditetapkan dengan ukuran font 1 rem dan warna teks biru tua, menjaga konsistensi dengan gaya keseluruhan halaman web.

Pada baris [87] [88] [89] [90] [91], syntax

```

h1 {

    text-align: center;

    margin-top: 20px;

    color: #fff;

}

```

penggunaan kode-kode CSS tersebut bertujuan untuk menyesuaikan gaya elemen <h1> dalam halaman web. Penggunaan, properti text-align diterapkan dengan nilai center, yang mengatur teks dalam elemen <h1> agar terpusat secara horizontal. Hal ini menciptakan tampilan yang seragam dan estetik untuk judul halaman. Selanjutnya, margin-top diterapkan dengan nilai 20 piksel, memberikan jarak atau ruang kosong sebesar 20 piksel di bagian atas elemen <h1>. Penggunaan margin ini memungkinkan penyesuaian posisi elemen terhadap elemen lain di sekitarnya, memberikan fleksibilitas dalam tata letak halaman. Lalu, properti color diatur dengan nilai #fff, yang mewakili warna putih.

Pada baris [92], syntax </style> merupakan tag penutup dari baris 10.

Pada baris [93], syntax </head> merupakan tag penutup dari baris 3.

Pada baris [95], syntax <body> tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [96] [97] [98] [99] [100] [101] [102], syntax

```

<?php

    ob_start();

    include_once('Model.php');

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

```

```

        editMember();
    }

?>

```

penggunaan kode tersebut memiliki beberapa fungsi penting dalam pengelolaan halaman web. Penggunaan, fungsi `ob_start()` digunakan untuk memulai output buffering, sebuah teknik yang berguna dalam mengelola output pada PHP. Dengan memulai output buffering, semua output yang dihasilkan oleh skrip PHP akan dialirkan ke dalam buffer sementara sebelum dikirimkan ke klien, memungkinkan manipulasi atau pengaturan pesan kesalahan sebelum ditampilkan kepada pengguna. Selanjutnya, perintah `include_once('Model.php')` berfungsi untuk memasukkan file `Model.php` ke dalam skrip saat ini. File `Model.php` berisi definisi-definisi yang diperlukan seperti fungsi-fungsi untuk operasi data, koneksi database, atau definisi kelas yang dibutuhkan. Dengan memasukkan file `Model.php`, maka bisa mengakses kode-kode tersebut dan menggunakan fungsionalitas yang disediakan. Lalu, terdapat sebuah kondisional `if` yang memeriksa apakah parameter `GET` dengan nama `id_member` telah diset. Apabila parameter ini ada, maka fungsi `editMember()` akan dipanggil. Hal ini digunakan ketika halaman perlu memuat data anggota yang akan diedit sebelum menampilkan formulir atau halaman untuk mengeditnya. Dengan memanggil fungsi `editMember()`, data anggota bisa dimuat dan halaman bisa disiapkan dengan data yang sesuai untuk diedit.

Pada baris [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132] [133] [134] 135] [136], syntax

```

<div class="container form-container">

    <div class="row">

        <form action="" method="post">

            <h1 class="mt-2">Tambah Data Member</h1>

```



```
<label for="nama_member" class="form-label" style="color: #fff;">Nama Member</label>
```

```
<input type="text" name="nama_member" id="nama_member" class="form-control mb-3"
```

```
value="<?php if (isset($_GET['id_member'])) echo $result[0]['nama_member']; ?>" required>
```

```
<label for="nomor_member" class="form-label" style="color: #fff;">Nomor Member</label>
```

```
<input type="text" name="nomor_member" id="nomor_member" class="form-control mb-3"
```

```
value="<?php if (isset($_GET['id_member'])) echo $result[0]['nomor_member']; ?>" required>
```

```
<label for="alamat" class="form-label" style="color: #fff;">Alamat Member</label>
```

```
<textarea name="alamat" id="alamat" class="form-control mb-3" required><?php if (isset($_GET['id_member'])) echo $result[0]['alamat']; ?></textarea>
```

```
<label for="tgl_mendaftar" class="form-label" style="color: #fff;">Tanggal Mendaftar</label>
```

```

        <input                                type="datetime-local"
name="tgl_mendaftar"      id="tgl_mendaftar"      class="form-
control mb-3"

        value="<?php                                if
(isset($_GET['id_member']))      echo      date('Y-m-d\TH:i',
strtotime($result[0]['tgl_mendaftar'])); ?>" required>

        <label                                for="tgl_terakhir_bayar"
class="form-label" style="color: #fff;">Tanggal Terakhir
Bayar</label>

        <input                                type="date"
name="tgl_terakhir_bayar"      id="tgl_terakhir_bayar"
class="form-control mb-3"

        value="<?php                                if
(isset($_GET['id_member']))      echo
$result[0]['tgl_terakhir_bayar']; ?>" required>

        <?php

        ob_start();

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

            echo '<button type="submit" class="btn
btn-success mt-3" name="update">Update</button>';

        } else {

            echo '<button type="submit" class="btn
btn-success mt-3" name="tambah">Tambah</button>';

        }

```

```

        ?>

    </form>

</div>

</div>

```

penggunaan kode-kode di atas adalah formulir HTML yang berfungsi untuk menambah atau mengedit data member. Secara rinci, elemen `<div class="container form-container">` menggunakan kelas CSS Bootstrap untuk membuat kontainer formulir dengan latar belakang semi-transparan. Selanjutnya, `<form action="" method="post">` digunakan untuk membuat formulir dengan metode POST, dengan aksi formulir yang ditetapkan kosong (""), yang berarti data formulir akan dikirimkan ke halaman yang sama. Judul formulir ditampilkan dengan menggunakan elemen `<h1 class="mt-2">Tambah Data Member</h1>`. Setiap elemen `<input>` dan `<textarea>` memiliki atribut `name` dan `id` untuk mengidentifikasi elemen tersebut dalam pengiriman formulir dan manipulasi JavaScript. Nilai-nilai atribut `value` diatur untuk menetapkan nilai default ke dalam input saat ini sesuai dengan data yang diedit, jika dalam mode edit. Kondisional PHP `if (isset($_GET['id_member']))` digunakan untuk menentukan apakah formulir dalam mode "tambah" atau "edit". Jika `id_member` tersedia di URL (dalam mode edit), maka tombol akan menampilkan "Update", jika tidak, maka tombol akan menampilkan "Tambah".

Pada baris [137] [138] [139] [140] [141] [142] [143] [144] [145] [146] [147] [148], syntax

```

<?php

    ob_start();

    if (isset($_POST['update'])) {

        $tgl_mendaftar = date('Y-m-d H:i:s',
strtotime($_POST['tgl_mendaftar']));

```

```

        updateMember($_GET['id_member'],
$_POST['nama_member'],                $_POST['nomor_member'],
$_POST['alamat'],                    $tgl_mendaftar,
$_POST['tgl_terakhir_bayar']);

    }

    if (isset($_POST['tambah'])) {

        $tgl_mendaftar      =      date('Y-m-d      H:i:s',
strtotime($_POST['tgl_mendaftar']));

        insertDataMember($_POST['nama_member'],
$_POST['nomor_member'], $_POST['alamat'], $tgl_mendaftar,
$_POST['tgl_terakhir_bayar']);

    }

?>

```

penggunaan kode PHP ini bertanggung jawab untuk mengelola aksi formulir yang dikirimkan melalui metode POST. Pertama, <?php menandakan awal dari blok kode PHP. Kemudian, ob_start(); digunakan untuk memulai output buffering, yang memungkinkan output PHP disimpan ke dalam buffer sementara sebelum dikirimkan ke browser. Selanjutnya, if (isset(\$_POST['update'])) { digunakan untuk memeriksa apakah tombol "Update" telah diklik di formulir. Jika ya, maka kode dalam blok if akan dieksekusi. Di dalamnya, tanggal mendaftar dikonversi ke format yang dapat disimpan di database, dan fungsi updateMember() dipanggil untuk memperbarui data member yang ada. Kemudian, if (isset(\$_POST['tambah'])) { memeriksa apakah tombol "Tambah" telah diklik. Jika ya, maka tanggal mendaftar dikonversi lagi dan fungsi insertDataMember() dipanggil untuk menambahkan data member baru ke database. Lalu, ?> menandakan akhir dari blok kode PHP.

pada baris [149], syntax </body> merupakan tag penutup dari baris 95.

Pada baris [150], syntax `</html>` merupakan tag penutup dari baris 2.

N. Source Code Buku.php

Table 23. Source Code Modul 5 Buku.php

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	<code> <meta http-equiv="X-UA-Compatible"</code> <code>content="IE=edge"></code>
6	<code> <meta name="viewport" content="width=device-width,</code> <code>initial-scale=1.0"></code>
7	<code> <link</code> <code>href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dis</code> <code>t/css/bootstrap.min.css" rel="stylesheet"</code> <code>integrity="sha384-</code> <code>1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy</code> <code>l2QvZ6jIW3" crossorigin="anonymous"></code>
8	<code> <title>Data Buku</title></code>
9	<code> <style></code>
10	<code> body {</code>
11	<code> background-image: url('library_page.jpg');</code>
12	<code> background-size: cover;</code>
13	<code> background-position: center;</code>
14	<code> color: #ffffff;</code>
15	<code> font-family: Arial, sans-serif;</code>
16	<code> margin: 0;</code>
17	<code> display: flex;</code>
18	<code> flex-direction: column;</code>
19	<code> align-items: center;</code>

20	justify-content: center;
21	overflow: auto;
22	}
23	table, tr, td {
24	border-collapse: collapse;
25	border-radius: 21px;
26	padding: 20px 40px;
27	font-size: 18px;
28	}
29	table {
30	width: max-content;
31	border-radius: 21px;
32	background: rgba(255, 255, 255, 0.8);
33	color: black;
34	}
35	td {
36	width: 100px;
37	height: 10px;
38	text-align: center;
39	}
40	.buttons-container {
41	display: flex;
42	justify-content: flex-end;
43	margin-bottom: 20px;
44	}
45	.buttons-container a:first-child {
46	margin-right: 10px;
47	}
48	button {
49	border-radius: 10px;

50	border: 1px solid #03045e;
51	position: relative;
52	overflow: hidden;
53	transition: all 0.5s ease-in;
54	z-index: 1;
55	}
56	button::before, button::after {
57	content: '';
58	position: absolute;
59	top: 0;
60	width: 0;
61	height: 100%;
62	transform: skew(15deg);
63	transition: all 0.5s;
64	overflow: hidden;
65	z-index: -1;
66	}
67	button::before {
68	left: -10px;
69	background: #537188;
70	}
71	button::after {
72	right: -10px;
73	background: #537188;
74	}
75	button:hover::before, button:hover::after {
76	width: 70%;
77	}
78	button:hover span {
79	color: #e0aaff;

80	transition: 0.3s;
81	}
82	button span {
83	color: #03045e;
84	font-size: 18px;
85	transition: all 0.3s ease-in;
86	}
87	h1 {
88	font-size: 48px;
89	margin: 40px auto 20px;
90	color: #FFFAF0;
91	text-transform: uppercase;
92	letter-spacing: 2px;
93	font-weight: bold;
94	text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
95	animation: fadeInDown 1s ease-in-out,
	scaleIn 1s ease-in-out;
96	border-bottom: 2px solid #FFFAF0;
97	padding-bottom: 10px;
98	background-color: rgba(0, 0, 0, 0.5);
99	width: 80%;
100	text-align: center;
101	}
102	</style>
103	</head>
104	
105	<body class="p-3" style="background-color: #E1D4BB;">
106	<?php
107	include_once("Koneksi.php");
108	


```

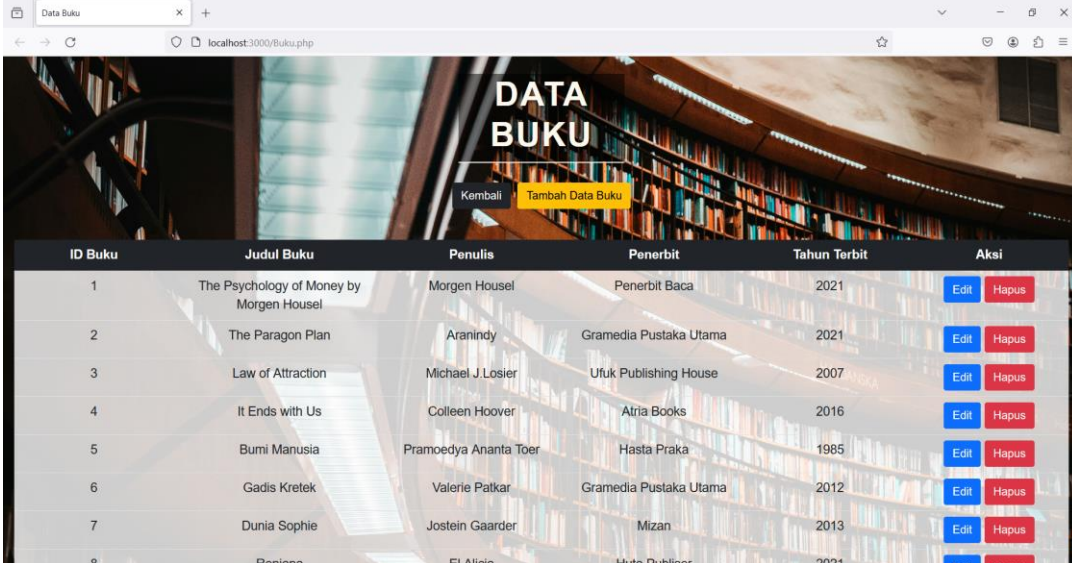
109 function deleteBuku($id_buku)
110 {
111     global $conn;
112     $stmt = $conn->prepare("DELETE FROM peminjaman WHERE
id_buku = ?");
113     $stmt->execute([$id_buku]);
114     $stmt = $conn->prepare("DELETE FROM buku WHERE
id_buku = ?");
115     $stmt->execute([$id_buku]);
116
117     header("Location: Buku.php");
118     exit();
119 }
120
121 function readData($nama_tabel)
122 {
123     global $conn;
124     $stmt = $conn->prepare("SELECT * FROM $nama_tabel");
125     $stmt->execute();
126     $result = $stmt->fetchAll();
127
128     if (!empty($result)) {
129         if ($nama_tabel == "buku") {
130             foreach ($result as $hasil) {
131                 echo "<tr>";
132                 echo "    <td    class='text-center'>" .
$hasil['id_buku'] . "</td>";
133                 echo "    <td>" . $hasil['judul_buku'] .
"</td>";
134

```

135	<pre> echo "<td>" . \$hasil['penulis'] . "</td>"; echo "<td>" . \$hasil['penerbit'] . "</td>"; echo "<td>" . \$hasil["tahun_terbit"] . "</td>"; echo "<td>"; echo "Edit"; echo " "; echo "Hapus"; echo "</td>"; echo "</tr>"; } } } } if (isset(\$_GET['id_buku'])) { deleteBuku(\$_GET['id_buku']); } ?> <h2> <center><h1>Data Buku</h1></center> </h2> </pre>
136	
137	
138	
139	
140	
141	
142	
143	
144	
145	
146	
147	
148	
149	
150	
151	
152	
153	
154	
155	
156	

157	<div class="buttons-container">
	<button class="btn btn-dark
158	mb-4">Kembali</button>
	<button class="btn btn-
159	warning mb-1">Tambah Data Buku</button>
160	</div>
161	<table class="table">
162	<thead class="table-dark">
163	<tr>
164	<th class="text-center">ID Buku</th>
165	<th class="text-center">Judul Buku</th>
166	<th class="text-center">Penulis</th>
167	<th class="text-center">Penerbit</th>
168	<th class="text-center">Tahun
	Terbit</th>
169	<th class="text-center">Aksi</th>
170	</tr>
171	</thead>
172	<tbody>
173	<?php readData("buku"); ?>
174	</tbody>
175	</table>
176	</body>
	</html>

O. Output Program Buku.php



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/Buku.php'. The page title is 'DATA BUKU'. Below the title, there are two buttons: 'Kembali' and 'Tambah Data Buku'. The main content is a table with the following data:

ID Buku	Judul Buku	Penulis	Penerbit	Tahun Terbit	Aksi
1	The Psychology of Money by Morgen Housel	Morgen Housel	Penerbit Baca	2021	Edit Hapus
2	The Paragon Plan	Aranindy	Gramedia Pustaka Utama	2021	Edit Hapus
3	Law of Attraction	Michael J Losier	Ufuk Publishing House	2007	Edit Hapus
4	It Ends with Us	Colleen Hoover	Alria Books	2016	Edit Hapus
5	Bumi Manusia	Pramoedya Ananta Toer	Hasta Praka	1985	Edit Hapus
6	Gadis Kretek	Valerie Patkar	Gramedia Pustaka Utama	2012	Edit Hapus
7	Dunia Sophie	Jostein Gaarder	Mizan	2013	Edit Hapus
8	Renjana	El Alcia	Huta Publisher	2021	Edit Hapus

Gambar 28. Screenshot Hasil Jawaban Modul 5 Buku.php

P. Pembahasan Buku.php

Pada Pada baris [1], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [2], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [3], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [4], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [5], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [6], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [7], syntax

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/c
ss/bootstrap.min.css" rel="stylesheet"

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2Q
vZ6jIW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS. Terdapat penggunaan atribut penting dalam tag ini yaitu, `href=https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css` digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini

mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag <link> adalah rel, yang dengan nilai rel="stylesheet" menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut integrity seperti dalam integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka akan dilakukan penghitungan hash dari file yang diunduh dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut crossorigin dengan nilai crossorigin="anonymous" digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan. Penggunaan, anonymous berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

Pada baris [8], syntax <title>Data Buku</title> tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [9] – [102], syntax

```
<style>

    body {

        background-image: url('library_page.jpg');

        background-size: cover;

        background-position: center;

        color: #ffffff;
```

```

    font-family: Arial, sans-serif;

    margin: 0;

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    overflow: auto;
}

table, tr, td {

    border-collapse: collapse;

    border-radius: 21px;

    padding: 20px 40px;

    font-size: 18px;
}

table {

    width: max-content;

    border-radius: 21px;

    background: rgba(255, 255, 255, 0.8);

    color: black;
}

td {

    width: 100px;

```

```

        height: 10px;

        text-align: center;
    }

.buttons-container {

    display: flex;

    justify-content: flex-end;

    margin-bottom: 20px;
}

.buttons-container a:first-child {

    margin-right: 10px;
}

button {

    border-radius: 10px;

    border: 1px solid #03045e;

    position: relative;

    overflow: hidden;

    transition: all 0.5s ease-in;

    z-index: 1;
}

button::before, button::after {

    content: '';

    position: absolute;

```



```

    top: 0;

    width: 0;

    height: 100%;

    transform: skew(15deg);

        transition: all 0.5s;

    overflow: hidden;

    z-index: -1;
}

button::before {

    left: -10px;

    background: #537188;

}

button::after {

    right: -10px;

    background: #537188;

}

button:hover::before, button:hover::after {

    width: 70%;

}

button:hover span {

    color: #e0aaff;

    transition: 0.3s;

```

```

    }

    button span {

        color: #03045e;

        font-size: 18px;

        transition: all 0.3s ease-in;

    }

    h1 {

        font-size: 48px;

        margin: 40px auto 20px;

        color: #FFFAF0;

        text-transform: uppercase;

        letter-spacing: 2px;

        font-weight: bold;

        text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);

        animation: fadeInDown 1s ease-in-out, scaleIn
1s ease-in-out;

        border-bottom: 2px solid #FFFAF0;

        padding-bottom: 10px;

        background-color: rgba(0, 0, 0, 0.5);

        width: 80%;

        text-align: center;

    }

```

```
</style>
```

penggunaan kode-kode css terseut untuk mengatur beberap agaya pada web. Pengaturan css pada `body { ... }`: digunakan untu mendefinisikan gaya untuk elemen `<body>`, termasuk pengaturan gambar latar belakang, warna teks, dan tata letak. Latar belakang menggunakan gambar dari 'library_page.jpg', dengan ukuran yang menyesuaikan ukuran halaman dan posisi yang diatur ke tengah. Warna teks diatur menjadi putih, menggunakan font Arial atau sans-serif, dan tata letak halaman ditata agar berada di tengah dengan menggunakan flexbox. Lalu `table, tr, td { ... }`: Mendefinisikan gaya untuk elemen tabel dan selnya, termasuk pengaturan tata letak, padding, dan ukuran teks. Padding diatur menjadi 20px di atas dan bawah serta 40px di kiri dan kanan untuk setiap sel. Selanjutnya, gaya untuk elemen-elemen tabel seperti `<table>`, `<tr>`, dan `<td>` memastikan tampilan tabel yang rapi dengan padding yang merata di setiap selnya. Tambahan gaya untuk elemen tabel mengatur lebar maksimum, warna latar belakang, dan penataan teks. Selain itu, wadah untuk tombol (`buttons-container`) didefinisikan dengan penataan `flex-end` untuk meletakkan tombol di bagian kanan wadah dengan margin bawah yang sesuai. Efek visual tambahan pada tombol diperkenalkan menggunakan pseudo-element (`::before` dan `::after`), dengan latar belakang berwarna tertentu yang mengikuti interaksi mouse saat tombol dihover. Gaya untuk elemen `<h1>` menetapkan ukuran font besar, warna, efek bayangan, dan efek animasi agar judul halaman menonjol dan menarik perhatian pengguna.

Pada baris [103], syntax `<head>` untuk menutup tag pada baris 3.

Pada baris [105] – [151], syntax `<body class="p-3" style="background-color: #E1D4BB;">`

```
<?php
```

```
include_once("Koneksi.php");
```

```
function deleteBuku($id_buku)
```

```

{

    global $conn;

    $stmt = $conn->prepare("DELETE FROM peminjaman WHERE
id_buku = ?");

    $stmt->execute([$id_buku]);

    $stmt = $conn->prepare("DELETE FROM buku WHERE id_buku
= ?");

    $stmt->execute([$id_buku]);

    header("Location: Buku.php");

    exit();

}

```

```

function readData($nama_tabel)
{

    global $conn;

    $stmt = $conn->prepare("SELECT * FROM $nama_tabel");

    $stmt->execute();

    $result = $stmt->fetchAll();

    if (!empty($result)) {

        if ($nama_tabel == "buku") {

```

```

        foreach ($result as $hasil) {

            echo "<tr>";

            echo "    <td    class='text-center'>" .
$hasil['id_buku'] . "</td>";

            echo "    <td>" . $hasil['judul_buku'] .
"</td>";

            echo "<td>" . $hasil['penulis'] . "</td>";

            echo "    <td>" . $hasil['penerbit'] .
"</td>";

            echo "    <td>" . $hasil["tahun_terbit"] .
"</td>";

            echo "<td>";

            echo "    <a    class='btn    btn-primary'
href='FormBuku.php?id_buku=" . $hasil['id_buku'] .
"'>Edit</a>";

            echo " ";

            echo "    <a    class='btn    btn-danger'
href='Buku.php?id_buku=" . $hasil['id_buku'] . " '
onclick=\"return    confirm('Yakin    Ingin
Dihapus?')\">Hapus</a>";

            echo "</td>";

            echo "</tr>";

        }

    }

```

```

    }

}

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

    deleteBuku($_GET['id_buku']);

}

?>

```

penggunaan dari `<body class="p-3" style="background-color: #E1D4BB;">`: Membuat elemen `<body>` dengan kelas `p-3` yang berfungsi untuk memberikan padding sebesar 3 unit ke semua sisi elemen, dan properti gaya inline untuk mengatur warna latar belakang menjadi `#E1D4BB`. Selanjutnya `<?php include_once("Koneksi.php"); ?>`: menggunakan pernyataan PHP `include_once` untuk menyisipkan file `Koneksi.php` ke dalam halaman. Ini memungkinkan kode dalam file `Koneksi.php` dijalankan di halaman saat ini. Untuk penggunaan function `deleteBuku($id_buku) { ... }`: sebagai pendefinisian sebuah fungsi PHP bernama `deleteBuku` yang menerima parameter `$id_buku`. Fungsi ini bertujuan untuk menghapus data buku dari database berdasarkan `$id_buku` yang diberikan. Penggunaan fungsi `function readData($nama_tabel) { ... }`: mendefinisikan sebuah fungsi PHP bernama `readData` yang menerima parameter `$nama_tabel`. Fungsi ini bertujuan untuk membaca data dari tabel yang disebutkan dalam parameter dan menampilkannya dalam bentuk tabel HTML. global `$conn`:: digunakan untuk mendeklarasikan variabel global `$conn` yang merepresentasikan koneksi ke database. Variabel ini akan digunakan di dalam fungsi-fungsi yang ada di dalam file ini. Penggunaan `$stmt = $conn->prepare("DELETE FROM peminjaman WHERE id_buku = ?");`: untuk mempersiapkan pernyataan SQL untuk menghapus data buku dari tabel `peminjaman` berdasarkan `$id_buku`. `$stmt->execute([$id_buku]);`: menjalankan pernyataan SQL untuk menghapus data buku berdasarkan `$id_buku` yang diberikan. Lalu `header("Location: Buku.php");`: mengarahkan pengguna ke halaman

Buku.php setelah proses penghapusan data buku selesai. `exit();` bertujuan untuk menghentikan eksekusi skrip PHP secara paksa setelah mengarahkan pengguna ke halaman Buku.php. Dan `if (isset($_GET['id_buku'])) { ... }`: memeriksa apakah parameter `id_buku` telah diset dalam URL. Apabila ya, maka akan memanggil fungsi `deleteBuku` untuk menghapus buku berdasarkan id yang diberikan.

Pada baris [153] – [174], syntax

```
<h2>

    <center><h1>Data Buku</h1></center>

</h2>

<div class="buttons-container">

    <a href="index.php"><button class="btn btn-dark
mb-4">Kembali</button></a>

    <a href="FormBuku.php"><button class="btn btn-
warning mb-1">Tambah Data Buku</button></a>

</div>

<table class="table">

    <thead class="table-dark">

        <tr>

            <th class="text-center">ID Buku</th>

            <th class="text-center">Judul Buku</th>

            <th class="text-center">Penulis</th>

            <th class="text-center">Penerbit</th>

            <th class="text-center">Tahun Terbit</th>

            <th class="text-center">Aksi</th>
```

```

        </tr>

    </thead>

    <tbody>

        <?php readData("buku"); ?>

    </tbody>

</table>

```

penggunaan baris-baris tersebut merupakan kombinasi dari kode HTML dan PHP. Penggunaan, tag <h2> digunakan untuk membuat heading level 2. Selanjutnya, div dengan kelas CSS buttons-container berfungsi sebagai wadah untuk tombol-tombol yang akan ditempatkan di atas atau bawah halaman web. Kemudian, tautan menggunakan tag <a> mengarahkan pengguna ke halaman tertentu, dengan masing-masing tombol yang terkait memiliki gaya yang ditentukan oleh kelas CSS dari framework Bootstrap. Tag <table> digunakan untuk membuat tabel, dengan kelas CSS table untuk menambahkan gaya default dari Bootstrap. Bagian header tabel ditandai dengan tag <thead> yang diberi kelas table-dark untuk memberikan latar belakang yang gelap. Setiap sel heading dalam tabel ditandai dengan tag <th>, di mana kelas CSS text-center digunakan untuk menengahkan teks di dalam sel. Lalu, pengeksekusian fungsi PHP readData("buku") bertujuan untuk membaca dan menampilkan data dari tabel buku dalam bentuk baris-baris pada tabel HTML.

Pada baris [175], syntax </body> merupakan tag penutup dari batis 105.

Pada baris [176], syntax </html> merupakan tag penutup dari baris 2.

Q. Source Code FormBuku.php

Table 24. Source Code Modul 5 FormBuku.php

1	<?php
2	ob_start();
3	include 'Koneksi.php';

4	include 'Model.php';
5	?>
6	
7	<!DOCTYPE html>
8	<html lang="en">
9	<head>
10	<meta charset="UTF-8">
	<meta http-equiv="X-UA-Compatible"
11	content="IE=edge">
	<meta name="viewport" content="width=device-width,
12	initial-scale=1.0">
	<link
13	href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dis
	t/css/bootstrap.min.css" rel="stylesheet"
	integrity="sha384-
14	1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy
	l2QvZ6jIW3" crossorigin="anonymous">
	<title>Formulir Buku</title>
15	<style>
16	body {
17	background-image: url('library_page.jpg');
18	background-size: cover;
19	background-position: center;
20	color: #ffffff;
21	font-family: Arial, sans-serif;
22	margin: 0;
23	display: flex;
24	justify-content: center;
25	align-items: center;
26	height: 100vh;

27	overflow: auto;
28	}
29	.form-container {
30	background-color: rgba(225, 212, 187, 0.5);
31	padding: 20px;
32	border-radius: 10px;
33	}
34	input {
35	font-weight: 500;
36	font-size: .8vw;
37	color: #fff;
38	background-color: rgb(28, 28, 30);
39	box-shadow: 0 0 .4vw rgba(0, 0, 0, 0.5), 0
40	0 0 .15vw transparent;
	border-radius: 0.4vw;
41	border: none;
42	outline: none;
43	padding: 0.4vw;
44	transition: .4s;
45	}
46	input:hover {
47	box-shadow: 0 0 0 .15vw rgba(135, 207, 235,
48	0.186);
	}
49	input:focus {
50	box-shadow: 0 0 0 .15vw skyblue;
51	}
52	button {
53	display: inline-block;
54	border-radius: 10px;

55	border: 1px solid #03045e;
56	position: relative;
57	overflow: hidden;
58	transition: all 0.5s ease-in;
59	z-index: 1;
60	}
61	button::before,
62	button::after {
63	content: '';
64	position: absolute;
65	top: 0;
66	width: 0;
67	height: 100%;
68	transform: skew(15deg);
69	transition: all 0.5s;
70	overflow: hidden;
71	z-index: -1;
72	}
73	button::before {
74	left: -10px;
75	background: #537188;
76	}
77	button::after {
78	right: -10px;
79	background: #537188;
80	}
81	button:hover::before,
82	button:hover::after {
83	width: 70%;
84	}

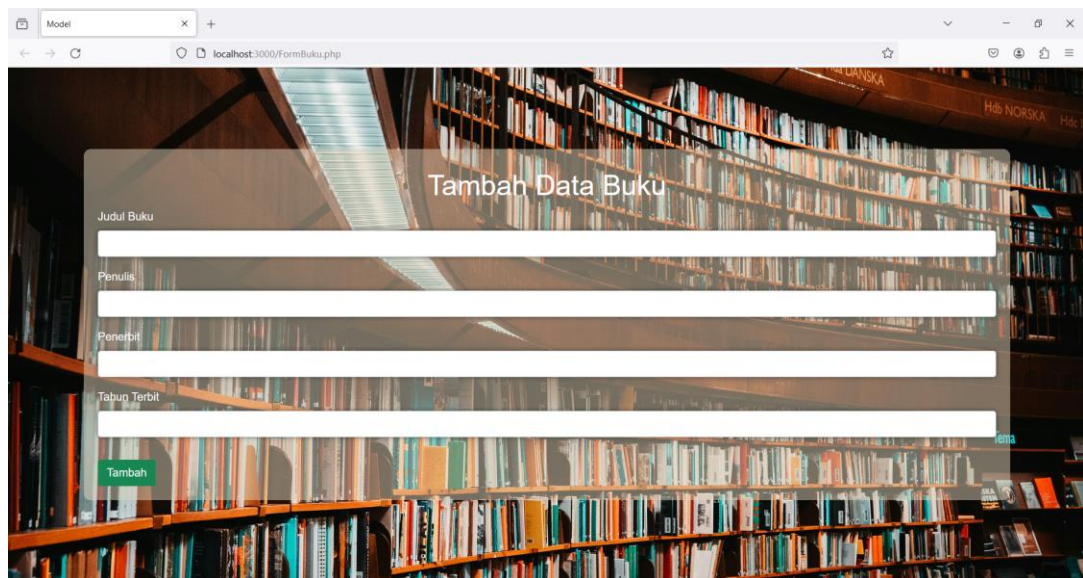
85	button:hover span {
86	color: #e0aaff;
87	transition: 0.3s;
88	}
89	button span {
90	color: #03045e;
91	font-size: 18px;
92	transition: all 0.3s ease-in;
93	}
94	</style>
95	</head>
96	
97	<body class="p-3">
98	<?php
99	ob_start();
100	include_once('Model.php');
101	if (isset(\$_GET['id_buku'])) {
102	editBuku();
103	}
104	?>
105	
106	<div class="container form-container">
107	<div class="row">
108	<form action="" method="post">
109	<h1 style="text-align: center;"
110	class="mt-2" style="color: #fff;">Tambah Data Buku</h1>
111	<label for="judul_buku" class="form-
112	label" style="color: #fff;">Judul Buku</label>

113	<code><input type="text" name="judul_buku"</code>
	<code>id="judul_buku" class="form-control mb-3" value="<?php</code>
114	<code>if (isset(\$_GET['id_buku']))</code>
	<code>echo \$result[0]["judul_buku"] ?>"</code>
	<code>required></code>
115	
116	<code><label for="penulis" class="form-</code>
	<code>label" style="color: #fff;">Penulis</label></code>
117	<code><input type="text" name="penulis"</code>
	<code>id="penulis" class="form-control mb-3" value="<?php if</code>
	<code>(isset(\$_GET['id_buku']))</code>
118	<code>echo \$result[0]["penulis"] ?>"</code>
119	<code>required></code>
120	<code><label for="penerbit" class="form-</code>
	<code>label" style="color: #fff;">Penerbit</label></code>
121	<code><input type="text" name="penerbit"</code>
	<code>id="penerbit" class="form-control mb-3" value="<?php if</code>
	<code>(isset(\$_GET['id_buku']))</code>
122	<code>echo \$result[0]["penerbit"] ?>"</code>
	<code>required></code>
123	
	<code><label for="tahun_terbit"</code>
	<code>class="form-label" style="color: #fff;">Tahun</code>
	<code>Terbit</label></code>
124	<code><input type="text"</code>
	<code>name="tahun_terbit" id="tahun_terbit" class="form-</code>
	<code>control mb-3" value="<?php if (isset(\$_GET['id_buku']))</code>
125	<code>echo \$result[0]["tahun_terbit"] ?>"</code>
	<code>required></code>

126	<?php
127	ob_start();
128	if (isset(\$_GET['id_buku'])) {
129	echo "<button type=\"submit\"
130	class=\"btn btn-success mt-3\" name=\"update\"> Update
131	</button>";
132	} else {
133	echo "<button type=\"submit\"
134	class=\"btn btn-success mt-3\" name=\"tambah\"> Tambah
135	</button>";
136	}
137	?>
138	</div>
139	</div>
140	</form>
141	<?php
142	ob_start();
143	if (isset(\$_POST['update'])) {
144	updateBuku(\$_GET['id_buku'],
145	\$_POST['judul_buku'], \$_POST['penulis'],
146	\$_POST['penerbit'], \$_POST['tahun_terbit']);
147	}
148	if (isset(\$_POST['tambah'])) {
149	insertDataBuku(\$_POST['id_buku'],
150	\$_POST['judul_buku'], \$_POST['penulis'],
151	\$_POST['penerbit'], \$_POST['tahun_terbit']);
152	}
153	?>
154	</body>

149	</html>
-----	---------

R. Output Program FormBuku.php



Gambar 29. Screenshot Hasil Jawaban Modul 5 FormBuku.php

S. Pembahasan FormBuku.php

Pada baris [1] [2] [3] [4] [5], syntax

```
<?php
```

```
ob_start();  
  
include 'Koneksi.php';  
  
include 'Model.php';  
  
?>
```

Penggunaan pada baris-baris tersebut adalah untuk mengelola output dan menyertakan file eksternal yang diperlukan untuk koneksi dan operasi database. Penggunaan fungsi `ob_start()` digunakan untuk memulai output buffering, yang memungkinkan skrip PHP menahan pengiriman output ke browser hingga `ob_end_flush()` atau `ob_flush()` dipanggil. Selanjutnya, pernyataan `include 'Koneksi.php';` menyertakan file `Koneksi.php`, yang berisi kode untuk menghubungkan aplikasi ke database. Dengan menyertakan file ini, skrip PHP mendapatkan akses ke variabel dan fungsi yang mendefinisikan koneksi database, sehingga memungkinkan operasi database dilakukan di seluruh skrip. Selain itu, pernyataan `include 'Model.php';` menyertakan file `Model.php`, yang berisi definisi fungsi-fungsi untuk berinteraksi dengan database, seperti fungsi untuk mengambil data, menyimpan data, memperbarui data, dan menghapus data. Dengan menyertakan file ini, skrip PHP dapat menggunakan fungsi-fungsi tersebut untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada data.

Pada baris [7], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [8], syntax `<html lang="en">` tag `html` merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [9], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag

Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [10], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [11], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [12], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsicve. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [13] [14], syntax

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/c
ss/bootstrap.min.css" rel="stylesheet"

integrity="sha384-
1BmE4kWbQ78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2Q
vZ6jIW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS. Terdapat penggunaan atribut penting dalam tag ini yaitu, `href=https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css` digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag `<link>` adalah `rel`, yang dengan nilai `rel="stylesheet"` menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut `integrity` seperti dalam `integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"` berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka akan dilakukan penghitungan hash dari file yang diunduh dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut `crossorigin` dengan nilai `crossorigin="anonymous"` digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan. Penggunaan, `anonymous` berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

Pada baris [15], syntax `<title>Formulir Buku</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [16] – [95], syntax

```
<style>

    body {

        background-image: url('library_page.jpg');
```

```

background-size: cover;

background-position: center;

color: #ffffff;

font-family: Arial, sans-serif;

margin: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

overflow: auto;
}

.form-container {

background-color: rgba(225, 212, 187, 0.5);

padding: 20px;

border-radius: 10px;
}

input {

font-weight: 500;

font-size: .8vw;

color: #fff;

background-color: rgb(28, 28, 30);

```

```

        box-shadow: 0 0 .4vw rgba(0, 0, 0, 0.5), 0 0 0
.15vw transparent;

        border-radius: 0.4vw;

        border: none;

        outline: none;

        padding: 0.4vw;

        transition: .4s;
    }

    input:hover {

        box-shadow: 0 0 0 .15vw rgba(135, 207, 235,
0.186);
    }

    input:focus {

        box-shadow: 0 0 0 .15vw skyblue;
    }

    button {

        display: inline-block;

        border-radius: 10px;

        border: 1px solid #03045e;

        position: relative;

        overflow: hidden;

        transition: all 0.5s ease-in;
    }

```

```

        z-index: 1;
    }
    button::before,
    button::after {
        content: '';
        position: absolute;
        top: 0;
        width: 0;
        height: 100%;
        transform: skew(15deg);
        transition: all 0.5s;
        overflow: hidden;
        z-index: -1;
    }
    button::before {
        left: -10px;
        background: #537188;
    }
    button::after {
        right: -10px;
        background: #537188;
    }

```

```

        button:hover::before,
        button:hover::after {
            width: 70%;
        }

        button:hover span {
            color: #e0aaff;
            transition: 0.3s;
        }

        button span {
            color: #03045e;
            font-size: 18px;
            transition: all 0.3s ease-in;
        }
    </style>

```

penggunaan baris-baris tersebut untuk mengaatur tampilan halaman web. Selector `body` menetapkan gaya untuk elemen `body`, termasuk gambar latar belakang dari file `'library_page.jpg'` yang disetel agar mencakup seluruh area tanpa mengubah proporsi dan diposisikan di tengah. Warna teks diatur menjadi putih, menggunakan font Arial atau sans-serif, dengan margin dihapus dan tata letak halaman ditata di tengah menggunakan flexbox. Tinggi elemen `body` diatur sebesar 100% dari tinggi viewport dan konten yang meluap diizinkan untuk digulir. Selector `.form-container` menetapkan warna latar belakang dengan transparansi 50%, menambahkan ruang dalam sebesar 20px di sekitar konten, dan membulatkan sudut elemen dengan radius 10px. Selector `input` menetapkan berbagai gaya untuk elemen input, termasuk ketebalan font, ukuran font relatif terhadap lebar viewport, warna teks putih, warna latar belakang gelap,

bayangan di sekitar elemen, sudut yang membulat, dan berbagai properti lain untuk memastikan tampilan yang konsisten dan responsif. Selain itu, gaya tambahan diterapkan saat elemen input di-hover dan difokuskan, dengan menambahkan bayangan berwarna tertentu. Selector button dan pseudo-elementnya menetapkan berbagai gaya untuk tombol, termasuk membuat elemen menjadi inline-block, membulatkan sudut, menetapkan border, dan menambahkan efek transisi. Pseudo-element button::before dan button::after digunakan untuk menambahkan elemen tambahan sebelum dan sesudah konten tombol, dengan berbagai properti posisi dan transformasi untuk menciptakan efek visual yang menarik saat tombol di-hover. Selector button:hover::before dan button:hover::after mengubah lebar elemen pseudo menjadi 70% saat tombol di-hover, sementara button:hover span mengubah warna teks menjadi #e0aaff dan menetapkan durasi transisi selama 0.3 detik. Selector button span menetapkan warna teks span menjadi #03045e, ukuran font 18px, dan durasi serta jenis transisi untuk memastikan perubahan yang halus dan konsisten.

Pada baris [96], syntax `</head>` merupakan tag penutup dari baris 9.

Pada baris [98], syntax `<body class="p-3">` merupakan framework css.

Pada baris [99] [100] [101] [102] [103] [104] [105], syntax

```
<?php

    ob_start();

    include_once('Model.php');

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

        editBuku();

    }

?>
```

Penggunaan baris-baris tersebut digunakan dalam sebuah skrip PHP untuk mengelola data buku dalam suatu aplikasi web. Penggunaan baris pertama yaitu, `<?php ob_start();`

?>, memulai output buffering. Ketika output buffering diaktifkan, alih-alih mengirim output (seperti HTML atau teks lainnya) langsung ke browser, PHP akan menahan output tersebut dalam buffer internal. Hal ini memungkinkan untuk melakukan perubahan atau pemeriksaan pada output sebelum dikirim ke browser, dan berguna untuk mengelola header HTTP serta mencegah output yang tidak diinginkan sebelum header dikirim. Selanjutnya, `include_once('Model.php');` menyertakan file `Model.php` satu kali ke dalam skrip. File `Model.php` biasanya berisi fungsi-fungsi yang digunakan untuk berinteraksi dengan database atau model data lainnya. Dengan menggunakan `include_once`, akan memastikan bahwa file `Model.php` hanya dimuat satu kali, menghindari potensi masalah seperti deklarasi ulang fungsi atau variabel yang sama. Kemudian, blok `if (isset($_GET['id_buku'])) { ... }` memeriksa apakah parameter `id_buku` ada dalam query string URL. Query string adalah bagian dari URL yang dimulai dengan tanda `?` dan diikuti oleh pasangan kunci-nilai. Ini memastikan bahwa `id_buku` telah disediakan oleh pengguna atau aplikasi sebelum mencoba melakukan operasi yang bergantung pada parameter ini. Jika `id_buku` tersedia, fungsi `editBuku()` dipanggil. Fungsi ini kemungkinan besar diambil dari `Model.php` dan berisi logika untuk mengedit data buku berdasarkan `id_buku` yang diterima dari query string. Ini memungkinkan pengeditan data buku yang ada di database atau sistem penyimpanan lainnya berdasarkan ID buku yang disediakan.

Pada baris [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132] [133] [134] [135] [136] [137], syntax

```
<div class="container form-container">

    <div class="row">

        <form action="" method="post">

            <h1 style="text-align: center;" class="mt-2" style="color: #fff;">Tambah Data Buku</h1>
```



```
        <label    for="judul_buku"    class="form-  
label" style="color: #fff;">Judul Buku</label>
```

```
        <input    type="text"    name="judul_buku"  
id="judul_buku" class="form-control mb-3" value="<?php if  
(isset($_GET['id_buku']))
```

```
        echo    $result[0]["judul_buku"]    ?>"  
required>
```

```
        <label    for="penulis"    class="form-  
label" style="color: #fff;">Penulis</label>
```

```
        <input    type="text"    name="penulis"  
id="penulis" class="form-control mb-3" value="<?php if  
(isset($_GET['id_buku']))
```

```
        echo    $result[0]["penulis"]    ?>"  
required>
```

```
        <label    for="penerbit"    class="form-  
label" style="color: #fff;">Penerbit</label>
```

```
        <input    type="text"    name="penerbit"  
id="penerbit" class="form-control mb-3" value="<?php if  
(isset($_GET['id_buku']))
```

```
        echo    $result[0]["penerbit"]    ?>"  
required>
```

```

                                <label                                for="tahun_terbit"
class="form-label"                                style="color:                                #fff;">Tahun
Terbit</label>

                                <input                                type="text"
name="tahun_terbit" id="tahun_terbit" class="form-control
mb-3" value="<?php if (isset($_GET['id_buku']))

                                echo $result[0]["tahun_terbit"] ?>"
required>

                                <?php

                                ob_start();

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

                                echo                                "<button                                type=\"submit\"
class=\"btn btn-success mt-3\" name=\"update\"> Update
</button>";

                                } else {

                                echo                                "<button                                type=\"submit\"
class=\"btn btn-success mt-3\" name=\"tambah\"> Tambah
</button>";

                                }

                                ?>

                                </div>

                                </div>

                                </form>

```

penggunaan baris-baris tersebut merupakan bagian dari sebuah formulir HTML yang digunakan untuk menambah atau mengedit data buku dalam suatu aplikasi web.

Penggunaan, `<div class="container form-container">` digunakan untuk membuat sebuah div dengan kelas container dan form-container. Div ini bertujuan untuk menempatkan elemen-elemen formulir dalam tata letak yang teratur dan terpusat di tengah halaman, sehingga meningkatkan tampilan dan kegunaan formulir dengan memberikan tata letak yang baik dan mudah diakses. Selanjutnya, `<div class="row">` digunakan untuk membuat sebuah div dengan kelas row. Div ini biasanya digunakan untuk menempatkan elemen-elemen formulir dalam satu baris secara horizontal, memastikan elemen-elemen formulir ditempatkan dalam satu baris sehingga tata letak formulir menjadi lebih terstruktur dan mudah dibaca. Kemudian, `<form action="" method="post">` membuat sebuah form HTML dengan method POST, yang digunakan untuk mengirim data ke server. Ini memungkinkan pengguna untuk memasukkan data buku dan mengirimkannya ke server untuk diproses. Judul "Tambah Data Buku" ditampilkan dengan menggunakan heading level 1 (h1) dan ditengahkan secara horizontal menggunakan `<h1 style="text-align: center;" class="mt-2" style="color: #fff;">Tambah Data Buku</h1>`, memberikan informasi kepada pengguna bahwa formulir ini digunakan untuk menambah data buku. Setiap field data buku memiliki label dan input field, seperti `<label for="judul_buku" class="form-label" style="color: #fff;">Judul Buku</label>` dan `<input type="text" name="judul_buku" id="judul_buku" class="form-control mb-3" value="<?php if (isset($_GET['id_buku'])) echo $result[0]['judul_buku'] ?>" required>`, yang memungkinkan pengguna untuk memasukkan informasi yang diperlukan dengan cara yang terstruktur dan mudah dipahami. Jika formulir digunakan untuk mengedit data buku, maka nilai field akan diisi dengan data buku yang sudah ada, sehingga memudahkan pengguna dalam melakukan perubahan. Tombol `<button>` digunakan untuk menyimpan perubahan jika pengguna melakukan edit data buku, atau untuk menambahkan data baru jika formulir digunakan untuk menambah data.

Pada baris [138] [139] [140] [141] [142] [143] [144] [145] [146] [147], syntax

```
<?php
```

```
ob_start();
```

```

        if (isset($_POST['update'])) {

            updateBuku($_GET['id_buku'],
$_POST['judul_buku'], $_POST['penulis'],
$_POST['penerbit'], $_POST['tahun_terbit']);

        }

        if (isset($_POST['tambah'])) {

            insertDataBuku($_POST['id_buku'],
$_POST['judul_buku'], $_POST['penulis'],
$_POST['penerbit'], $_POST['tahun_terbit']);

        }

    ?>

```

penggunaan baris-baris tersebut merupakan bagian dari logika pemrosesan formulir PHP. Fungsi `ob_start()` digunakan untuk memulai output buffering, yang memungkinkan penangkapan sementara dari output yang dihasilkan oleh skrip PHP sebelum dikirimkan ke browser. Hal ini sering digunakan untuk menghindari masalah header yang dikirimkan sebelum konten HTML. Selanjutnya, kondisional `if (isset($_POST['update']))` digunakan untuk memeriksa apakah tombol "Update" pada formulir telah ditekan. Jika ya, maka fungsi `updateBuku()` akan dipanggil untuk memperbarui data buku sesuai dengan nilai yang dimasukkan dalam formulir. Ini memastikan bahwa data buku dapat diperbarui dengan mudah melalui formulir yang sesuai. Kemudian, kondisional `if (isset($_POST['tambah']))` digunakan untuk memeriksa apakah tombol "Tambah" pada formulir telah ditekan. Apabila ya, maka fungsi `insertDataBuku()` akan dipanggil untuk menambahkan data buku baru sesuai dengan nilai yang dimasukkan dalam formulir.

Pada baris [148], syntax `</body>` merupakan tag penutup dari baris 98.

Pada baris [149], syntax `</html>` merupakan tag penutup dari baris 8.

T. Source Code Peminjaman.php

Table 25. Source Code Modul 5 Peminjaman.php

1	<code><?php</code>
2	<code>require "Koneksi.php";</code>
3	<code>require 'Model.php';</code>
4	<code>\$result = getPeminjaman(\$conn);</code>
5	<code>\$dataMember = getMember(\$conn);</code>
6	<code>\$dataBuku = getBuku(\$conn);</code>
7	<code>if (isset(\$_GET['id_peminjaman'])) {</code>
8	<code> deletePeminjaman(\$_GET['id_peminjaman']);</code>
9	<code>}</code>
10	<code>?></code>
11	
12	<code><!DOCTYPE html></code>
13	<code><html lang="en"></code>
14	
15	<code><head></code>
16	<code> <meta charset="UTF-8"></code>
17	<code> <meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>
18	<code> <meta name="viewport" content="width=device-width,</code>
	<code>initial-scale=1.0"></code>
19	<code> <link</code>
	<code>href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dis</code>
	<code>t/css/bootstrap.min.css" rel="stylesheet"</code>
20	<code> integrity="sha384-</code>
	<code>1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy</code>
	<code>l2QvZ6jIW3" crossorigin="anonymous"></code>

21	<title>Peminjaman</title>
22	<style>
23	body {
24	background-image: url('library_page.jpg');
25	background-size: cover;
26	background-position: center;
27	color: #ffffff;
28	font-family: Arial, sans-serif;
29	margin: 0;
30	display: flex;
31	flex-direction: column;
32	align-items: center;
33	justify-content: center;
34	overflow: auto;
35	}
36	
37	.buttons-container {
38	display: flex;
39	justify-content: flex-end;
40	margin-bottom: 20px;
41	}
42	
43	.buttons-container a:first-child {
44	margin-right: 10px;
45	}
46	
47	table,
48	tr,
49	td {
50	border-collapse: collapse;

51	border-radius: 21px;
52	padding: 20px;
53	font-size: 18px;
54	}
55	
56	table {
57	width: max-content;
58	border-radius: 21px;
59	background: rgba(255, 255, 255, 0.8);
60	color: black;
61	}
62	
63	td {
64	width: 100px;
65	text-align: center;
66	}
67	
68	button {
69	display: inline-block;
70	border-radius: 10px;
71	border: 1px solid #03045e;
72	position: relative;
73	overflow: hidden;
74	transition: all 0.5s ease-in;
75	z-index: 1;
76	}
77	
78	button::before,
79	button::after {
80	content: '';

81	position: absolute;
82	top: 0;
83	width: 0;
84	height: 100%;
85	transform: skew(15deg);
86	transition: all 0.5s;
87	overflow: hidden;
88	z-index: -1;
89	}
90	
91	button::before {
92	left: -10px;
93	background: #537188;
94	}
95	
96	button::after {
97	right: -10px;
98	background: #537188;
99	}
100	
101	button:hover::before,
102	button:hover::after {
103	width: 70%;
104	}
105	
106	button:hover span {
107	color: #e0aaff;
108	transition: 0.3s;
109	}
110	

111	button span {
112	color: #03045e;
113	font-size: 18px;
114	transition: all 0.3s ease-in;
115	}
116	
117	.header {
118	background-color: rgba(128, 128, 128, 0.7);
119	border-radius: 10px;
120	padding: 20px;
121	margin-bottom: 20px;
122	width: max-content;
123	}
124	
125	h1 {
126	font-size: 48px;
127	margin: 40px auto 20px;
128	color: #FFFAF0;
129	text-transform: uppercase;
130	letter-spacing: 2px;
131	font-weight: bold;
132	text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
133	animation: fadeInDown 1s ease-in-out,
	scaleIn 1s ease-in-out;
134	border-bottom: 2px solid #FFFAF0;
135	padding-bottom: 10px;
136	background-color: rgba(0, 0, 0, 0.5);
137	width: 80%;
138	text-align: center;
139	}

140	</style>
141	</head>
142	
143	<body class="p-3" style="background-color: #E1D4BB;">
144	<h2>
145	<center><h1>Data Peminjaman</h1></center>
146	</h2>
147	<div class="buttons-container">
148	<button class="btn btn-dark mb-4">Kembali</button>
149	<button class="btn btn-warning mb-1">Tambah Data Peminjaman</button>
150	</div>
151	<table class="table">
152	<thead class="table-dark">
153	<tr>
154	<th class="text-center">ID Peminjaman</th>
155	<th class="text-center">Tanggal Pinjam</th>
156	<th class="text-center">Tanggal Kembali</th>
157	<th class="text-center">Judul Buku</th>
158	<th class="text-center">Nama Member</th>
159	<th class="text-center">Aksi</th>
160	</tr>
161	</thead>
162	<tbody>
163	<?php

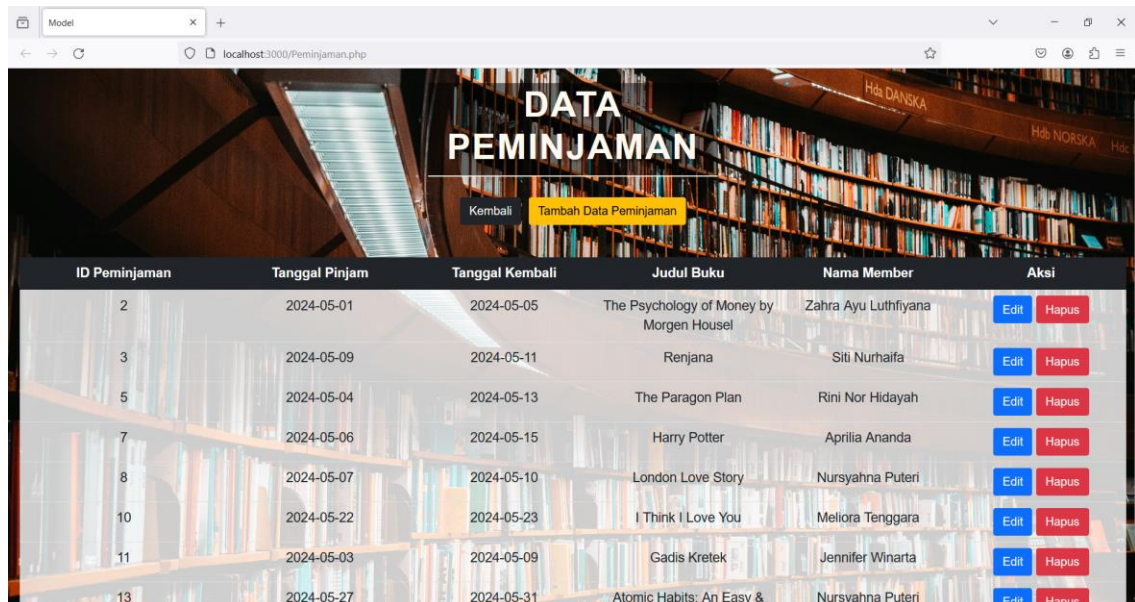
164	foreach (\$result as \$hasil) {
165	echo "<tr>";
166	echo "<td>" . \$hasil["id_peminjaman"] .
	"</td>";
167	echo "<td>" . \$hasil["tgl_pinjam"] .
	"</td>";
168	echo "<td>" . \$hasil["tgl_kembali"] .
	"</td>";
169	\$judulBuku = '';
170	foreach (\$dataBuku as \$temp) {
171	if (\$hasil['id_buku'] ==
	\$temp['id_buku']) {
172	\$judulBuku = \$temp['judul_buku'];
173	break;
174	}
175	}
176	echo "<td>" . \$judulBuku . "</td>";
177	\$namaMember = '';
178	foreach (\$dataMember as \$temp) {
179	if (\$hasil['id_member'] ==
	\$temp['id_member']) {
180	\$namaMember = \$temp['nama_member'];
181	break;
182	}
183	}
184	echo "<td>" . \$namaMember . "</td>";
185	echo "<td>";
186	echo "<a class='btn btn-primary'
	href='FormPeminjaman.php?id_peminjaman="
	\$hasil['id_peminjaman'] . "'>Edit";

```

187         echo " ";
188         echo "<a      class='btn      btn-danger'
href='Peminjaman.php?id_peminjaman="
$hasil['id_peminjaman']      '      onclick=\"return
confirm('Yakin Ingin Dihapus?')\">Hapus</a>";
189         echo "</td>";
190         echo "</tr>";
191     }
192     ?>
193     </tbody>
194     </table>
195 </body>
196 </html>

```

U. Output Program Peminjaman.php



ID Peminjaman	Tanggal Pinjam	Tanggal Kembali	Judul Buku	Nama Member	Aksi
2	2024-05-01	2024-05-05	The Psychology of Money by Morgen Housel	Zahra Ayu Luthfiyana	Edit Hapus
3	2024-05-09	2024-05-11	Renjana	Siti Nurhaifa	Edit Hapus
5	2024-05-04	2024-05-13	The Paragon Plan	Rini Nor Hidayah	Edit Hapus
7	2024-05-06	2024-05-15	Harry Potter	Aprilia Ananda	Edit Hapus
8	2024-05-07	2024-05-10	London Love Story	Nursyahna Puteri	Edit Hapus
10	2024-05-22	2024-05-23	I Think I Love You	Mellora Tenggara	Edit Hapus
11	2024-05-03	2024-05-09	Gadis Kretek	Jennifer Winarta	Edit Hapus
13	2024-05-27	2024-05-31	Atomic Habits: An Easy &	Nursyahna Puteri	Edit Hapus

Gambar 30. Screenshot Hasil Jawaban Modul 5 Peminjaman.php

V. Pembahasan Peminjaman.php

Pada baris [1] [2] [3] [4] [5] [6] [7] [8] [9] [10], syntax

```
<?php

require "Koneksi.php";

require 'Model.php';

$result = getPeminjaman($conn);

$dataMember = getMember($conn);

$dataBuku = getBuku($conn);

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

    deletePeminjaman($_GET['id_peminjaman']);

}

?>
```

penggunaan baris-baris tersebut bertujuan untuk mempersiapkan dan mengelola data yang akan ditampilkan di halaman web. Dengan menggunakan kode tersebut menghubungkan dengan database dengan menggunakan require untuk memasukkan file Koneksi.php, yang kemungkinan berisi informasi koneksi ke database. Selanjutnya, kode tersebut memanggil fungsi getPeminjaman, getMember, dan getBuku dari file Model.php. Fungsi-fungsi ini bertugas mengambil data-data terkait peminjaman, anggota, dan buku dari database. Kemudian, kode tersebut memiliki fungsi untuk penghapusan data peminjaman. Hal ini dilakukan dengan memeriksa apakah terdapat parameter id_peminjaman yang diterima melalui URL (\$_GET['id_peminjaman']). Apabila ada, maka panggilan fungsi deletePeminjaman dari Model.php digunakan untuk menghapus data peminjaman yang sesuai dengan ID yang diterima. Selain itu, kode tersebut juga melakukan penyimpanan data hasil panggilan fungsi getPeminjaman, getMember, dan getBuku ke dalam variabel \$result, \$dataMember, dan \$dataBuku masing-masing. Hal ini bertujuan untuk digunakan

dalam tampilan halaman web untuk menampilkan data peminjaman, anggota, dan buku.

Pada baris [12], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [13], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [15], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [16], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [17], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [18], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta

viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [19] [20], syntax

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/c
ss/bootstrap.min.css" rel="stylesheet"

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2Q
vZ6jIW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS.

Terdapat penggunaan atribut penting dalam tag ini yaitu, href=<https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css>

digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag <link> adalah rel, yang dengan nilai rel="stylesheet" menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut integrity seperti dalam integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka akan dilakukan penghitungan hash dari file yang diunduh dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut crossorigin dengan nilai crossorigin="anonymous" digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan.

Penggunaan, anonymous berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

Pada baris [21], syntax `<title>Peminjaman</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [23] – [140], syntax

```
<style>

    body {

        background-image: url('library_page.jpg');

        background-size: cover;

        background-position: center;

        color: #ffffff;

        font-family: Arial, sans-serif;

        margin: 0;

        display: flex;

        flex-direction: column;

        align-items: center;

        justify-content: center;

        overflow: auto;

    }
```



```

.buttons-container {
    display: flex;
    justify-content: flex-end;
    margin-bottom: 20px;
}

.buttons-container a:first-child {
    margin-right: 10px;
}

```

```

table,
tr,
td {
    border-collapse: collapse;
    border-radius: 21px;
    padding: 20px;
    font-size: 18px;
}

```

```

table {
    width: max-content;
    border-radius: 21px;
}

```

```

        background: rgba(255, 255, 255, 0.8);
        color: black;
    }

    td {
        width: 100px;
        text-align: center;
    }

    button {
        display: inline-block;
        border-radius: 10px;
        border: 1px solid #03045e;
        position: relative;
        overflow: hidden;
        transition: all 0.5s ease-in;
        z-index: 1;
    }

    button::before,
    button::after {
        content: '';

```

```
    position: absolute;

    top: 0;

    width: 0;

    height: 100%;

    transform: skew(15deg);

    transition: all 0.5s;

    overflow: hidden;

    z-index: -1;

}
```

```
button::before {

    left: -10px;

    background: #537188;

}
```

```
button::after {

    right: -10px;

    background: #537188;

}
```

```
button:hover::before,

button:hover::after {
```

```

        width: 70%;
    }

    button:hover span {
        color: #e0aaff;
        transition: 0.3s;
    }

    button span {
        color: #03045e;
        font-size: 18px;
        transition: all 0.3s ease-in;
    }

    .header {
        background-color: rgba(128, 128, 128, 0.7);
        border-radius: 10px;
        padding: 20px;
        margin-bottom: 20px;
        width: max-content;
    }

```

```

h1 {
    font-size: 48px;
    margin: 40px auto 20px;
    color: #FFFAF0;
    text-transform: uppercase;
    letter-spacing: 2px;
    font-weight: bold;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
    animation: fadeInDown 1s ease-in-out, scaleIn
1s ease-in-out;

    border-bottom: 2px solid #FFFAF0;
    padding-bottom: 10px;
    background-color: rgba(0, 0, 0, 0.5);
    width: 80%;
    text-align: center;
}

</style>

```

penggunaan baris-baris tersebut adalah kumpulan aturan gaya (CSS) yang ditujukan untuk mengatur tampilan dan tata letak elemen-elemen pada halaman web. Bagian "Body Styling" mengatur gambar latar belakang dengan background-image, serta menetapkan tata letak dan penempatan elemen-elemen menggunakan properti seperti display: flex, flex-direction: column, align-items: center, dan justify-content: center. Selain itu, bagian ini juga mengatur warna teks, jenis font, dan margin untuk meningkatkan keterbacaan dan tampilan keseluruhan halaman. Bagian "Buttons

Container" fokus pada penataan tombol dan container tombol, dengan tombol-tombol ditempatkan di sebelah kanan bawah halaman untuk memudahkan akses pengguna. Bagian "Table Styling" bertujuan untuk meningkatkan tampilan tabel, baris, dan sel dengan mengatur padding, ukuran teks, border-collapse, serta memberikan efek hover pada seluruh sel tabel. Di sisi lain, aturan gaya dalam "Button Styling" menambahkan efek transisi dan bayangan pada tombol, serta mengatur warna, ukuran, dan border-radius tombol, sambil memberikan efek hover untuk mengubah warna teks tombol saat disorot oleh kursor. Bagian "Header Styling" memusatkan perhatian pada penampilan header, termasuk warna latar belakang, padding, dan margin, dengan tambahan efek bayangan dan border bawah untuk meningkatkan kerapian dan visualisasi. Terakhir, aturan gaya dalam "Judul H1 Styling" menetapkan tampilan judul H1, termasuk ukuran teks, warna, penempatan, dan efek bayangan, yang membantu menonjolkan judul halaman.

Pada baris [141], syntax `</head>` merupakan tag penutup dari baris 15.

Pada baris [143] [144] [145] [146] [147] [148] [149] [150] [151] [152] [153] [154] [155] [156] [157] [158] [159] [160] [161], syntax

```
<body class="p-3" style="background-color: #E1D4BB;">

    <h2>

        <center><h1>Data Peminjaman</h1></center>

    </h2>

    <div class="buttons-container">

        <a href="index.php"><button class="btn btn-dark mb-4">Kembali</button></a>

        <a href="FormPeminjaman.php"><button class="btn btn-warning mb-1">Tambah Data Peminjaman</button></a>

    </div>
```

```

<table class="table">

    <thead class="table-dark">

        <tr>

            <th class="text-center">ID
Peminjaman</th>

            <th class="text-center">Tanggal
Pinjam</th>

            <th class="text-center">Tanggal
Kembali</th>

            <th class="text-center">Judul Buku</th>

            <th class="text-center">Nama Member</th>

            <th class="text-center">Aksi</th>

        </tr>

    </thead>

```

penggunaan kode-kode tersebut adalah bagian penting dari struktur HTML yang bertujuan untuk mengelola tampilan halaman web yang diperlukan. Setiap elemen dalam kode tersebut memiliki fungsi khusus yang mendukung tujuan keseluruhan halaman. Penggunaan, body class dengan atribut "p-3" memberikan padding sebesar 3 unit pada semua sisi elemen body, memastikan konten terletak dengan nyaman di dalamnya. Judul H1 "<h1>Data Peminjaman</h1>" menonjolkan judul halaman dengan ukuran yang besar, memberikan pengguna informasi yang jelas tentang konten yang ditampilkan. Bagian tombol navigasi, yang terkandung dalam div container "<div class='buttons-container'>", memberikan pengguna akses mudah ke halaman lain. Tombol "Kembali" dengan link "<button class='btn btn-dark mb-4'>Kembali</button>" memungkinkan pengguna untuk kembali ke halaman utama, sementara tombol "Tambah Data Peminjaman" "<a

`href="FormPeminjaman.php"><button class="btn btn-warning mb-1">Tambah Data Peminjaman</button>` mengarahkan pengguna ke halaman formulir untuk menambah data peminjaman baru. Selanjutnya, tabel data peminjaman "`<table class="table">`" dirancang untuk menampilkan informasi peminjaman dengan menggunakan kelas dari Bootstrap untuk penataan yang lebih baik. Bagian "`<thead class="table-dark">`" mengatur tampilan header tabel dengan latar belakang gelap, menciptakan kontras yang jelas dengan konten tabel. Setiap baris dalam tabel, ditandai dengan "`<tr>`", menampilkan judul kolom-kolom dengan menggunakan "`<th class="text-center">`" untuk memposisikan teksnya di tengah.

Pada baris [162] [163] [164] [165] [166] [167] [168] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183] [184] [185] [186] [187] [188] [189] [190] [191] [192] [193], syntax

```
<tbody>

    <?php
        foreach ($result as $hasil) {

            echo "<tr>";

            echo "    <td>" . $hasil["id_peminjaman"] .
"</td>";

            echo "<td>" . $hasil["tgl_pinjam"] . "</td>";
            echo "<td>" . $hasil["tgl_kembali"] . "</td>";

            $judulBuku = '';

            foreach ($dataBuku as $temp) {

                if ($hasil['id_buku'] == $temp['id_buku'])

                    $judulBuku = $temp['judul_buku'];
```



```

        break;
    }
}

echo "<td>" . $judulBuku . "</td>";

$namaMember = '';

foreach ($dataMember as $temp) {
    if ($hasil['id_member'] ==
$temp['id_member']) {
        $namaMember = $temp['nama_member'];
        break;
    }
}

echo "<td>" . $namaMember . "</td>";

echo "<td>";

echo "<a      class='btn      btn-primary'
href='FormPeminjaman.php?id_peminjaman="
$hasil['id_peminjaman'] . "'>Edit</a>";

echo " ";

echo "<a      class='btn      btn-danger'
href='Peminjaman.php?id_peminjaman="
$hasil['id_peminjaman'] . "'      onclick=\"return
confirm('Yakin Ingin Dihapus?')\">Hapus</a>";

echo "</td>";

echo "</tr>";

```

```

    }

    ?>

</tbody>

```

Penggunaan baris-baris tersebut adalah proses pembuatan baris-baris dalam table data peminjaman pada halaman web. Penggunaan <tbody> digunakan untuk menandai awal dari bagian tubuh tabel. Di sini, sebuah loop foreach dimulai dengan <?php foreach (\$result as \$hasil) { ?>, yang bertujuan untuk mengulangi setiap item dalam array \$result. Setiap iterasi dari loop tersebut akan menampilkan sebuah baris tabel yang dimulai dengan <tr>. Dalam setiap baris tabel, data peminjaman ditampilkan dalam beberapa kolom. Misalnya, echo "<td>" . \$hasil["id_peminjaman"] . "</td>"; menampilkan nilai dari kunci "id_peminjaman" dari setiap item dalam array \$result sebagai sebuah sel data dalam kolom tabel. Sama halnya dengan kolom "tgl_pinjam" dan "tgl_kembali", yang menampilkan tanggal peminjaman dan tanggal pengembalian. Namun, informasi tambahan seperti judul buku dan nama member juga disertakan dalam tabel. Untuk mendapatkan judul buku dan nama member, dilakukan loop foreach terpisah untuk setiap data buku dan data member. Misalnya, dengan mengakses data buku melalui \$dataBuku, dan jika id buku dalam data peminjaman sama dengan id buku dalam data buku, judul buku akan ditampilkan dalam kolom yang sesuai. Sama seperti untuk mendapatkan nama member, dilakukan loop foreach terhadap \$dataMember dan memeriksa kesesuaian id member. Setelah semua data ditampilkan dalam kolom yang sesuai, tombol aksi "Edit" dan "Hapus" juga disertakan di setiap baris. Tombol "Edit" akan mengarahkan pengguna ke halaman FormPeminjaman.php untuk mengedit data peminjaman, sedangkan tombol "Hapus" akan mengarahkan ke halaman Peminjaman.php untuk menghapus data peminjaman yang terkait. Sebuah konfirmasi akan muncul sebelum penghapusan untuk memastikan tindakan tersebut diinginkan. Setelah loop foreach selesai, elemen </tbody> menandakan akhir dari bagian tubuh tabel

Pada baris [194], syntax </table> merupakan tag penutup dari baris 151.

Pada baris [195], syntax `</body>` merupakan tag penutup dari baris 143.

Pada baris [196], syntax `</html>` merupakan tap penutup dari baris 13.

W. Source Code FormPeminjaman.php

Table 26. Source Code Modul 5 FormPeminjaman.php

1	<code><?php</code>
2	<code>require "Koneksi.php";</code>
3	<code>require 'Model.php';</code>
4	
5	<code>if (isset(\$_GET['id_peminjaman'])) {</code>
6	<code> editPeminjaman();</code>
7	<code>}</code>
8	
9	<code>\$dataMember = getMember(\$conn);</code>
10	<code>\$dataBuku = getBuku(\$conn);</code>
11	
12	<code>if (isset(\$_POST['update'])) {</code>
13	<code> updatePeminjaman(\$_GET["id_peminjaman"],</code>
	<code> \$_POST["tgl_pinjam"], \$_POST["tgl_kembali"],</code>
	<code> \$_POST["id_buku"], \$_POST["id_member"]);</code>
14	<code>}</code>
15	<code>if (isset(\$_POST['tambah'])) {</code>
16	<code> insertDataPeminjaman(\$_POST["tgl_pinjam"],</code>
	<code> \$_POST["tgl_kembali"], \$_POST["id_buku"],</code>
	<code> \$_POST["id_member"]);</code>
17	<code>}</code>
18	<code>?></code>
19	

20	<!DOCTYPE html>
21	<html lang="en">
22	
23	<head>
24	<meta charset="UTF-8">
25	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
26	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
27	<link
	href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dis
	t/css/bootstrap.min.css" rel="stylesheet"
28	integrity="sha384-
	1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy
	l2QvZ6jIW3" crossorigin="anonymous">
29	<title>Form Peminjaman</title>
30	<style>
31	body {
32	background-image: url('library_page.jpg');
33	background-size: cover;
34	background-position: center;
35	color: #ffffff;
36	font-family: Arial, sans-serif;
37	margin: 0;
38	display: flex;
39	justify-content: center;
40	align-items: center;
41	height: 100vh;
42	overflow: auto;
43	}

44	input {
45	font-weight: 500;
46	font-size: .8vw;
47	color: #fff;
48	background-color: rgb(28, 28, 30);
49	box-shadow: 0 0 .4vw rgba(0, 0, 0, 0.5), 0
	0 0 .15vw transparent;
50	border-radius: 0.4vw;
51	border: none;
52	outline: none;
53	padding: 0.4vw;
54	transition: .4s;
55	}
56	.form-container {
57	background-color: rgba(225, 212, 187, 0.5);
58	padding: 20px;
59	border-radius: 10px;
60	}
61	input:hover {
62	box-shadow: 0 0 0 .15vw rgba(135, 207, 235,
	0.186);
63	}
64	
65	input:focus {
66	box-shadow: 0 0 0 .15vw skyblue;
67	}
68	
69	button {
70	display: inline-block;
71	border-radius: 10px;

72	border: 1px solid #03045e;
73	position: relative;
74	overflow: hidden;
75	transition: all 0.5s ease-in;
76	z-index: 1;
77	}
78	
79	button::before,
80	button::after {
81	content: '';
82	position: absolute;
83	top: 0;
84	width: 0;
85	height: 100%;
86	transform: skew(15deg);
87	transition: all 0.5s;
88	overflow: hidden;
89	z-index: -1;
90	}
91	
92	button::before {
93	left: -10px;
94	background: #537188;
95	}
96	
97	button::after {
98	right: -10px;
99	background: #537188;
100	}
101	

102	button:hover::before,
103	button:hover::after {
104	width: 70%;
105	}
106	
107	button:hover span {
108	color: #e0aaff;
109	transition: 0.3s;
110	}
111	
112	button span {
113	color: #03045e;
114	font-size: 18px;
115	transition: all 0.3s ease-in;
116	}
117	</style>
118	</head>
119	
120	<div class="container form-container">
121	<div class="container">
122	<div class="row">
123	<form action="" method="post">
124	<h1 style="text-align: center;"
	class="mt-2">Tambah Data Peminjaman</h1>
125	<label for="tgl_pinjam" class="form-
	label">Tanggal Peminjaman</label>
126	<input type="date" name="tgl_pinjam"
	id="tgl_pinjam" class="form-control mb-3" value="<?php
	if (isset(\$_GET['id_peminjaman']))
127	

	echo \$result[0]["tgl_pinjam"] ?>"
128	required>
129	
	<label for="tgl_kembali"
	class="form-label">Tanggal Kembali</label>
130	<input type="date"
	name="tgl_kembali" id="tgl_kembali" class="form-control
	mb-3" value="<?php if (isset(\$_GET['id_peminjaman']))
131	echo \$result[0]["tgl_kembali"] ?>"
	required>
132	
133	<label for="id_buku">Judul
	Buku:</label>
134	<select id="id_buku"
	name="id_buku">
135	<?php
136	if (!isset(\$_GET['id_peminjaman'])) {
	?>
137	<option disabled
	selected></option>
138	<?php
139	}
140	foreach (\$dataBuku as \$barisBuku) {
141	\$selected = '';
142	if (
143	isset(\$_GET['id_peminjaman'])
	&& \$hasil[0]['id_buku'] ==
144	
145	\$barisBuku['id_buku']
146) {

147	
148	\$selected = 'selected';
149	}
150	?>
	<option value="<?php echo
	\$barisBuku['id_buku']; ?>" <?php echo \$selected;
151	?>><?php echo \$barisBuku['judul_buku']; ?></option>
	<?php } ?>
	</select>
152	
153	
154	<label for="id_member">Nama
155	Member:</label>
156	<select id="id_member"
	name="id_member">
157	<?php
	if (!isset(\$_GET['id_peminjaman']))
158	{ ?>
159	<option disabled
	selected></option>
160	<?php
	}
161	foreach (\$dataMember as
162	\$barisMember) {
163	\$selected = '';
	if (
164	
165	isset(\$_GET['id_peminjaman']) && \$hasil[0]['id_member']
166	==
167	

	\$barisMember['id_member']
) {
168	
169	\$selected = 'selected';
170	}
171	?>
172	<option value="<?php echo
173	\$barisMember['id_member']; ?>" <?php echo \$selected;
174	?><?php echo \$barisMember['nama_member']; ?></option>
	<?php } ?>
	</select>
175	
176	
177	<?php
178	if (isset(\$_GET['id_peminjaman'])) {
179	echo "<button type=\"submit\"
180	class=\"btn btn-success mt-3\" name=\"update\"> Update
181	</button>";
	} else {
	echo "<button type=\"submit\"
182	class=\"btn btn-success mt-3\" name=\"tambah\"> Tambah
183	</button>";
	}
	?>
184	</div>
185	</div>
186	</form>
187	</body>
188	</html>
189	

X. Output Program FormPeminjaman.php

Gambar 31. Screenshot Hasil Jawaban Modul 5 FormPeminjaman.php

Y. Pembahasan FormPeminjaman.php

Pada baris [1] [2] [3], syntax

```
<?php
```

```
require "Koneksi.php";
```

```
require 'Model.php';
```

penggunaannya baris kode tersebut bertujuan untuk mengimpor (require) dua file PHP yang penting untuk menjalankan skrip saat ini. Penggunaan, file "Koneksi.php" berisi konfigurasi koneksi ke database, termasuk informasi seperti host, nama pengguna, kata sandi, dan nama database. Dengan mengimpor file ini, skrip dapat terhubung ke

database untuk melakukan berbagai operasi seperti pengambilan, pembaruan, atau penghapusan data. Kemudian, file 'Model.php' kemungkinan berisi definisi dari fungsi-fungsi yang terkait dengan model data dalam aplikasi. Hal ini dapat mencakup fungsi-fungsi yang bertanggung jawab atas pengambilan, pembaruan, atau penghapusan data dari database, serta fungsi-fungsi lain yang terkait dengan logika bisnis aplikasi. Dengan mengimpor kedua file ini, skrip PHP memperoleh akses ke fungsionalitas yang disediakan di dalamnya. Sehingga memungkinkan skrip untuk menjalankan berbagai operasi yang diperlukan, seperti mengambil data dari database atau memanipulasi model data sesuai dengan kebutuhan aplikasi yang sedang berjalan.

Pada baris [5] [6] [7], syntax

```
if (isset($_GET['id_peminjaman'])) {  
    editPeminjaman();  
}
```

Penggunaan kode tersebut bertujuan untuk memeriksa keberadaan parameter `id_peminjaman` dalam URL menggunakan fungsi `isset($_GET['id_peminjaman'])`. Fungsi ini digunakan dalam konteks aplikasi web untuk menentukan apakah pengguna sedang dalam mode pengeditan atau penambahan data. Apabila parameter `id_peminjaman` ada dalam URL, maka dapat disimpulkan bahwa pengguna sedang dalam mode pengeditan data peminjaman dengan ID tertentu. Dalam hal ini, fungsi `editPeminjaman()` akan dipanggil untuk menangani proses pengeditan data peminjaman yang spesifik tersebut. Pemeriksaan keberadaan parameter dalam URL adalah langkah penting dalam pengembangan aplikasi web yang memungkinkan respons yang sesuai dengan tindakan pengguna. Dengan memeriksa keberadaan parameter seperti `id_peminjaman`, aplikasi dapat menyesuaikan perilakunya, seperti menampilkan formulir pengeditan data, menyimpan perubahan yang dibuat, atau mengarahkan pengguna ke halaman yang sesuai. Sehingga memungkinkan aplikasi untuk memberikan pengalaman pengguna yang lebih dinamis dan responsif, sesuai dengan kebutuhan dan interaksi yang terjadi.

Pada baris [9] [10], syntax

```
$dataMember = getMember($conn);  
  
$dataBuku = getBuku($conn);
```

penggunaan kode tersebut bertujuan untuk mendapatkan data member dan data buku dari database menggunakan fungsi `getMember()` dan `getBuku()` yang telah didefinisikan di dalam file `Model.php`. Variabel `$dataMember` akan berisi data dari tabel member, sedangkan `$dataBuku` akan berisi data dari tabel buku. Kedua data ini kemungkinan akan digunakan dalam proses selanjutnya, seperti menampilkan informasi ke layar atau melakukan operasi CRUD (Create, Read, Update, Delete) terhadap data tersebut.

Pada abris [12] [13] [14] [15] [16] [17] [18] [19], syntax

```
if (isset($_POST['update'])) {  
  
    updatePeminjaman($_GET["id_peminjaman"],  
$_POST["tgl_pinjam"],          $_POST["tgl_kembali"],  
$_POST["id_buku"], $_POST["id_member"]);  
  
}  
  
if (isset($_POST['tambah'])) {  
  
    insertDataPeminjaman($_POST["tgl_pinjam"],  
$_POST["tgl_kembali"],          $_POST["id_buku"],  
$_POST["id_member"]);  
  
}  
  
?>
```

penggunaan kode tersebut bertanggung jawab atas pemrosesan data yang diterima melalui metode POST. Ketika variabel POST dengan nama 'update' terdeteksi, langkah selanjutnya adalah memanggil fungsi `updatePeminjaman()`. Fungsi ini bertugas

memperbarui data peminjaman di dalam database, dengan memperhitungkan nilai-nilai yang diterima dari formulir, seperti tanggal pinjam, tanggal kembali, id buku, dan id member. Selanjutnya, jika terdapat variabel POST dengan nama 'tambah', fungsi `insertDataPeminjaman()` akan dijalankan. Tugasnya adalah menambahkan data peminjaman baru ke dalam basis data, berdasarkan informasi yang diperoleh dari formulir. Data tambahan ini termasuk tanggal pinjam, tanggal kembali, id buku, dan id member. Baik fungsi `updatePeminjaman()` maupun `insertDataPeminjaman()` memiliki peran krusial dalam manajemen data peminjaman aplikasi, memastikan bahwa informasi yang dimasukkan oleh pengguna diproses dengan benar dan disimpan secara akurat dalam sistem.

Pada baris [20], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [21], syntax `<html lang="en">` tag `html` merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [23], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [24], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk

menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [25], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [26], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsicve. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [27] [28], syntax

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/c
ss/bootstrap.min.css" rel="stylesheet"

integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2Q
vZ6jIW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS. Terdapat penggunaan atribut penting dalam tag ini yaitu, `href=https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css` digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag `<link>` adalah `rel`, yang dengan nilai `rel="stylesheet"` menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut `integrity` seperti dalam `integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW`

3" berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka akan dilakukan penghitungan hash dari file yang diunduh dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut `crossorigin` dengan nilai `crossorigin="anonymous"` digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan. Penggunaan, `anonymous` berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

Pada baris [29], syntax `<title>Form Peminajamn</title>` tag `Title` merupakan salah satu contoh informasi yang terdapat di dalam tag `Head`. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [30] – [117], syntax

```
<style>

    body {

        background-image: url('library_page.jpg');

        background-size: cover;

        background-position: center;

        color: #ffffff;

        font-family: Arial, sans-serif;

        margin: 0;

        display: flex;
```



```

        justify-content: center;

        align-items: center;

        height: 100vh;

        overflow: auto;
    }

    input {

        font-weight: 500;

        font-size: .8vw;

        color: #fff;

        background-color: rgb(28, 28, 30);

        box-shadow: 0 0 .4vw rgba(0, 0, 0, 0.5), 0 0 0
.15vw transparent;

        border-radius: 0.4vw;

        border: none;

        outline: none;

        padding: 0.4vw;

        transition: .4s;
    }

    .form-container {

        background-color: rgba(225, 212, 187, 0.5);

        padding: 20px;

        border-radius: 10px;

```

```
}  
  
input:hover {  
    box-shadow: 0 0 0 .15vw rgba(135, 207, 235,  
0.186);  
}
```

```
input:focus {  
    box-shadow: 0 0 0 .15vw skyblue;  
}
```

```
button {  
    display: inline-block;  
    border-radius: 10px;  
    border: 1px solid #03045e;  
    position: relative;  
    overflow: hidden;  
    transition: all 0.5s ease-in;  
    z-index: 1;  
}
```

```
button::before,  
button::after {
```

```
    content: '';  
    position: absolute;  
    top: 0;  
    width: 0;  
    height: 100%;  
    transform: skew(15deg);  
    transition: all 0.5s;  
    overflow: hidden;  
    z-index: -1;  
}
```

```
button::before {  
    left: -10px;  
    background: #537188;  
}
```

```
button::after {  
    right: -10px;  
    background: #537188;  
}
```

```
button:hover::before,
```

```

        button:hover::after {

            width: 70%;

        }

        button:hover span {

            color: #e0aaff;

            transition: 0.3s;

        }

        button span {

            color: #03045e;

            font-size: 18px;

            transition: all 0.3s ease-in;

        }

    </style>

```

penggunaan kode-kode tersebut merupakan kode CSS, penggunaan properti yang diterapkan pada elemen body mengontrol aspek-aspek penting seperti gambar latar belakang, warna teks, jenis font, dan tata letak keseluruhan halaman. Kemudian, untuk elemen input, berbagai properti ditetapkan untuk mengatur tampilan input, termasuk font, warna teks, latar belakang, bayangan, dan transisi saat pengguna mengarahkan kursor ke input tersebut. Selanjutnya, pengaturan pada kelas .form-container bertujuan untuk merancang tampilan kontainer formulir. Properti seperti warna latar belakang, padding, dan border-radius diterapkan untuk memberikan tampilan yang konsisten dan menarik pada area formulir. Pada baris tersebut, gaya tombol juga diatur secara detail

menggunakan properti CSS. Elemen button mendapat penyesuaian seperti border-radius, border, bayangan, dan transisi saat tombol disorot dengan kursor. Efek tambahan sebelum dan sesudah tombol diimplementasikan melalui pseudo-elements button::before dan button::after dengan menggunakan efek skewing (memiringkan), yang memberikan kesan visual yang menarik saat tombol dihover. Lalu, untuk menyesuaikan tampilan teks di dalam tombol, elemen button span juga mendapat perhatian dengan pengaturan warna teks dan transisi yang sesuai. Dengan semua pengaturan ini, kombinasi CSS tersebut memastikan bahwa halaman web akan memiliki tampilan yang menarik dan responsif ketika digunakan untuk mengisi formulir.

Pada baris [118], syntax `</head>` merupakan tag penutup dari bari 23.

Pada baris [120] – [188], syntax

```
<div class="container form-container">

    <div class="container">

        <div class="row">

            <form action="" method="post">

                <h1 style="text-align: center;" class="mt-2">Tambah Data Peminjaman</h1>

                <label    for="tgl_pinjam"    class="form-label">Tanggal Peminjaman</label>

                <input    type="date"    name="tgl_pinjam"
id="tgl_pinjam" class="form-control mb-3" value="<?php if
(isset($_GET['id_peminjaman']))

                    echo    $result[0]["tgl_pinjam"]    ?>"
required>
```

```

        <label for="tgl_kembali" class="form-label">Tanggal Kembali</label>

```

```

        <input type="date" name="tgl_kembali"
id="tgl_kembali" class="form-control mb-3" value="<?php if
(isset($_GET['id_peminjaman']))

```

```

        echo $result[0]["tgl_kembali"] ?>"
required>

```

```

        <label for="id_buku">Judul
Buku:</label><br>

```

```

        <select id="id_buku" name="id_buku">

```

```

        <?php

```

```

        if (!isset($_GET['id_peminjaman'])) { ?>

```

```

        <option disabled
selected></option>

```

```

        <?php

```

```

        }

```

```

        foreach ($dataBuku as $barisBuku) {

```

```

            $selected = '';

```

```

            if (

```

```

                isset($_GET['id_peminjaman']) &&

```

```

                $hasil[0]['id_buku'] ==

```

```

                $barisBuku['id_buku']

```

```

    ) {

        $selected = 'selected';

    }

    ?>

        <option        value="<?php        echo
$barisBuku['id_buku']; ?>" <?php echo $selected; ?>><?php
echo $barisBuku['judul_buku']; ?></option>

        <?php } ?>

    </select>

    <br><br>

    <label                for="id_member">Nama
Member:</label><br>

    <select id="id_member" name="id_member">

        <?php

            if (!isset($_GET['id_peminjaman'])) {

?>

                <option                disabled
selected></option>

                <?php

                }

                foreach ($dataMember as $barisMember)

{

```

```

        $selected = '';

        if (

            isset($_GET['id_peminjaman'])

&& $hasil[0]['id_member'] ==

            $barisMember['id_member']

        ) {

            $selected = 'selected';

        }

        ?>

        <option        value="<?php        echo
$barisMember['id_member']; ?>"        <?php        echo        $selected;
?>><?php        echo        $barisMember['nama_member']; ?></option>

        <?php        }        ?>

</select>

<br><br>

<?php

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

            echo        "<button        type=\"submit\"
class=\"btn        btn-success        mt-3\"        name=\"update\">        Update
</button>";

```



```

        } else {

            echo      "<button      type=\"submit\"
class=\"btn btn-success mt-3\"  name=\"tambah\">  Tambah
</button>";

        }

        ?>

    </div>

</div>

</form>

```

penggunaan baris-baris tersebut merupakan bagian penting dari formulir yang bertujuan untuk menambah data peminjaman pada halaman web. Setiap baris kode memiliki fungsi spesifik dalam mengatur tampilan dan perilaku formulir tersebut. Pertama, elemen `<div class="container form-container">` digunakan untuk membuat sebuah kontainer yang mengatur tata letak dan tampilan formulir secara keseluruhan. Selanjutnya, `<div class="container">` digunakan sebagai kontainer tambahan untuk mengatur tampilan elemen-elemen dalam formulir, sedangkan `<div class="row">` adalah elemen untuk menyusun elemen-elemen secara horizontal dalam grid sistem Bootstrap. Formulir itu sendiri didefinisikan menggunakan `<form>` dengan metode POST untuk mengirim data ke halaman yang sama. Judul formulir, "Tambah Data Peminjaman", ditampilkan di tengah halaman dengan `<h1>` dan diberi gaya untuk perataan teks. Setiap input tanggal untuk "Tanggal Peminjaman" dan "Tanggal Kembali" disertai dengan label yang jelas menggunakan `<label>`, serta atribut-atribut yang mengatur tampilan dan validasi input. Selain itu, terdapat dropdown untuk memilih judul buku dan nama member dengan menggunakan elemen `<select>`, yang diisi dengan opsi-opsi yang diperoleh dari array `$dataBuku` dan `$dataMember` melalui PHP. Lalu, melalui PHP, ditentukan jenis tombol yang akan ditampilkan berdasarkan keberadaan parameter GET `"id_peminjaman"`. Jika parameter tersebut ada, tombol

yang ditampilkan adalah tombol "Update" untuk mengedit data peminjaman yang sudah ada. Jika tidak, tombol yang ditampilkan adalah tombol "Tambah" untuk menambahkan data peminjaman baru.

Pada baris [189], syntax `</body>` merupakan tag penutup .

pada baris [190], syntax `</html>` merupakan tag penutup.

MODUL 6:

CRUD DAN LOGIN PADA CODEIGNITER

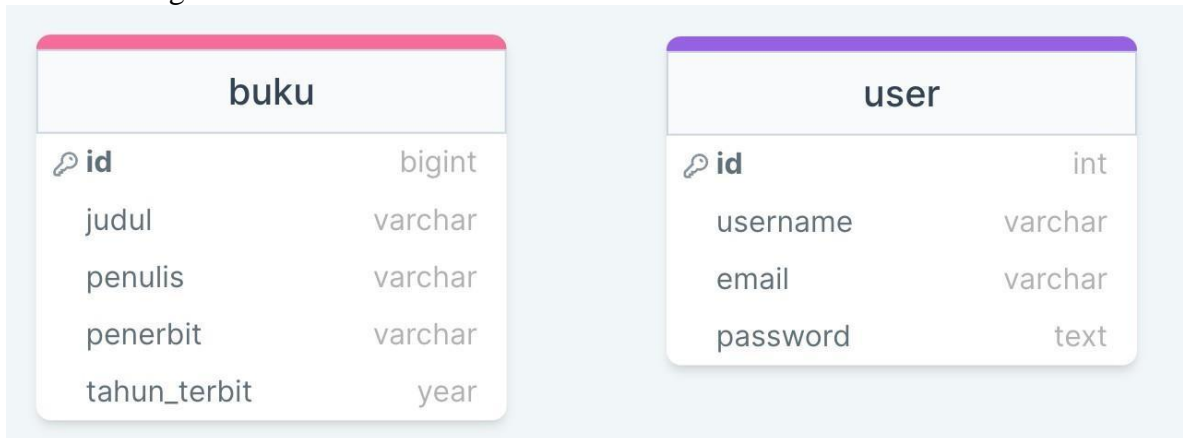
SOAL PRAKTIKUM

Buatlah sebuah website yang memiliki fitur login dan logout. Setelah login pengguna website dapat melakukan CRUD yang terhubung langsung dengan database.

Beberapa syarat yang harus ada pada website:

- Jika seorang pengguna belum login dan mengakses halaman yang mengharuskan login, maka tampilkan sebuah pesan peringatan di halaman login, peringatan berisi “Login terlebih dahulu!”.
- Berikan validasi pada bagian ‘Create’ atau ‘tambah data’ di form, sebagai berikut:
 - Judul : Harus diisi dan berupa string
 - penulis : Harus diisi dan berupa string
 - penerbit : Harus diisi dan berupa string
 - tahun_terbit : Harus diisi dan berupa angka, angka harus lebih besardari 1800 dan lebih kecil dari 2024
- (opsional) buatlah custom pesan validasi menggunakan bahasa indonesia

- Buat file migration pada codeigniter dengan Desain konseptual database sebagai berikut:



Simpan dengan nama folder github Github: PRAK601

A. Source Code *Routes.php*

Table 27. Source Code Modul 6 *Routes.php*

1	<?php
2	
3	namespace Config;
4	
5	\$routes = Services::routes();
6	
7	\$routes->setDefaultNamespace('App\Controllers');
8	\$routes->setDefaultController('Home');
9	\$routes->setDefaultMethod('index');
10	\$routes->setTranslateURIDashes(false);
11	\$routes->set404Override();
12	
13	\$routes->get('/', 'Home::index', ['filter' =>
14	'auth']);
15	\$routes->get('/buku', 'BukuController::index');

16	\$routes->get('/buku/create',
17	'BukuController::create');
	\$routes->post('/buku/create',
18	'BukuController::store');
	\$routes->get('/buku/(:num)/edit',
19	'BukuController::edit/\$1');
	\$routes->post('/buku/(:num)/edit',
20	'BukuController::update/\$1');
21	\$routes->delete('/buku/(:num)/delete',
22	'BukuController::delete/\$1');
23	
24	\$routes->get('/login', 'UserController::index');
25	
26	\$routes->get('/register',
27	'RegisterController::index');
28	\$routes->post('/register',
29	'RegisterController::store');
30	\$routes->post('/login', 'UserController::login');
	\$routes->get('/logout', 'UserController::logout');
31	
	if (is_file(APPPATH . 'Config/' . ENVIRONMENT .
	'/Routes.php')) {
	require APPPATH . 'Config/' . ENVIRONMENT .
	'/Routes.php';
	}

B. Pembahasan *Routes.php*

Pada baris [1] [3] [5], syntax

```
<?php  
  
namespace Config;  
  
$routes = Services::routes();
```

pada baris-baris tersebut digunakan untuk mendefinisikan namespace Config dan menginisialisasi variabel \$routes dengan layanan routes() dari CodeIgniter. Tujuan dari penggunaan objek '\$routes' tersebut adalah untuk mengatur atau routing permintaan. Pada penggunaan kode ini merupakan awal dalam memberikan pengaturan rute yang terorganisir dengan baik untuk aplikasi web berbasis CodeIgniter.

Pada baris [7] [8] [9], syntax

```
$routes->setDefaultNamespace('App\Controllers');  
  
$routes->setDefaultController('Home');  
  
$routes->setDefaultMethod('index');
```

Pada baris-baris tersebut digunakan untuk mengatur nilai default untuk namespace kontroler (App\Controllers), kontroler default (Home), dan metode default (index). Disaat tidak ada rute yang cocok, aplikasi web akan mengarahkan ke Home controller dan metode index.

Pada baris [10] [11], syntax

```
$routes->setTranslateURIDashes(false);  
  
$routes->set404Override();
```

pada baris-baris tersebut terdapat beberapa fungsi dari penggunaan kode-kode yang ada, penggunaan \$routes->setTranslateURIDashes(false);: digunakan untuk mengatur agar karakter dash (-) dalam URI tidak diartikan sebagai pemisah segmen. Lalu

penggunaan `$routes->set404Override();`: untuk mengaktifkan penggantian halaman 404 default CodeIgniter dengan penggunaan kustom.

Pada baris [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27],
syntax

```
$routes->get('/', 'Home::index', ['filter' => 'auth']);

$routes->get('/buku', 'BukuController::index');

$routes->get('/buku/create', 'BukuController::create');

$routes->post('/buku/create', 'BukuController::store');

$routes->get('/buku/(:num)/edit',
'BukuController::edit/$1');

$routes->post('/buku/(:num)/edit',
'BukuController::update/$1');

$routes->delete('/buku/(:num)/delete',
'BukuController::delete/$1');

$routes->get('/login', 'UserController::index');

$routes->get('/register', 'RegisterController::index');

$routes->post('/register', 'RegisterController::store');

$routes->post('/login', 'UserController::login');

$routes->get('/logout', 'UserController::logout');
```

pada baris-baris tersebut mendefinisikan beberapa rute yang tersedia dalam aplikasi web yang dibuat. Pada aplikais tersbeut terdapat rute untuk halaman Home yang diberlengkapi dengan filter autentikasi, memastikan hanya pengguna yang terautentikasi yang dapat mengaksesnya. Selanjutnya, rute-rute untuk BukuController disediakan, termasuk rute index untuk menampilkan daftar buku, create untuk menampilkan formulir pembuatan buku, store untuk menyimpan data buku baru, edit

untuk menampilkan formulir edit buku, update untuk memperbarui informasi buku yang ada, dan delete untuk menghapus buku menggunakan metode HTTP yang sesuai seperti GET, POST, dan DELETE. Selanjutnya terdapat juga rute untuk UserController, yang memiliki rute index untuk menampilkan profil pengguna atau daftar pengguna, login untuk proses autentikasi pengguna, dan logout untuk proses keluar dari sesi pengguna. Memiliki juga rute-rute untuk RegisterController, termasuk rute index untuk menampilkan formulir pendaftaran pengguna baru untuk menyimpan data pendaftaran tersebut.

Pada baris [29] [30] [31], syntax

```
if (is_file(APPPATH . 'Config/' . ENVIRONMENT .
'/Routes.php')) {
    require APPPATH . 'Config/' . ENVIRONMENT .
'/Routes.php';
}
```

Penggunaan baris-baris tersebut untuk memeriksa apakah ada file khusus rute untuk lingkungan tertentu. Apabila ada, file tersebut dimuat untuk menggantikan atau menambahkan rute yang ada.

C. Source Code *BaseController.php*

Table 28. Source Code Modul 6 *BaseController.php*

1	<?php
2	
3	namespace App\Controllers;
4	
5	use CodeIgniter\Controller;
6	use CodeIgniter\HTTP\CLIRequest;
7	use CodeIgniter\HTTP\IncomingRequest;
8	use CodeIgniter\HTTP\RequestInterface;

9	use CodeIgniter\HTTP\ResponseInterface;
10	use Psr\Log\LoggerInterface;
11	
12	abstract class BaseController extends Controller
13	{
14	/**
15	* Instance of the main Request object.
16	*
17	* @var CLIRequest IncomingRequest
18	*/
19	protected \$request;
20	
21	/**
22	* An array of helpers to be loaded automatically
	upon
23	* class instantiation. These helpers will be
	available
24	* to all other controllers that extend
	BaseController.
25	*
26	* @var array
27	*/
28	protected \$helpers = [];
29	
30	public function initController(RequestInterface
	\$request, ResponseInterface \$response,
	LoggerInterface \$logger)
31	{
32	
33	

	<code>parent::initController(\$request, \$response,</code>
34	<code>\$logger);</code>
35	
36	<code>}</code>
	<code>}</code>

D. Pembahasan *BaseController.php*

Pada baris [1] [2] [3] [4] [5] [6] [7] [8] [9] [10], syntax

```
<?php
```

```
namespace App\Controllers;
```

```
use CodeIgniter\Controller;
```

```
use CodeIgniter\HTTP\CLIRequest;
```

```
use CodeIgniter\HTTP\IncomingRequest;
```

```
use CodeIgniter\HTTP\RequestInterface;
```

```
use CodeIgniter\HTTP\ResponseInterface;
```

```
use Psr\Log\LoggerInterface;
```

penggunaan baris-baris tersebut terdapat untuk mendefinisikan namespace `App\Controllers` yang menunjukkan bahwa kelas `BaseController` berada dalam direktori `App\Controllers`. Pada baris tersebut juga terdapat use statements yang digunakan untuk mengimpor kelas-kelas yang diperlukan dari framework CodeIgniter dan PSR-3 Logger untuk digunakan dalam kelas `BaseController`.

Pada baris [12], syntax `abstract class BaseController extends Controller` digunakan untuk menyatakan bahwa `BaseController` adalah kelas abstrak yang mengimplementasikan `Controller` dari CodeIgniter. Hal ini berarti semua kontroler dalam aplikasi yang meng-extend `BaseController` akan mewarisi fungsionalitas dasar dari `Controller`.

Pada baris [13] [14] [15] [16] [17] [18] [19], syntax

```
{  
  
    /**  
  
     * Instance of the main Request object.  
  
     *  
  
     * @var CLIRequest|IncomingRequest  
  
     */  
  
    protected $request;
```

penggunaan properti tersebut untuk menyimpan instance dari objek utama Request. Hal ini dapat berupa CLIRequest yang digunakan untuk permintaan CLI atau IncomingRequest yang digunakan untuk permintaan HTTP.

Pada baris [21] [22] [23] [24] [25] [26] [27] [28], syntax

```
/**  
  
     * An array of helpers to be loaded automatically upon  
  
     * class instantiation. These helpers will be available  
  
     * to all other controllers that extend BaseController.  
  
     *  
  
     * @var array  
  
     */  
  
    protected $helpers = [];
```

penggunaan properti tersebut yang berisi daftar helpers atau bantuan yang akan dimuat secara otomatis saat kelas BaseController di-instantiasi. Helpers ini akan tersedia untuk semua kontroler lain yang meng-extend BaseController.

Pada baris [31] [32] [33] [34] [35] [36], syntax

```
public function initController(RequestInterface $request,
ResponseInterface $response, LoggerInterface $logger)
{
    parent::initController($request, $response,
$logger);
}
}
```

pada baris-baris tersebut merupakan metode inisialisasi yang akan dipanggil secara otomatis oleh framework saat sebuah kontroler di-instanstiasi. Metode tersebut menerima tiga parameter, yaitu \$request atau objek Request, \$response atau objek Response, dan \$logger atau objek Logger. Lalu pada baris tersebut juga memanggil metode `initController()` dari kelas induk (`Controller`) menggunakan `parent::initController()` untuk menginisialisasi kelas `Controller`.

E. Source Code BukuController.php

Table 29. Source Code Modul 6 BukuController.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Controllers\BaseController;
6	use App\Models\BukuModel;
7	
8	class BukuController extends BaseController

9	{
10	public function index()
11	{
12	return view('buku/index', [
13	'buku' => (new BukuModel())->findAll()
14]);
15	}
16	
17	public function create()
18	{
19	return view('buku/create');
20	}
21	
22	public function store()
23	{
24	\$buku = new BukuModel();
25	
26	\$buku->insert([
27	'judul' => \$this->request->getPost('judul'),
28	'penulis' => \$this->request-
29	>getPost('penulis'),
29	'penerbit' => \$this->request-
30	>getPost('penerbit'),
30	'tahun_terbit' => \$this->request-
31	>getPost('tahun_terbit'),
31]);
32	
33	return redirect()->to('/buku');
34	}
35	

36	public function edit(\$id)
37	{
38	return view('buku/edit', [
39	'buku' => (new BukuModel())->find(\$id)
40]);
41	}
42	
43	public function update(\$id)
44	{
45	\$buku = new BukuModel();
46	
47	\$buku->update(\$id, [
48	'judul' => \$this->request->getPost('judul'),
49	'penulis' => \$this->request-
	>getPost('penulis'),
50	'penerbit' => \$this->request-
	>getPost('penerbit'),
51	'tahun_terbit' => \$this->request-
	>getPost('tahun_terbit'),
52]);
53	
54	return redirect()->to('/buku');
55	}
56	
57	public function delete(\$id)
58	{
59	\$buku = new BukuModel();
60	
61	\$buku->delete(\$id);
62	

63	return redirect()->to('/buku');
64	}
65	}

F. Pembahasan BukuController.php

Pada baris [1] [2] [3] [4] [5] [6], syntax

```
<?php
```

```
namespace App\Controllers;
```

```
use App\Controllers\BaseController;
```

```
use App\Models\BukuModel;
```

pada baris-baris terdapat beberap apenggunaan, terdapat penggunaan untuk mendefinisikan namespace App\Controllers, yang menunjukkan bahwa kelas BukuController berada dalam direktori App\Controllers. Selanjutnya, terdapat penggunaan kode mengimpor BaseController untuk meng-extend dari kelas dasar yang telah dijelaskan sebelumnya. Lalu terdapat juga kode untuk mengimpor BukuModel, yang merupakan model untuk entitas buku. Model ini bertanggung jawab untuk mengakses dan memanipulasi data dari tabel buku dalam database.

Pada baris [8], syntax class BukuController extends BaseController memberikan keterangan bahwa BukuController meng-extend atau mewarisi fungsionalitas dari BaseController.

Pada abris [10] [11] [12] [13] [14] [15], syntax public function index()

```
{
    return view('buku/index', [
        'buku' => (new BukuModel())->findAll()
    ]);
}
```

```

    });
}

```

penggunaan baris-baris tersebut untuk mengatasi permintaan untuk halaman indeks buku dengan menggunakan BukuModel untuk mengambil semua data buku menggunakan metode findAll(). Data yang diambil kemudian dikirimkan ke view buku/index, di mana view akan menerima data tersebut dalam variabel \$buku untuk ditampilkan kepada pengguna.

Pada baris [17] [18] [19] [20], syntax

```

public function create()
{
    return view('buku/create');
}

```

Penggunaan baris-baris tersebut memiliki tanggung jawab atas permintaan untuk menampilkan formulir pembuatan buku dengan mengembalikan view buku/create. View ini berisi formulir kosong yang memungkinkan pengguna untuk menambahkan informasi buku baru sesuai dengan kebutuhan.

Pada baris [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34], syntax

```

public function store()
{
    $buku = new BukuModel();

    $buku->insert([
        'judul' => $this->request->getPost('judul'),
        'penulis' => $this->request->getPost('penulis'),
    ]);
}

```



```

        'penerbit' => $this->request-
>getPost('penerbit'),

        'tahun_terbit' => $this->request-
>getPost('tahun_terbit'),

    ]);

    return redirect()->to('/buku');

}

```

Penggunaan baris-baris tersebut digunakan untuk mengatasi permintaan untuk menyimpan data buku baru yang dikirimkan melalui formulir. Dengan menggunakan kode-kode tersebut akan dilakukan pembuatan instance BukuModel untuk memanipulasi data buku. Kemudian, menggunakan metode insert() dari BukuModel untuk menambahkan data buku baru ke dalam database. Setelah data berhasil disimpan, pengguna langsung diarahkan kembali ke halaman indeks buku, memastikan bahwa proses penambahan buku berjalan lancar.

Pada baris [37] [38] [39] [40] [41], syntax

```

public function edit($id)

{

    return view('buku/edit', [

        'buku' => (new BukuModel())->find($id)

    ]);

}

```

penggunaan baris-baris tersebut digunakan untuk menangani permintaan untuk menampilkan formulir edit buku berdasarkan ID tertentu dengan menggunakan BukuModel. Dengan menggunakan kode-kode tersebut akan dilakukan pencarian buku menggunakan metode find(\$id) dari BukuModel untuk mengambil data buku yang

sesuai dengan ID yang diminta. Setelah data buku ditemukan, informasi buku tersebut kemudian dikirimkan ke view buku/edit.

Pada baris [43] [44] [45] [46] [47] [48] [49] [50] [1] [52] [53] [54], syntax

```
public function update($id)
{
    $buku = new BukuModel();

    $buku->update($id, [
        'judul' => $this->request->getPost('judul'),
        'penulis' => $this->request->getPost('penulis'),
        'penerbit' => $this->request->getPost('penerbit'),
        'tahun_terbit' => $this->request->getPost('tahun_terbit'),
    ]);

    return redirect()->to('/buku');
}
```

penggunaan baris-baris tersebut digunakan untuk mengatasi permintaan untuk menyimpan perubahan pada buku dengan ID tertentu yang dilakukan melalui formulir edit. Proses dimulai dengan menggunakan BukuModel untuk memanggil metode update() dengan parameter ID buku yang ingin diperbarui dan data yang telah diubah.

Setelah perubahan berhasil diterapkan pada database, pengguna akan diarahkan kembali ke halaman indeks buku.

Pada baris [57] [58] [59] [60] [61] [62] [63] [64] [65], syntax

```
public function delete($id)
{
    $buku = new BukuModel();

    $buku->delete($id);

    return redirect()->to('/buku');
}
```

penggunaan baris-baris tersebut digunakan untuk mengatasi permintaan untuk menghapus buku berdasarkan ID tertentu dengan menggunakan BukuModel. Proses yang akan dilakukan dengan kode-kode tersebut adalah akan memanggil metode delete() dari BukuModel dengan parameter ID buku yang akan dihapus. Setelah buku berhasil dihapus dari database, pengguna akan diarahkan kembali ke halaman indeks buku.

G. Source Code *Home.php*

Table 30. Source Code Modul 6 *Home.php*

1	<?php
2	
3	namespace App\Controllers;
4	
5	class Home extends BaseController

6	{
7	public function index()
8	{
9	return view('welcome_message');
10	}
11	}

H. Pembahasan *Home.php*

Pada baris [1] [2] [3], syntax

```
<?php
```

```
namespace App\Controllers;
```

pada baris tersebut memiliki fungsi untuk mendefinisikan namespace App\Controllers. Sehingga menunjukkan bahwa kelas Home berada di dalam direktori App\Controllers.

Pada baris [5], syntax `class Home extends BaseController` memiliki fungsi untuk mendeklarasikan kelas Home yang meng-extend atau mewarisi fungsionalitas dari BaseController. BaseController itu sendiri merupakan kelas dasar yang memuat fungsionalitas dasar yang dibutuhkan oleh semua kontroler dalam aplikasi.

Pada baris [6] [7] [8] [9] [10] [11], syntax

```
{
    public function index()
    {
        return view('welcome_message');
    }
}
```

penggunaan baris-baris tersebut berisikan metode `index()`, metode `index` adalah metode publik yang menangani permintaan untuk halaman utama atau halaman indeks dari kontroler `Home`. Metode ini menggunakan fungsi `view()` untuk memuat view yang disebut `'welcome_message'`. View ini dapat berisi markup HTML dan elemen-elemen tampilan halaman depan aplikasi, berfungsi sebagai halaman selamat datang yang diperlihatkan kepada pengguna saat mereka mengunjungi aplikasi atau situs web tersebut.

I. Source Code RegisterController.php

Table 31. Source Code Modul 6 *RegisterController.php*

1	<code><?php</code>
2	
3	<code>namespace App\Controllers;</code>
4	
5	<code>use App\Models\UserModel;</code>
6	
7	<code>class RegisterController extends BaseController</code>
8	<code>{</code>
9	<code> protected \$model;</code>
10	
11	<code> public function __construct()</code>
12	<code> {</code>
13	<code> \$this->model = new UserModel();</code>
14	<code> \$this->helpers = ['form', 'url'];</code>
15	<code> }</code>
16	
17	<code> public function index()</code>
18	<code> {</code>
19	<code> \$data = [</code>
20	<code> 'title' => 'Register'</code>

21];
22	
23	return view('register', \$data);
24	}
25	
26	public function store()
27	{
28	\$data = \$this->request->getPost(['username',
	'email', 'password']);
29	
30	if (! \$this->validate([
31	'username' =>
	'required min_length[5] max_length[20] is_unique[user.
	username]',
32	'email' =>
	'required valid_email is_unique[user.email]',
33	'password' => 'required min_length[8]',
34])) {
35	return redirect()->back()->withInput()-
	>with('errors', \$this->validator->getErrors());
36	}
37	
38	\$data['password'] =
	password_hash(\$data['password'], PASSWORD_DEFAULT);
39	
40	\$save = \$this->model->save(\$data);
41	
42	if (\$save) {
43	session()->setFlashdata('success',
	'Register Berhasil!');

44	return redirect()->to(base_url('login'));
45	} else {
46	session()->setFlashdata('error', \$this->model->errors());
47	return redirect()->back()->withInput();
48	}
49	}
50	}

J. Pembahasan RegsiterController.php

Pada baris [1] [2] [3] [4] [5], syntax

```
<?php
```

```
namespace App\Controllers;
```

```
use App\Models\UserModel;
```

penggunaan baris-baris tersebut digunakan untuk mendefinisikan namespace App\Controllers, yang menunjukkan bahwa kelas RegisterController terletak di dalam direktori App\Controllers. Selanjutnya, kode mengimpor kelas UserModel yang berfungsi untuk berinteraksi dengan data pengguna atau users dalam database.

Pada baris [7], syntax class RegisterController extends BaseController digunakan untuk mendeklarasikan kelas RegisterController yang meng-extend atau mewarisi fungsionalitas dari BaseController.

Pada baris [8] [9] [10] [11] [12] [13] [14] [15], syntax

```
{
    protected $model;

    public function __construct()
    {
```

```

$this->model = new UserModel();

$this->helpers = ['form', 'url'];

}

```

pada baris-baris tersebut merupakan properti `$model`, objek `UserModel` yang digunakan untuk melakukan operasi-operasi terkait database terhadap entitas pengguna (users). Konstruktor (`__construct()`) bertanggung jawab atas inisialisasi objek `$model`, memastikan bahwa kontroler dapat berinteraksi dengan data pengguna dengan efisien. Selain itu, konstruktor juga mengatur `$helpers` untuk memuat helper form dan url, yang akan tersedia untuk digunakan dalam kontroler ini.

Pada abris [17] [18] [19] [20] [21] [22] [23] [24], syntax

```

public function index()

{

    $data = [

        'title' => 'Register'

    ];

    return view('register', $data);

}

```

Pada baris-baris tersebut merupakan metode `index()` yang menangani permintaan untuk menampilkan halaman registrasi dengan membuat array `$data` yang berisi judul halaman (`'title' => 'Register'`). Selanjutnya, metode ini juga mengembalikan view `'register'` dengan data `$data`, yang akan menampilkan formulir registrasi dengan judul "Register".

Pada baris [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36][37] [38][39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50], syntax

```

public function store()

```



```

{
    $data = $this->request->getPost(['username',
'email', 'password']);

    if (! $this->validate([
        'username' =>
'required|min_length[5]|max_length[20]|is_unique[user.use
rname]',
        'email' =>
'required|valid_email|is_unique[user.email]',
        'password' => 'required|min_length[8]',
    ])) {
        return redirect()->back()->withInput()-
>with('errors', $this->validator->getErrors());
    }

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

    $save = $this->model->save($data);

    if ($save) {
        session()->setFlashdata('success', 'Register
Berhasil!');
    }
}

```

```

        return redirect()->to(base_url('login'));

    } else {

        session()->setFlashdata('error',          $this-
>model->errors());

        return redirect()->back()->withInput();

    }

}
}

```

pada baris-baris tersebut merupakan penggunaan metode store() yang menangani proses penyimpanan data registrasi pengguna setelah formulir registrasi dikirimkan. Data yang dikirimkan diambil menggunakan \$this->request->getPost(), kemudian divalidasi dengan \$this->validate() untuk memastikan bahwa username dan email unik serta memenuhi persyaratan lainnya seperti panjang minimum dan validitas email. Apabila validasi gagal, pengguna diarahkan kembali ke halaman sebelumnya dengan data input sebelumnya dan pesan kesalahan. Namun, apabila validasi berhasil, password di-hash menggunakan password_hash() sebelum disimpan ke database. Data kemudian disimpan menggunakan \$this->model->save(\$data). Saat penyimpanan berhasil, pengguna diarahkan ke halaman login dengan pesan sukses; dan apabila gagal, maka pengguna diarahkan kembali ke halaman registrasi dengan pesan kesalahan.

K. Source UserController.php

Table 32. Source Code Modul 6 *UserController.php*

1	<?php
2	
3	namespace App\Controllers;
4	

5	use App\Controllers\BaseController;
6	use App\Models\UserModel;
7	
8	class UserController extends BaseController
9	{
10	public function index()
11	{
12	return view('login');
13	}
14	
15	public function login()
16	{
17	\$user = new UserModel();
18	\$username = \$this->request->getPost('username');
19	\$pw = \$this->request->getPost('password');
20	\$dataUser = \$user->where(['username' => \$username])->orWhere(['email' => \$username])->first();
21	if (\$dataUser) {
22	if (password_verify(\$pw, \$dataUser['password'])) {
23	session()->set([
24	'username' => \$dataUser['username'],
25	'email' => \$dataUser['email'],
26	'logged_in' => true
27]);
28	
29	return redirect()->to(base_url('buku'));

30	} else {
31	session()->setFlashdata('pesan',
	'Password salah');
32	return redirect()-
	>to(base_url('login'));
33	}
34	} else {
35	session()->setFlashdata('pesan', 'Username
	atau email tidak ditemukan');
36	return redirect()->to(base_url('login'));
37	}
38	}
39	
40	public function logout()
41	{
42	session()->destroy();
43	return redirect()->to(base_url('login'));
44	}
45	}

L. Pembahasan *UserController.php*

Pada Pada baris [1] [2] [3] [4] [5] [6], syntax

```
<?php
```

```
namespace App\Controllers;
```

```
use App\Controllers\BaseController;
```

```
use App\Models\UserModel;
```

pada baris tersebut digunakan untuk mendefinisikan namespace App\Controllers, yang menunjukkan bahwa kelas UserController berada di dalam direktori App\Controllers.

Selain itu, kode tersebut juga digunakan untuk mengimpor kelas BaseController dan UserModel, dimana BaseController menyediakan fungsionalitas dasar untuk semua kontroler, dan UserModel digunakan untuk berinteraksi dengan tabel pengguna dalam database.

Pada baris [8], syntax `class UserController extends BaseController` digunakan untuk mendeklarasikan kelas UserController yang meng-extend BaseController.

Pada baris [9] [10] [11] [12] [13], syntax

```
{  
  
    public function index()  
    {  
  
        return view('login');  
  
    }  
}
```

pada baris-baris tersebut merupakan Metode `index()` digunakan untuk menampilkan halaman login dengan mengembalikan view 'login', yang mungkin berisi formulir login.

Pada baris [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38], syntax

```
public function login()  
  
    {  
  
        $user = new UserModel();  
  
        $username = $this->request->getPost('username');  
  
        $pw = $this->request->getPost('password');  
  
        $dataUser = $user->where(['username' =>  
$username])->orWhere(['email' => $username])->first();  
  
    }  
}
```

```

        if ($dataUser) {

            if (password_verify($pw,
$dataUser['password'])) {

                session()->set([

                    'username' => $dataUser['username'],

                    'email' => $dataUser['email'],

                    'logged_in' => true

                ]);

                return redirect()->to(base_url('buku'));

            } else {

                session()->setFlashdata('pesan',
'Password salah');

                return redirect()->to(base_url('login'));

            }

        } else {

            session()->setFlashdata('pesan',      'Username
atau email tidak ditemukan');

            return redirect()->to(base_url('login'));

        }

    }
}

```

pada baris-baris tersebut merupakan metode login(), yang digunakan untuk menangani proses autentikasi pengguna dengan mengambil data username dan password dari

formulir login menggunakan `$this->request->getPost()`. Metode ini akan mencari pengguna berdasarkan username atau email menggunakan `UserModel`. Apabila pengguna ditemukan, verifikasi password dilakukan menggunakan `password_verify()`. Apabila verifikasi berhasil, data pengguna disimpan dalam sesi dan pengguna diarahkan ke halaman 'buku'. Apabila verifikasi gagal, pesan kesalahan disimpan dalam sesi dan pengguna diarahkan kembali ke halaman login. Lalu apabila pengguna tidak ditemukan, pesan kesalahan juga disimpan dalam sesi dan pengguna diarahkan kembali ke halaman login.

Pada baris [40] [41] [42] [43] [44] [45], syntax

```
public function logout()

    {

        session()->destroy();

        return redirect()->to(base_url('login'));

    }

}
```

pada baris-baris tersebut merupakan metode `logout()` digunakan untuk mengakhiri sesi pengguna yang sedang login dengan menghancurkan sesi pengguna menggunakan `session()->destroy()` dan kemudian mengarahkan pengguna kembali ke halaman login.

M. Source Code *AuthFilter.php*

Table 33. Source Code Modul 6 *AuthFilter.php*

1	<?php
2	
3	namespace App\Filters;
4	
5	use CodeIgniter\Filters\FilterInterface;
6	use CodeIgniter\HTTP\RequestInterface;

7	use CodeIgniter\HTTP\ResponseInterface;
8	
9	class AuthFilter implements FilterInterface
10	{
11	public function before(RequestInterface \$request,
	\$arguments = null)
12	{
13	if(!session()->get('logged_in')){
14	session()->setFlashdata('pesan', 'Login
	terlebih dahulu!');
15	return redirect()->to(base_url('login'));
16	}
17	}
18	
19	public function after(RequestInterface \$request,
	ResponseInterface \$response, \$arguments = null)
20	{
21	// Do something here
22	}
23	}

N. Pembahasan *AuthFilter.php*

Pada baris [1] [2] [3] [4] [5] [6] [7], syntax

```
<?php
```

```
namespace App\Filters;
```

```
use CodeIgniter\Filters\FilterInterface;
```

```
use CodeIgniter\HTTP\RequestInterface;
```

```
use CodeIgniter\HTTP\ResponseInterface;
```


pada baris-baris tersebut digunakan untuk mendefinisikan namespace App\Filters, menunjukkan bahwa kelas AuthFilter berada di dalam direktori App\Filters. Kode-kode tersebut juga mengimpor antarmuka FilterInterface serta kelas RequestInterface dan ResponseInterface dari CodeIgniter. Penggunaan FilterInterface untuk menentukan metode yang harus diimplementasikan oleh filter.

Pada baris [9], syntax `class AuthFilter implements FilterInterface` digunakan untuk mendeklarasikan kelas AuthFilter yang mengimplementasikan FilterInterface.

Pada baris [10] [11] [12] [13] [14] [15] [16] [17], syntax

```
{  
  
    public function before(RequestInterface $request,  
$arguments = null)  
  
    {  
  
        if(!session()->get('logged_in')){  
  
            session()->setFlashdata('pesan', 'Login  
terlebih dahulu!');  
  
            return redirect()->to(base_url('login'));  
  
        }  
  
    }  
}
```

pada baris-baris tersebut merupakan deklarasi metode before() yang dieksekusi sebelum permintaan HTTP mencapai kontroler, memeriksa apakah sesi pengguna memiliki nilai logged_in yang bernilai true. Apabila pengguna belum login, nilai logged_in tidak ada atau false, maka metode ini mengatur pesan flash menggunakan session()->setFlashdata('pesan', 'Login terlebih dahulu!') dan mengarahkan pengguna ke halaman login dengan redirect()->to(base_url('login')).

Pada baris [19] [20] [21] [22] [23], syntax

```
public function after(RequestInterface $request,
ResponseInterface $response, $arguments = null)
{
    // Do something here
}
}
```

pada baris-baris tersebut merupakan deklarasi metode `after()` yang dieksekusi setelah permintaan HTTP diproses oleh kontroler dan sebelum respons dikirim ke klien. Dalam kode ini, metode `after()` tidak melakukan apa-apa, tetapi disediakan untuk menambahkan logika pasca-pemrosesan. Ini bisa berarti mengevaluasi hasil dari respons yang telah dikirim atau menambahkan informasi tambahan ke dalam respons sebelum dikirim kembali ke klien.

O. Source Code *Create.php*

Table 34. Source Code Modul 6 *Create.php*

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Home</title>
9	

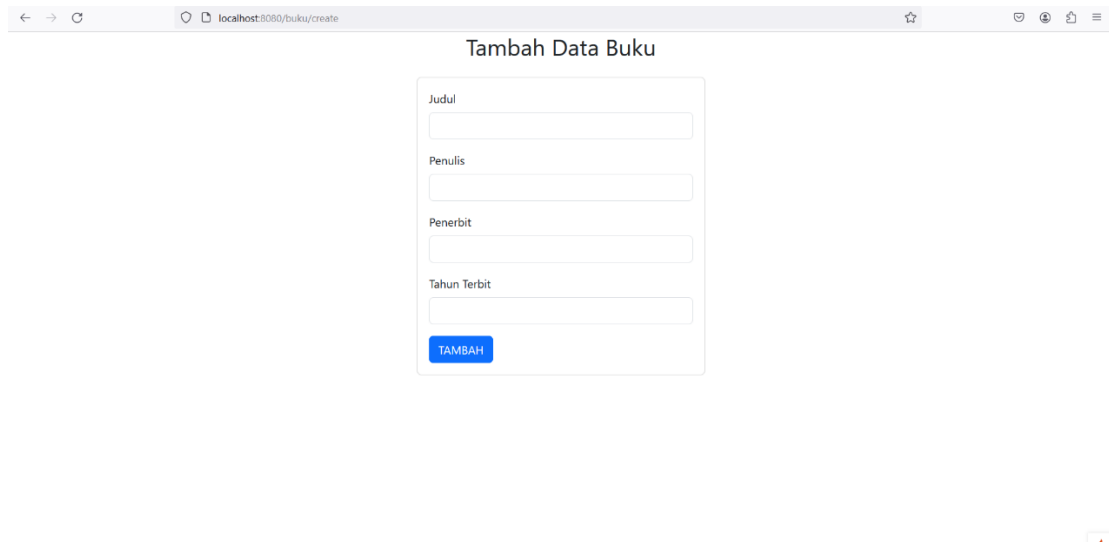
10	<pre> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0- alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384- </pre>
11	<pre> KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5IL </pre>
12	<pre> y+dN9+nJOZ" crossorigin="anonymous"> </pre>
13	
14	<pre> <style> </pre>
15	<pre> .card-form { </pre>
16	<pre> max-width: 400px; </pre>
17	<pre> } </pre>
18	<pre> </style> </pre>
19	<pre> </head> </pre>
20	
21	<pre> <body> </pre>
22	<pre> <div class="container"> </pre>
23	
24	<pre> <div class="row justify-content-center"> </pre>
25	<pre> <div class="col-12"> </pre>
26	<pre> <center><h2 class="mb-4">Tambah Data </pre>
27	<pre> Buku</h2></center> </pre>
28	<pre> </div> </pre>
29	<pre> <div class="col-6"> </pre>
30	<pre> <div class="card card-form mx-auto"> </pre>
31	<pre> <!-- Card view dan posisinya di tengah halaman --> </pre>
32	<pre> <div class="card-body"> </pre>
	<pre> <form action="<?= </pre>
	<pre> base_url('/buku/create') ?>" method="post" </pre>
	<pre> onsubmit="return validateForm()"> </pre>

	<input <="" td="" type="text"/>
50	class="form-control" id="tahun_terbit"
51	name="tahun_terbit">
52	<small< td=""></small<>
	id="tahun_terbitError" class="text-danger"></small>
53	</div>
54	
55	<button <="" td="" type="submit"></button>
56	class="btn btn-primary">TAMBAH</button>
57	</form>
58	</div>
59	</div>
60	</div>
61	
62	</div>
63	</div>
64	<script>
	function validateForm() {
65	var judul =
	document.getElementById("judul").value;
66	var penulis =
	document.getElementById("penulis").value;
67	var penerbit =
68	document.getElementById("penerbit").value;
	var tahun_terbit =
69	document.getElementById("tahun_terbit").value;
70	var judulError =
	document.getElementById("judulError");

71	var penulisError =
	document.getElementById("penulisError");
72	var penerbitError =
73	document.getElementById("penerbitError");
74	var tahun_terbitError =
75	document.getElementById("tahun_terbitError");
76	
	var valid = true;
77	
78	if (judul === "") {
79	judulError.innerText = "Judul harus
80	diisi";
81	valid = false;
82	} else {
83	judulError.innerText = "";
	}
84	
85	if (penulis === "") {
86	penulisError.innerText = "Penulis harus
87	diisi";
88	valid = false;
89	} else {
90	penulisError.innerText = "";
	}
91	
92	if (penerbit === "") {
93	penerbitError.innerText = "Penerbit
94	harus diisi";
95	valid = false;
96	} else {

97	penerbitError.innerText = "";
98	}
99	if (tahun_terbit === "") {
100	tahun_terbitError.innerText = "Tahun
101	terbit harus diisi";
102	valid = false;
103	} else if (isNaN(tahun_terbit)
104	tahun_terbit < 1800 tahun_terbit > 2024) {
105	tahun_terbitError.innerText = "Tahun
106	terbit harus berupa angka antara 1800 dan 2024";
107	valid = false;
108	} else {
109	tahun_terbitError.innerText = "";
110	}
111	return valid;
112	}
113	</script>
114	<script
115	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
116	alpha3/dist/js/bootstrap.bundle.min.js"
117	integrity="sha384-
118	ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7N
119	TKfAdVQSZe" crossorigin="anonymous">
120	</script>
121	</body>
122	</html>

P. Ouput Program *create.php*



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/buku/create'. The page title is 'Tambah Data Buku'. The form contains four text input fields labeled 'Judul', 'Penulis', 'Penerbit', and 'Tahun Terbit'. A blue button with the text 'TAMBAH' is positioned below the 'Tahun Terbit' field.

Gambar 32. Screenshot Hasil Jawaban Modul 6 *create.php*

Q. Pembahasan *Create.php*

Pada baris [1] – [17], syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

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

```
    <title>Home</title>
```

```
    <link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-  
alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
```



```

        integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+d
N9+nJOZ" crossorigin="anonymous">

<style>

        .card-form {

                max-width: 400px;

        }

</style>

</head>

```

pada baris-baris tersebut merupakan bagian dari dokumen HTML yang mendefinisikan bagian <head> dari halaman web. Elemen <head> dimulai dengan deklarasi dokumen dan bahasa: <!DOCTYPE html> menunjukkan bahwa dokumen ini menggunakan HTML5, sementara <html lang="en"> menunjukkan bahwa bahasa utama dokumen ini adalah bahasa Inggris, yang penting untuk SEO dan aksesibilitas. Karakter encoding didefinisikan dengan <meta charset="UTF-8">, yang menetapkan encoding dokumen sebagai UTF-8 untuk mendukung berbagai karakter dari berbagai bahasa. Untuk kompatibilitas browser, <meta http-equiv="X-UA-Compatible" content="IE=edge"> bertugas memberitahu browser untuk menggunakan mesin rendering terbaru atau Edge mode. Elemen <meta name="viewport" content="width=device-width, initial-scale=1.0"> mengatur viewport agar sesuai dengan lebar perangkat dan skala awal 1.0, memastikan tampilan yang responsif dan cocok pada berbagai ukuran layar. Lalu untuk judul halaman ditetapkan dengan <title>Home</title>, yang penting untuk tampilan pada tab browser dan SEO. Pada kode tersebut juga terdapat proses mengimpor Bootstrap CSS yang dilakukan dengan <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-

+nJOZ" crossorigin="anonymous">, dimana akan mengimpor file CSS dari Bootstrap versi 5.3.0-alpha3 dari CDN untuk membantu dalam membuat tata letak halaman yang responsif dan modern. Atribut integrity digunakan untuk keamanan, memverifikasi bahwa file yang diunduh tidak berubah, dan crossorigin="anonymous" menentukan bahwa permintaan sumber daya ini adalah permintaan cross-origin tanpa memerlukan kredensial. Pada kode tersebut juga terdapat penggunaan gaya CSS dengan menggunakan tag <style> untuk menambahkan kelas .card-form yang membatasi lebar maksimal elemen dengan kelas tersebut hingga 400px, css digunakan untuk memodifikasi gaya khusus yang tidak disediakan oleh Bootstrap.

Pada baris [19] – [59], syntax

```
<body>

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-12">

                <center><h2    class="mb-4">Tambah    Data
Buku</h2></center>

            </div>

            <div class="col-6">

                <div class="card card-form mx-auto"> <!--
Card view dan posisinya di tengah halaman -->

                    <div class="card-body">

                        <form                                action="<?=
base_url('/buku/create')                                ?>"                                method="post"
onsubmit="return validateForm()">

                            <div class="mb-3">
```

```

                                <label          for="judul"
class="form-label">Judul</label>

                                <input          type="text"
class="form-control" id="judul" name="judul">

                                <small      id="judulError"
class="text-danger"></small>

                                </div>

                                <div class="mb-3">

                                    <label          for="penulis"
class="form-label">Penulis</label>

                                    <input          type="text"
class="form-control" id="penulis" name="penulis">

                                    <small      id="penulisError"
class="text-danger"></small>

                                    </div>

                                    <div class="mb-3">

                                        <label          for="penerbit"
class="form-label">Penerbit</label>

                                        <input          type="text"
class="form-control" id="penerbit" name="penerbit">

                                        <small      id="penerbitError"
class="text-danger"></small>

                                        </div>

                                        <div class="mb-3">

```

```

                                <label for="tahun_terbit"
class="form-label">Tahun Terbit</label>

                                <input          type="text"
class="form-control"          id="tahun_terbit"
name="tahun_terbit">

                                <small
id="tahun_terbitError" class="text-danger"></small>

                                </div>

                                <button          type="submit"
class="btn btn-primary">TAMBAH</button>

                                </form>

                                </div>

                                </div>

                                </div>

                                </div>

                                </div>

```

pada baris-baris tersebut bertugas untuk mendefinisikan struktur dan tampilan sebuah halaman web menggunakan HTML dan Bootstrap untuk membuat form penambahan data buku. Pada kode tersebut dimulai dengan tag <body> sebagai bagian utama dari dokumen HTML yang menampung semua konten yang akan ditampilkan di halaman web, lalu <div class="container"> yang menggunakan kelas container dari Bootstrap untuk memberikan margin dan padding. Selanjutnya, <div class="row justify-content-center"> membuat baris grid dengan konten dipusatkan di dalamnya. Judul halaman ditampilkan menggunakan <div class="col-12"> yang mengisi seluruh lebar kontainer,

dan `<h2 class="mb-4">Tambah Data Buku</h2>` untuk teks judul dengan margin bawah. Formulir tambah data buku ditempatkan dalam `<div class="col-6">`, di dalamnya terdapat `<div class="card card-form mx-auto">` untuk membuat kartu dengan margin otomatis yang memposisikan kartu di tengah horizontal. Bagian `<div class="card-body">` mengelompokkan konten kartu, dan `<form action="<?= base_url('/buku/create') ?>" method="post" onsubmit="return validateForm()">` mengatur pengiriman data ke URL dengan metode POST, serta memanggil fungsi JavaScript `validateForm()` saat formulir dikirimkan untuk validasi. Selanjutnya tombol `<button type="submit" class="btn btn-primary">TAMBAH</button>` digunakan untuk mengirimkan formulir dengan gaya tombol biru dari Bootstrap. Setiap input dalam formulir mengikuti pola yang terdiri dari `<div class="mb-3">` untuk margin bawah, `<label for="judul" class="form-label">Judul</label>` untuk memberikan label pada input, `<input type="text" class="form-control" id="judul" name="judul">` untuk input teks, dan `<small id="judulError" class="text-danger"></small>` untuk menampilkan pesan error dengan gaya teks merah.

Pada baris [61] – [108], syntax

```
<script>

    function validateForm() {

        var                                judul                                =
document.getElementById("judul").value;

        var                                penulis                                =
document.getElementById("penulis").value;

        var                                penerbit                                =
document.getElementById("penerbit").value;

        var                                tahun_terbit                                =
document.getElementById("tahun_terbit").value;
```

```

        var                                judulError                                =
document.getElementById("judulError");

        var                                penulisError                                =
document.getElementById("penulisError");

        var                                penerbitError                                =
document.getElementById("penerbitError");

        var                                tahun_terbitError                                =
document.getElementById("tahun_terbitError");

        var valid = true;

        if (judul === "") {

            judulError.innerHTML    =    "Judul    harus
diisi";

            valid = false;

        } else {

            judulError.innerHTML = "";

        }

        if (penulis === "") {

            penulisError.innerHTML = "Penulis harus
diisi";

            valid = false;

        } else {

            penulisError.innerHTML = "";

        }

```

```

        if (penerbit === "") {

            penerbitError.innerText = "Penerbit harus
diisi";

            valid = false;

        } else {

            penerbitError.innerText = "";

        }

        if (tahun_terbit === "") {

            tahun_terbitError.innerText = "Tahun
terbit harus diisi";

            valid = false;

        } else if (isNaN(tahun_terbit) || tahun_terbit
< 1800 || tahun_terbit > 2024) {

            tahun_terbitError.innerText = "Tahun
terbit harus berupa angka antara 1800 dan 2024";

            valid = false;

        } else {

            tahun_terbitError.innerText = "";

        }

        return valid;

    }

```

```
</script>
```

pada baris-baris tersebut merupakan kode JavaScript yang digunakan untuk memvalidasi form input data buku sebelum form tersebut dikirimkan. Fungsi `validateForm()` bertugas memastikan semua input dalam form diisi dengan benar sebelum form dikirimkan. Fungsi ini dimulai dengan mendapatkan nilai dari input form: `var judul = document.getElementById("judul").value, var penulis = document.getElementById("penulis").value, var penerbit = document.getElementById("penerbit").value, var tahun_terbit = document.getElementById("tahun_terbit").value`. Kemudian, elemen-elemen kecil yang akan digunakan untuk menampilkan pesan error jika ada kesalahan input diambil: `var judulError = document.getElementById("judulError"), var penulisError = document.getElementById("penulisError"), var penerbitError = document.getElementById("penerbitError"), var tahun_terbitError = document.getElementById("tahun_terbitError")`. Variabel `valid` diinisialisasi sebagai `true` untuk melacak status validasi form dan akan diubah menjadi `false` jika ditemukan kesalahan input. Validasi dilakukan untuk setiap input: jika input judul kosong, tampilkan pesan error dan set `valid` ke `false`, jika tidak, kosongkan pesan error. Hal yang sama juga dilakukan untuk input penulis dan penerbit. Untuk input `tahun_terbit`, jika kosong, bukan angka, atau tidak dalam rentang 1800 hingga 2024, tampilkan pesan error dan set `valid` ke `false`, jika tidak, kosongkan pesan error. Fungsi tersebut akhirnya mengembalikan nilai `valid` yang menunjukkan apakah form valid atau tidak. Apabila `false`, form tidak akan dikirim.

Pada baris [110] – [114], syntax

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-  
alpha3/dist/js/bootstrap.bundle.min.js"
```



```

        integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKf
AdVQSZe" crossorigin="anonymous">

    </script>

</body>

</html>

```

Pada baris-baris tersebut merupakan library Bootstrap dari CDN (Content Delivery Network) dengan versi 5.3.0-alpha3.

R. Source Code *edit.php*

Table 35. Source Code Modul 6 *edit.php*

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Edit Data</title>
9	<link
	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
	alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
10	integrity="sha384-
	KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5IL
	y+dN9+nJOZ" crossorigin="anonymous">
11	
12	</head>

13	
14	<body>
15	<div class="container">
16	
17	
18	<div class="row">
19	<div class="col-12">
20	<center><h2 class="mb-4">Edit Data
21	Buku</h2></center>
	</div>
22	
23	<div class="col-md-6 mx-auto">
24	<div class="card">
25	<div class="card-body">
26	<form action="<?=
27	base_url('/buku/' . \$buku['id'] . '/edit') ?>"
	method="post"
	onsubmit="return
28	validateForm()">
	<div class="mb-3">
	<label for="judul"
29	class="form-label">Judul</label>
	<input type="text"
30	class="form-control" id="judul" name="judul"
	value="<?=
31	\$buku['judul'] ?>">
	<small id="judulError"
32	class="text-danger"></small>
	</div>
33	<div class="mb-3">

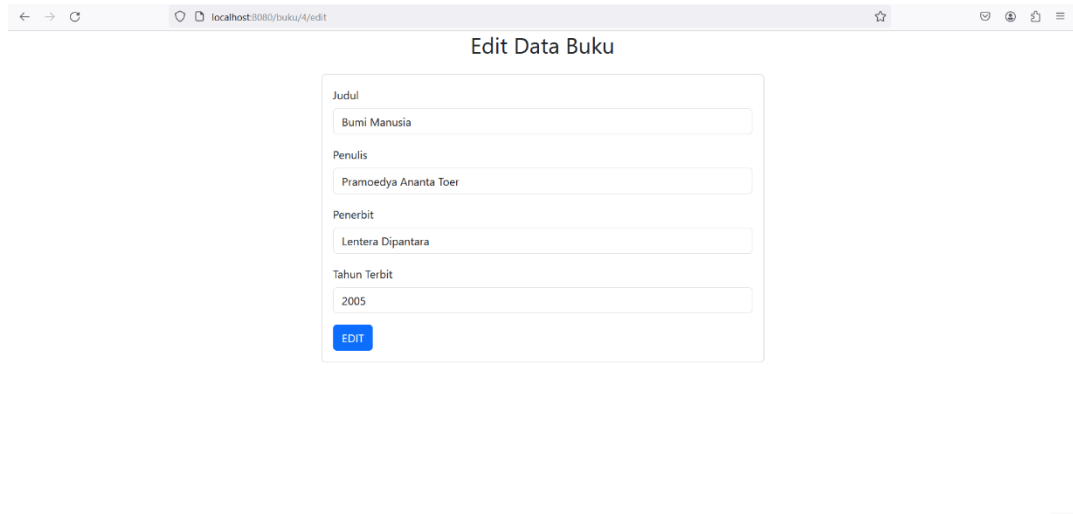
34	<label for="penulis"
35	class="form-label">Penulis</label>
	<input type="text"
36	class="form-control" id="penulis" name="penulis"
	value="<?=
37	\$buku['penulis'] ?>">
	<small
38	id="penulisError" class="text-danger"></small>
	</div>
39	<div class="mb-3">
40	<label for="penerbit"
41	class="form-label">Penerbit</label>
	<input type="text"
42	class="form-control" id="penerbit" name="penerbit"
	value="<?=
43	\$buku['penerbit'] ?>">
	<small
44	id="penerbitError" class="text-danger"></small>
	</div>
45	<div class="mb-3">
46	<label
47	for="tahun_terbit" class="form-label">Tahun
	Terbit</label>
	<input type="text"
48	class="form-control" id="tahun_terbit"
	name="tahun_terbit"
	value="<?=
49	\$buku['tahun_terbit'] ?>">
	<small
50	id="tahun_terbitError" class="text-danger"></small>

	</div>	
51		
52	<button type="submit"	
53	class="btn btn-primary">EDIT</button>	
	</form>	
54	</div>	
55	</div>	
56	</div>	
57		
58	</div>	
59	</div>	
60		
61	<script>	
62	function validateForm() {	
63	var judul	=
64	document.getElementById("judul").value;	
65	var penulis	=
66	document.getElementById("penulis").value;	
67	var penerbit	=
68	document.getElementById("penerbit").value;	
69	var tahun_terbit	=
70	document.getElementById("tahun_terbit").value;	
71	var judulError	=
	document.getElementById("judulError");	
	var penulisError	=
	document.getElementById("penulisError");	
	var penerbitError	=
	document.getElementById("penerbitError");	

72	var tahun_terbitError =
	document.getElementById("tahun_terbitError");
73	
74	var valid = true;
75	
76	if (judul === "") {
77	judulError.innerText = "Judul harus
	diisi";
78	valid = false;
79	} else {
80	judulError.innerText = "";
81	}
82	
83	if (penulis === "") {
84	penulisError.innerText = "Penulis harus
	diisi";
85	valid = false;
86	} else {
87	penulisError.innerText = "";
88	}
89	
90	if (penerbit === "") {
91	penerbitError.innerText = "Penerbit
	harus diisi";
92	valid = false;
93	} else {
94	penerbitError.innerText = "";
95	}
96	
97	if (tahun_terbit === "") {

98	tahun_terbitError.innerText = "Tahun terbit harus diisi";
99	valid = false;
100	} else if (isNaN(tahun_terbit) tahun_terbit < 1800 tahun_terbit > 2024) {
101	tahun_terbitError.innerText = "Tahun terbit harus berupa angka antara 1800 dan 2024";
102	valid = false;
103	} else {
104	tahun_terbitError.innerText = "";
105	}
106	
107	return valid;
108	}
109	</script>
110	<script
	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"
111	integrity="sha384-
	ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7N
	TKfAdV

S. Ouput Program *edit.php*



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/buku/4/edit'. The page title is 'Edit Data Buku'. The form contains the following data:

Field	Value
Judul	Bumi Manusia
Penulis	Pramoedya Ananta Toer
Penerbit	Lentera Dipantara
Tahun Terbit	2005

An 'EDIT' button is located at the bottom of the form.

Gambar 33. Screenshot Hasil Jawaban Modul 6 *edit.php*

T. Pembahasan *edit.php*

Pada baris [1] -15, syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

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

```
    <title>Edit Data</title>
```

```
    <link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
```

```

        integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+d
N9+nJOZ" crossorigin="anonymous">

</head>

```

pada baris-baris tersebut merupakan deklarasi Dokumen `<!DOCTYPE html>` dan `<html lang="en">` yang bertugas untuk membantu browser memahami dan merender halaman dengan benar sesuai standar HTML5. Penggunaan metadata yaitu `<meta charset="UTF-8">`, `<meta http-equiv="X-UA-Compatible">`, dan `<meta name="viewport">` digunakan untuk memberikan informasi penting kepada browser tentang karakter set, mode rendering, dan cara halaman harus ditampilkan di berbagai perangkat. Selanjutnya terdapat tag `<title>` memberikan judul halaman yang muncul di tab browser, memudahkan pengguna mengidentifikasi halaman tersebut. Penggunaan `<link href="...">` untuk menghubungkan halaman HTML dengan file CSS Bootstrap dari CDN sehingga dapat dilakukan penggunaan gaya dan tata letak yang konsisten dari Bootstrap.

Pada baris [14] – [60], syntax

```

<body>

    <div class="container">

        <div class="row">

            <div class="col-12">

                <center><h2      class="mb-4">Edit      Data
Buku</h2></center>

            </div>

            <div class="col-md-6 mx-auto">

                <div class="card">

```



```

        <div class="card-body">

            <form                                action="<?=
base_url('/buku/' . $buku['id'] . '/edit') ?>"
method="post"

                                onsubmit="return
validateForm()" ">

                <div class="mb-3">

                    <label                        for="judul"
class="form-label">Judul</label>

                    <input                        type="text"
class="form-control" id="judul" name="judul"

                                value="<?=
$buku['judul'] ?>">

                    <small      id="judulError"
class="text-danger"></small>

                </div>

                <div class="mb-3">

                    <label                        for="penulis"
class="form-label">Penulis</label>

                    <input                        type="text"
class="form-control" id="penulis" name="penulis"

                                value="<?=
$buku['penulis'] ?>">

                    <small      id="penulisError"
class="text-danger"></small>

```

```

        </div>

        <div class="mb-3">

            <label      for="penerbit"
class="form-label">Penerbit</label>

            <input      type="text"
class="form-control" id="penerbit" name="penerbit"

                value="<?=$buku['penerbit'] ?>">

            <small id="penerbitError"
class="text-danger"></small>

        </div>

        <div class="mb-3">

            <label for="tahun_terbit"
class="form-label">Tahun Terbit</label>

            <input      type="text"
class="form-control"      id="tahun_terbit"
name="tahun_terbit"

                value="<?=$buku['tahun_terbit'] ?>">

            <small
id="tahun_terbitError" class="text-danger"></small>

        </div>

        <button      type="submit"
class="btn btn-primary">EDIT</button>

    </form>

```

```

        </div>

    </div>

</div>

</div>

</div>

```

pada baris-baris tersebut merupakan bagian dari halaman HTML yang digunakan untuk mengedit data buku. <body> adalah elemen utama yang mengelilingi semua konten halaman, sementara .container digunakan untuk membatasi lebar konten halaman. .row digunakan bersama kelas Bootstrap untuk mengatur baris dalam halaman, yang dapat dibagi menjadi .col-12 untuk pengaturan penuh layar dan .col-md-6 untuk ukuran layar medium. .mx-auto diterapkan untuk menengahkan elemen secara horizontal di dalam parentnya. Elemen <center> digunakan untuk menengahkan teks, sementara <h2> menampilkan judul "Edit Data Buku". .card adalah komponen kartu Bootstrap yang mengelompokkan informasi dengan bingkai abu-abu yang sedikit dibulatkan. <form> digunakan untuk membuat formulir dengan action yang menentukan URL untuk pengeditan data buku dan method="post" untuk pengiriman data. Elemen <input> menerima input pengguna untuk judul, penulis, penerbit, dan tahun terbit, dengan class="form-control" untuk gaya Bootstrap. <small> menampilkan pesan error validasi, dan <button type="submit"> menyediakan tombol untuk mengirimkan formulir. Script JavaScript digunakan untuk validasi form, dan <script> eksternal memuat JavaScript Bootstrap dari CDN.

Pada baris [62] – [11], syntax

```

<script>

    function validateForm() {

        var                                judul                                =
document.getElementById("judul").value;

```

```

        var                penulis                =
document.getElementById("penulis").value;

        var                penerbit                =
document.getElementById("penerbit").value;

        var                tahun_terbit                =
document.getElementById("tahun_terbit").value;

        var                judulError                =
document.getElementById("judulError");

        var                penulisError                =
document.getElementById("penulisError");

        var                penerbitError                =
document.getElementById("penerbitError");

        var                tahun_terbitError                =
document.getElementById("tahun_terbitError");

        var valid = true;

        if (judul === "") {

            judulError.innerText    =    "Judul    harus
diisi";

            valid = false;

        } else {

            judulError.innerText = "";

        }

        if (penulis === "") {

```

```

        penulisError.innerText = "Penulis harus
diisi";

        valid = false;
    } else {

        penulisError.innerText = "";
    }

    if (penerbit === "") {

        penerbitError.innerText = "Penerbit harus
diisi";

        valid = false;
    } else {

        penerbitError.innerText = "";
    }

    if (tahun_terbit === "") {

        tahun_terbitError.innerText = "Tahun
terbit harus diisi";

        valid = false;

    } else if (isNaN(tahun_terbit) || tahun_terbit
< 1800 || tahun_terbit > 2024) {

        tahun_terbitError.innerText = "Tahun
terbit harus berupa angka antara 1800 dan 2024";

        valid = false;
    } else {

```

```

        tahun_terbitError.innerText = "";
    }

    return valid;
}

</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"

    integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKf
AdV

```

pada baris-baris tersebut merupakan kode JavaScript yang berfungsi untuk melakukan validasi formulir sebelum data dikirim ke server. Fungsi `validateForm()` memanfaatkan metode `document.getElementById("id_element").value` untuk mengambil nilai dari elemen input yang sesuai, seperti judul, penulis, penerbit, dan tahun terbit. Setiap elemen ini kemudian divalidasi: jika kosong, pesan kesalahan ditampilkan di elemen `<small>` yang sesuai, dan variabel `valid` diubah menjadi `false`. Tahun terbit juga diperiksa apakah berupa angka dan berada dalam rentang 1800-2024. `<script>` dengan `src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"` memuat file JavaScript Bootstrap dari CDN, yang menggunakan `integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"` untuk memastikan keaslian versi yang dimuat, dan `crossorigin="anonymous"` untuk permintaan lintas asal yang aman.

U. Source Code *index.php*

Table 36. Source Code Modul 6 *index.php*

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	<code> <meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>
6	<code> <meta name="viewport" content="width=device-</code>
	<code>width, initial-scale=1.0"></code>
7	<code> <title>Home</title></code>
8	<code> <style></code>
9	<code> body {</code>
10	<code> font-family: Arial, sans-serif;</code>
11	<code> background-color: #f8f9fa;</code>
12	<code> }</code>
13	
14	<code> .container {</code>
15	<code> max-width: 900px;</code>
16	<code> margin: 50px auto;</code>
17	<code> padding: 20px;</code>
18	<code> background-color: #fff;</code>
19	<code> border-radius: 10px;</code>
20	<code> box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);</code>
21	<code> }</code>
22	
23	<code> .btn-logout {</code>
24	<code> margin-top: 20px;</code>
25	<code> float: right;</code>
26	<code> }</code>

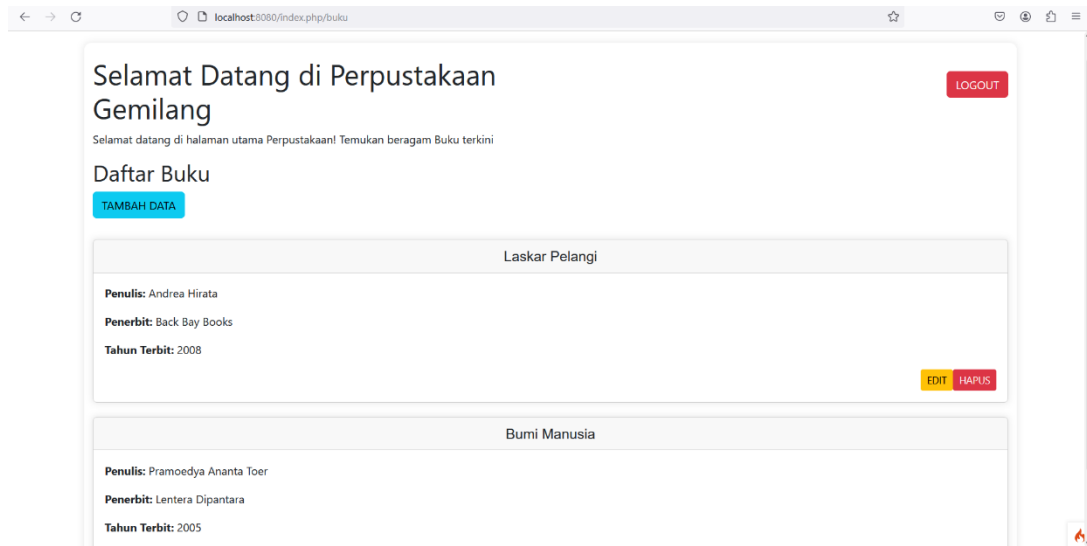
27	
28	.btn-info {
29	margin-bottom: 10px;
30	}
31	
32	.card {
33	margin-top: 20px;
34	border: 1px solid #dee2e6;
35	border-radius: .25rem;
36	box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
37	}
38	
39	.card-header {
40	background-color: #007bff;
41	color: #fff;
42	padding: 10px;
43	border-bottom: 1px solid rgba(0, 0, 0,
	0.125);
44	}
45	
46	.card-body {
47	padding: 20px;
48	}
49	
50	.table {
51	margin-top: 20px;
52	}
53	</style>
54	<link
	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-

	alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJJOZ" crossorigin="anonymous">
55	</head>
56	<body>
57	<div class="container">
58	<div class="row">
59	<div class="col">
60	<h1>Selamat Datang di Perpustakaan Gemilang</h1>
61	<p>Selamat datang di halaman utama Perpustakaan! Temukan beragam Buku terkini</p>
62	</div>
63	<div class="col">
64	LOGOUT
65	</div>
66	</div>
67	
68	<!-- create list of books -->
69	<div class="row">
70	<div class="col">
71	<h2>Daftar Buku</h2>
72	TAMBAH DATA
73	</div>
74	</div>
75	
76	<div class="row">

77	<div class="col">
78	<?php foreach(\$buku as \$key =>
	\$value): ?>
79	<div class="card">
80	<div class="card-header"
	style="font-family: Arial; font-size: 20px;">
81	<center><?= \$value['judul']
	?></center>
82	</div>
83	<div class="card-body">
84	<p>Penulis:
	<?= \$value['penulis'] ?></p>
85	<p>Penerbit:
	<?= \$value['penerbit'] ?></p>
86	<p>Tahun
	Terbit: <?= \$value['tahun_terbit'] ?></p>
87	<div class="d-flex justify-
	content-end">
88	<!-- hapus button -->
89	<a href="<?=
	base_url('/buku/' . \$value['id'] . '/edit') ?>"
	class="btn btn-warning btn-sm">EDIT
90	<form action="<?=
	base_url('/buku/' . \$value['id'] . '/delete') ?>"
	method="POST" style="display: inline-block;">
91	<input type="hidden"
	name="_method" value="DELETE">
92	<button type="submit"
	onclick="return confirm('Are you sure you want to

93	delete this item?')" class="btn btn-danger btn-
94	sm">HAPUS</button>
95	</form>
96	</div>
97	</div>
98	<?php endforeach; ?>
99	</div>
100	</div>
101	</div>
102	<script
103	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
	alpha3/dist/js/bootstrap.bundle.min.js"
	integrity="sha384-
	ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7
	NTKfAdVQSZe" crossorigin="anonymous"></script>
	</body>
	</html>

V. Output Program *index.php*



Gambar 34. Screenshot Hasil Jawaban Modul 6 *index.php*

W. Pembahasan *index.php*

Pada baris [1] – [55], syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

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

```
    <title>Home</title>
```

```
    <style>
```

```
        body {
```

```

        font-family: Arial, sans-serif;

        background-color: #f8f9fa;
    }

    .container {

        max-width: 900px;

        margin: 50px auto;

        padding: 20px;

        background-color: #fff;

        border-radius: 10px;

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

    .btn-logout {

        margin-top: 20px;

        float: right;
    }

    .btn-info {

        margin-bottom: 10px;
    }

    .card {

        margin-top: 20px;

        border: 1px solid #dee2e6;

        border-radius: .25rem;
    }

```

```

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

    .card-header {

        background-color: #007bff;

        color: #fff;

        padding: 10px;

        border-bottom: 1px solid rgba(0, 0, 0, 0.125);
    }

    .card-body {

        padding: 20px;
    }

    .table {

        margin-top: 20px;
    }

</style>

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css"          rel="stylesheet"
integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+d
N9+nJOZ" crossorigin="anonymous">

</head>

```

pada baris-baris tersebut merupakan kode HTML dan CSS yang ditunjukkan untuk memberikan gambaran tentang halaman web yang menggunakan Bootstrap 5 dengan

gaya tambahan lainnya. Deklarasi dokumen (`<!DOCTYPE html>` dan `<html lang="en">`) menetapkan jenis dokumen dan bahasa yang digunakan. Bagian `<head>` mencakup meta tags untuk karakter set, kompatibilitas IE, dan responsif, serta judul halaman. Penggunaan internal styles menetapkan gaya CSS langsung, sementara stylesheet eksternal mengimpor Bootstrap dari CDN untuk pengaturan gaya tambahan. Lalu penggunaan CSS untuk mengatur body dengan font dan latar belakang, container untuk tampilan terpusat dengan lebar maksimum dan bayangan kotak, dan tombol logout (`btn-logout`) ditempatkan di sudut kanan atas. Selain itu, gaya kartu dan tabel ditentukan untuk margin, bayangan, dan warna latar belakang yang konsisten.

Pada baris [56] – [73], syntax

```
<body>

    <div class="container">

        <div class="row">

            <div class="col">

                <h1>Selamat    Datang    di    Perpustakaan
Gemilang</h1>

                <p>Selamat    datang    di    halaman    utama
Perpustakaan! Temukan beragam Buku terkini</p>

            </div>

            <div class="col">

                <a    href="<?=    base_url('logout')    ?>"
class="btn btn-danger btn-logout">LOGOUT</a>

            </div>

        </div>

    </div>
```

```

<div class="row">

    <div class="col">

        <h2>Daftar Buku</h2>

        <a    class="btn    btn-info"    href="=
base_url('/buku/create') ?&gt;"&gt;TAMBAH DATA&lt;/a&gt;

    &lt;/div&gt;

&lt;/div&gt;
</pre

```

pada baris-baris tersebut merupakan penggunaan dari container `<div class="container">`) yang digunakan untuk membatasi konten agar tetap rapi di tengah halaman dengan lebar maksimum 900px, dan secara otomatis menyesuaikan jarak tepi (padding) dari layar. Grid system Bootstrap, seperti `<div class="row">` dan `<div class="col">`, memungkinkan pembagian halaman menjadi bagian-bagian yang responsif dan mudah diatur. Setiap `<div class="row">` digunakan untuk menandai baris baru di dalam container, sementara `<div class="col">` digunakan untuk membuat kolom di dalam baris, dengan lebar yang merata secara default. Pada halaman tersebut, elemen-elemen seperti header utama untuk menampilkan judul perpustakaan dan deskripsi singkat, tombol logout dengan gaya Bootstrap yang menonjol, judul untuk daftar buku, dan tombol tambah data untuk mengarahkan pengguna ke halaman tambah buku (`/buku/create`) membantu dalam navigasi dan tampilan yang terorganisir.

Pada baris [75] – [99], syntax

```

<div class="row">

    <div class="col">

        <?php foreach($buku as $key => $value): ?>

            <div class="card">

```



```

        <div class="card-header" style="font-
family: Arial; font-size: 20px;">

            <center><?=$value['judul']
?></center>

        </div>

        <div class="card-body">

            <p><strong>Penulis:</strong> <?=$value['penulis'] ?></p>

            <p><strong>Penerbit:</strong> <?=$value['penerbit'] ?></p>

            <p><strong>Tahun Terbit:</strong>
<?=$value['tahun_terbit'] ?></p>

            <div class="d-flex justify-
content-end">

                <a href="<?=$value['base_url'] . $value['id'] . '/edit'> ?>" class="btn
btn-warning btn-sm">EDIT</a>

                <form action="<?=$value['base_url'] . $value['id'] . '/delete'> ?>"
method="POST" style="display: inline-block;">

                    <input type="hidden"
name="_method" value="DELETE">

                    <button type="submit"
onclick="return confirm('Are you sure you want to delete
this item?')" class="btn btn-danger btn-sm">HAPUS</button>

```

```

        </form>

    </div>

</div>

</div>

<?php endforeach; ?>

</div>

</div>

</div>

```

pada baris-baris tersebut merupakan bagian dari HTML yang di-generate oleh PHP untuk menampilkan daftar buku dari variabel \$buku. Penggunaan konstruksi PHP `foreach($buku as $key => $value)` digunakan untuk mengiterasi setiap elemen dalam array \$buku, di mana setiap elemen direpresentasikan sebagai sebuah kartu (`<div class="card">`). Setiap kartu ini menampilkan informasi buku seperti judul, penulis, penerbit, dan tahun terbit. Bagian header kartu (`<div class="card-header">`) menampilkan judul buku dengan gaya tertentu, sedangkan dalam badan kartu (`<div class="card-body">`), informasi seperti penulis, penerbit, dan tahun terbit ditampilkan dalam paragraf (`<p>`). Setiap kartu juga dilengkapi dengan tombol EDIT (`<a>`) untuk mengarahkan pengguna ke halaman edit buku berdasarkan ID buku, serta form HAPUS (`<form>`) dengan tombol HAPUS untuk mengonfirmasi dan menghapus buku dari daftar.

Pada baris [100] – [102], syntax

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-

```

```

ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKf
AdVQSZe" crossorigin="anonymous"></script>

</body>

</html>

```

pada baris tersebut merupakan tag `<script>` yang digunakan untuk menyematkan kode JavaScript ke dalam halaman web. Pada kode tersebut, atribut `src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-`

`alpha3/dist/js/bootstrap.bundle.min.js"` menunjukkan bahwa browser harus mengunduh file JavaScript Bootstrap terbaru dari CDN JSDelivr. Atribut `integrity="sha384-`

`ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"` digunakan untuk memastikan integritas file yang diunduh dengan memverifikasi hash SHA-384 dari file tersebut. Hal ini membantu mencegah pengambilan file yang dimodifikasi secara tidak sah. Atribut `crossorigin="anonymous"` menetapkan bahwa permintaan untuk file JavaScript tidak akan mengirimkan kredensial pengguna

X. Source Code *login.php*

Table 37. Source Code Modul 6 *login.php*

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	
4	<code><head></code>
5	<code><meta charset="UTF-8"></code>
6	<code><meta http-equiv="X-UA-Compatible" content="IE=edge"></code>
7	<code><meta name="viewport" content="width=device-width,</code>
8	<code>initial-scale=1.0"></code>
9	<code><title>Login</title></code>
10	

	<pre> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0- alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384- 11 KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy 12 +dN9+nJOZ" crossorigin="anonymous"> 13 14 <style> 15 body { 16 background-color: #f8f9fa; 17 } 18 19 .card-view { 20 max-width: 400px; 21 margin: auto; 22 margin-top: 100px; 23 padding: 20px; 24 border-radius: 10px; 25 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); 26 background-color: #fff; 27 } 28 29 .card-view .card-title { 30 font-size: 24px; 31 font-weight: bold; 32 text-align: center; 33 margin-bottom: 20px; 34 } 35 36 .card-view form { </pre>
--	--

```

37 margin-bottom: 0;
38 }
39
40 .card-view .form-group {
41 margin-bottom: 20px;
42 }
43
44 .card-view .form-group label {
45 font-weight: bold;
46 }
47
48 .card-view .btn-primary {
49 width: 100%;
50 }
51
52 .card-view .mt-3 {
53 text-align: center;
54 }
55
56 </style>
57 </head>
58
59 <body>
60 <div class="container">
61   <div class="card-view">
62     <?php if (!empty(session()->getFlashdata('pesan'))):
63       ?>
64     <div class="alert alert-warning alert-dismissible fade
65 show" role="alert">
66       <?= session()->getFlashdata('pesan'); ?>

```

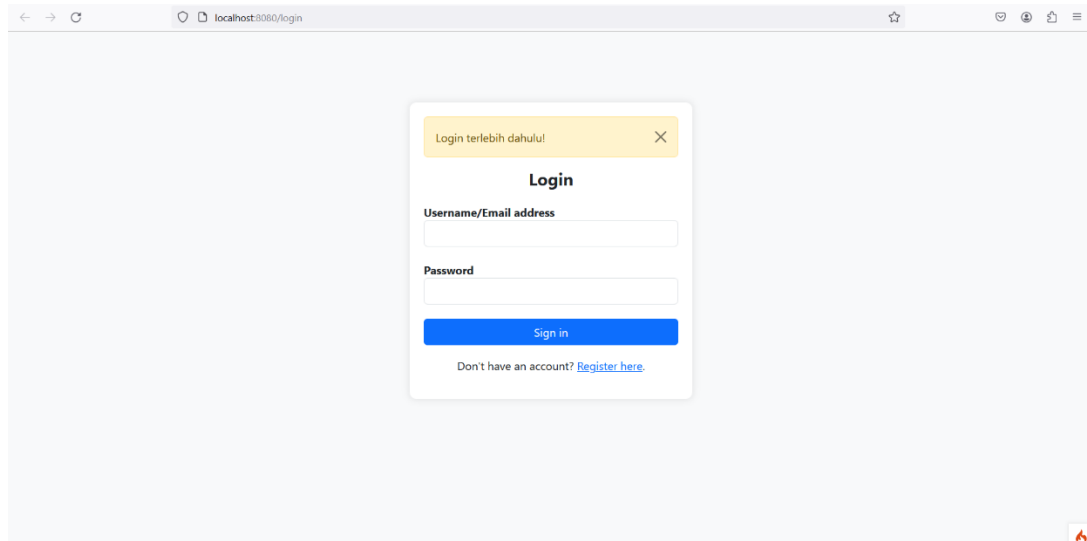
```

64 <button type="button" class="btn-close" data-bs-
65 dismiss="alert" aria-label="Close"></button>
66 </div>
67 <?php endif; ?>
68 <h5 class="card-title">Login</h5>
69 <form action="<?= base_url('/login') ?>" method="post">
70 <div class="form-group">
    <label for="email">Username/Email address</label>
71 <input type="text" class="form-control" id="email"
72 name="username" value="<?= old('username') ?>">
73 </div>
74 <div class="form-group">
    <label for="password">Password</label>
75 <input type="password" class="form-control"
76 id="password" name="password">
    </div>
77 <button type="submit" class="btn btn-primary">Sign
    in</button>
78 <p class="mt-3">Don't have an account? <a href="<?=
79 base_url('register') ?>">Register here</a>.</p>
80 </form>
81 </div>
    </div>
    <script
82 src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
    alpha3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
83 ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NT
84 KfAdVQSZe" crossorigin="anonymous">
85 </script>

```

	<pre> </body> </html> </pre>
--	--

Y. Output Program *login.php*



Gambar 35. Screenshot Hasil Jawaban Modul 6 *login.php*

Z. Pembahasan *login.php*

Pada baris [1] – [55], syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

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

```
    <title>Login</title>
```

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+d
N9+nJOZ" crossorigin="anonymous">

<style>

    body {

        background-color: #f8f9fa;

    }

    .card-view {

        max-width: 400px;

        margin: auto;

        margin-top: 100px;

        padding: 20px;

        border-radius: 10px;

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

        background-color: #fff;

    }

    .card-view .card-title {

        font-size: 24px;

        font-weight: bold;

        text-align: center;

```



```

        margin-bottom: 20px;
    }

    .card-view form {
        margin-bottom: 0;
    }

    .card-view .form-group {
        margin-bottom: 20px;
    }

    .card-view .form-group label {
        font-weight: bold;
    }

    .card-view .btn-primary {
        width: 100%;
    }

    .card-view .mt-3 {
        text-align: center;
    }

</style>

</head>

```

pada baris-baris tersebut merupakan kode HTML yang diberikan digunakan untuk membuat halaman login yang sederhana namun efektif dengan bantuan Bootstrap, sebuah framework CSS populer. Struktur dasar HTML termasuk deklarasi `<!DOCTYPE html>` untuk menunjukkan bahwa dokumen ini adalah HTML5, elemen

`<html lang="en">` untuk menetapkan bahasa utama dokumen, serta elemen meta untuk mengatur karakter encoding ke UTF-8, kompatibilitas dengan Internet Explorer, dan skala tampilan yang responsif. Judul halaman ditetapkan dengan `<title>Login</title>` dan CSS Bootstrap diimpor dari CDN untuk menggunakan gaya dan komponen yang sudah ada. Gaya kustom CSS diterapkan pada body untuk mengatur warna latar belakang (`#f8f9fa`), dan pada elemen div dengan kelas `.card-view` untuk menampilkan kartu login yang terpusat dengan lebar maksimum 400px, margin otomatis, padding, border-radius, box-shadow, dan warna latar belakang putih. Gaya tambahan untuk elemen di dalam `.card-view` termasuk pengaturan judul kartu, form, form-group, label, tombol utama, dan elemen dengan kelas `.mt-3`. Di dalam `<body>`, elemen `<div class="container">` digunakan untuk membungkus konten, sementara elemen `.card-view` menampilkan kartu login dengan pesan flash jika ada, judul "Login", dan form login yang mencakup input untuk username/email, password, tombol "Sign in", dan tautan ke halaman pendaftaran. Selanjutnya, tag `<script>` menghubungkan ke file JavaScript Bootstrap dari CDN, yang diperlukan untuk menjalankan komponen dinamis Bootstrap seperti alert yang dapat ditutup.

Pada baris [57] – [80], syntax

```
<body>

    <div class="container">

        <div class="card-view">

            <?php          if          (!empty(session()-
>getFlashdata('pesan'))): ?>

                <div class="alert alert-warning alert-
dismissible fade show" role="alert">

                    <?= session()->getFlashdata('pesan');

                ?>
```

```

        <button    type="button"    class="btn-
close"           data-bs-dismiss="alert"           aria-
label="Close"></button>

    </div>

    <?php endif; ?>

    <h5 class="card-title">Login</h5>

    <form    action="<? =    base_url('/login')    ?>"
method="post">

        <div class="form-group">

            <label        for="email">Username/Email
address</label>

            <input        type="text"        class="form-
control"        id="email"        name="username"        value="<? =
old('username') ?>">

        </div>

        <div class="form-group">

            <label
for="password">Password</label>

            <input    type="password"    class="form-
control" id="password" name="password">

        </div>

        <button    type="submit"    class="btn    btn-
primary">Sign in</button>

        <p class="mt-3">Don't have an account? <a
href="<? = base_url('register') ?>">Register here</a>.</p>

```

```

        </form>

    </div>

</div>

```

pada baris-baris tersebut merupakan bagian dari halaman login yang menggunakan PHP dan Bootstrap untuk memberikan tampilan yang bersih dan responsif serta menangani masukan pengguna secara efisien. Struktur utamanya dimulai dengan tag `<body>` yang membungkus seluruh konten halaman, diikuti oleh elemen `<div class="container">` yang merupakan container Bootstrap untuk memberikan padding dan margin otomatis, serta elemen `<div class="card-view">` yang digunakan untuk menampilkan formulir login dengan gaya khusus yang telah didefinisikan. Di dalam `card-view`, jika ada pesan flash yang disimpan dalam sesi, akan ditampilkan elemen `<div class="alert alert-warning alert-dismissible fade show" role="alert">` yang menampilkan pesan flash sebagai alert Bootstrap dengan opsi untuk menutup alert menggunakan tombol `<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>`. Judul formulir ditampilkan menggunakan `<h5 class="card-title">Login</h5>` yang memberikan tampilan tebal dan terpusat pada judul "Login". Formulir login dimulai dengan `<form action="<?= base_url('/login') ?>" method="post">` yang mengirimkan data login menggunakan metode POST ke endpoint `/login`. Di dalam formulir, terdapat elemen `<div class="form-group">` yang mengelompokkan label dan input untuk username/email dan password dengan gaya Bootstrap. Input username/email diwakili oleh `<input type="text" class="form-control" id="email" name="username" value="<?= old('username') ?>">` dan input password diwakili oleh `<input type="password" class="form-control" id="password" name="password">`. Formulir ditutup dengan tombol `<button type="submit" class="btn btn-primary">Sign in</button>` yang mengirimkan formulir, serta paragraf `<p class="mt-3">Don't have an account? <a href="<?= base_url('register') ?>">Register here</p>` yang memberikan tautan ke halaman pendaftaran bagi pengguna yang belum memiliki akun.

Pada baris [81] – [85], syntax

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKf
AdvQSZe" crossorigin="anonymous">

        </script>

</body>

</html>
```

pada baris-baris tersebut merupakan bagian akhir dari sebuah dokumen HTML. Penggunaan tag `<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdvQSZe" crossorigin="anonymous"></script>` memuat file JavaScript dari Bootstrap versi 5.3.0-alpha3 dari jsDelivr, sebuah CDN, yang berisi semua komponen JavaScript Bootstrap serta dependensi Popper.js yang diperlukan untuk beberapa komponen seperti dropdowns, tooltips, dan popovers. Atribut `integrity` digunakan untuk memastikan bahwa file yang dimuat belum dimodifikasi dan sesuai dengan hash yang diberikan, melindungi terhadap serangan berbahaya. Atribut `crossorigin="anonymous"` mengontrol pengaturan CORS untuk melakukan permintaan sumber daya tanpa mengirimkan kredensial ke server pihak ketiga. Selanjutnya pada Kode tersebut diakhiri dengan tag penutup `</body>` dan `</html>` yang menandai akhir dari konten utama dan dokumen HTML.

AA. Source Code *register.php*

Table 38. Source Code Modul 6 *register.php*

1	<!DOCTYPE html>
---	-----------------

2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
6	<meta name="viewport" content="width=device-
	width, initial-scale=1.0">
7	<title>Register</title>
8	<style>
9	body {
10	font-family: Arial, sans-serif;
11	background-color: #f8f9fa;
12	margin: 0;
13	padding: 0;
14	display: flex;
15	justify-content: center;
16	align-items: center;
17	height: 100vh;
18	}
19	
20	.card {
21	width: 500px;
22	padding: 30px;
23	background-color: #fff;
24	border-radius: 10px;
25	box-shadow: 0px 0px 10px 0px
	rgba(0,0,0,0.1);
26	}
27	
28	.form-group {

29	margin-bottom: 30px;
30	}
31	
32	.form-control {
33	width: 100%;
34	padding: 10px;
35	border: 1px solid #ced4da;
36	border-radius: 5px;
37	transition: border-color 0.3s ease;
38	font-size: 16px;
39	}
40	
41	.form-control:focus {
42	border-color: #007bff;
43	}
44	
45	.form-label {
46	color: #000;
47	transition: all 0.3s ease;
48	margin-bottom: 5px;
49	display: block;
50	}
51	
52	.form-control:focus ~ .form-label, .form-
	control:not(:placeholder-shown).form-
	control:not(:focus) ~ .form-label {
53	top: -10px;
54	font-size: 12px;
55	color: #007bff;
56	}

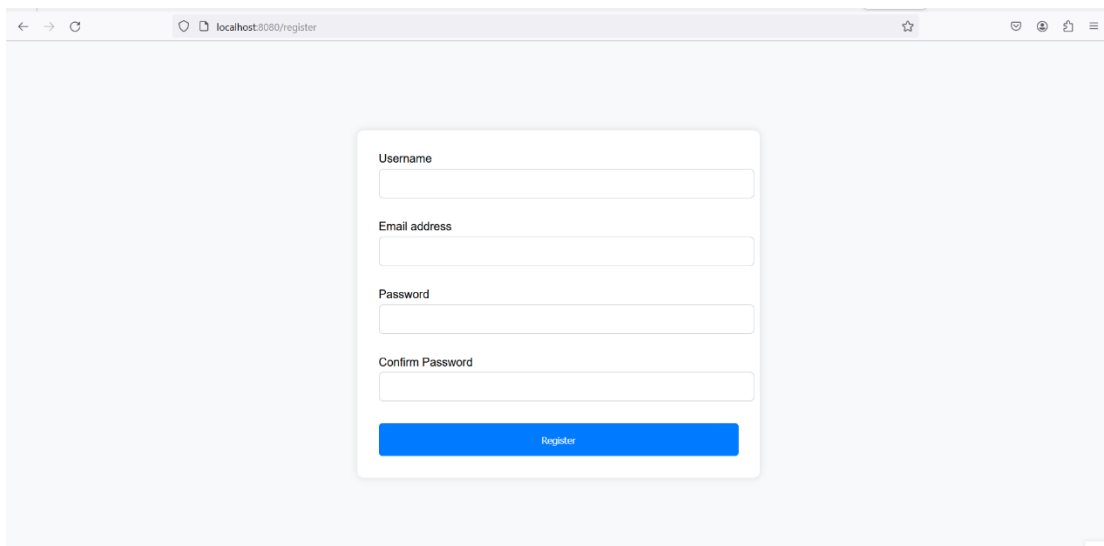
57	
58	.btn-primary {
59	background-color: #007bff;
60	border: none;
61	border-radius: 5px;
62	width: 100%;
63	padding: 15px;
64	color: #fff;
65	cursor: pointer;
66	transition: background-color 0.3s ease;
67	}
68	
69	.btn-primary:hover {
70	background-color: #0056b3;
71	}
72	
73	.alert {
74	border-radius: 5px;
75	padding: 15px;
76	margin-bottom: 30px;
77	}
78	
79	.alert-success {
80	background-color: #d4edda;
81	color: #155724;
82	border-color: #c3e6cb;
83	}
84	
85	.alert-danger {
86	background-color: #f8d7da;

87	color: #721c24;
88	border-color: #f5c6cb;
89	}
90	</style>
91	</head>
92	<body>
93	<div class="card">
94	<?php if (session()-
	>getFlashdata('success')) : ?>
95	<div class="alert alert-success"><?=
	session()->getFlashdata('success') ?></div>
96	<?php endif; ?>
97	
98	<?php if (session()-
	>getFlashdata('errors')) : ?>
99	<div class="alert alert-danger">
	<?php foreach (session()-
100	>getFlashdata('errors') as \$error) : ?>
	<p><?= \$error ?></p>
101	<?php endforeach; ?>
102	</div>
103	<?php endif; ?>
104	
105	<form action="<?= base_url('/register') ?>"
106	method="post">
	<div class="form-group">
107	<label class="form-label"
108	for="username">Username</label>
109	

	<code><input id="username" class="form-control" name="username" value="<?= 110 ?>" required /></code>
111	<code></div></code>
112	
113	<code><div class="form-group"></code>
	<code><label class="form-label"</code>
114	<code>for="email">Email address</label></code>
	<code><input id="email" class="form-</code>
115	<code>control" name="email" value="<?= 116 required /></code>
117	<code></div></code>
118	
	<code><div class="form-group"></code>
119	<code><label class="form-label"</code>
	<code>for="password">Password</label></code>
120	<code><input type="password"</code>
121	<code>id="password" name="password" class="form-control"</code>
122	<code>required /></code>
123	<code></div></code>
124	<code><div class="form-group"></code>
	<code><label class="form-label"</code>
	<code>for="confirm_password">Confirm Password</label></code>
125	<code><input type="password"</code>
126	<code>id="confirm_password" name="confirm_password"</code>
127	<code>class="form-control" required /></code>
	<code></div></code>
128	
129	

130	<code><button type="submit" class="btn btn-</code>
131	<code>primary">Register</button></code>
	<code></form></code>
	<code></div></code>
	<code></body></code>
	<code></html></code>

BB. Ouput Program *register.php*



Gambar 36. Screenshot Hasil Jawaban Modul 6 *register.php*

CC. Pembahasan *register.php*

Pada baris [1] – [91], syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```

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

    <title>Register</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f8f9fa;

            margin: 0;

            padding: 0;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

        }

        .card {

            width: 500px;

            padding: 30px;

            background-color: #fff;

            border-radius: 10px;

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

        }

        .form-group {

```

```

        margin-bottom: 30px;
    }

    .form-control {
        width: 100%;
        padding: 10px;
        border: 1px solid #ced4da;
        border-radius: 5px;
        transition: border-color 0.3s ease;
        font-size: 16px;
    }

    .form-control:focus {
        border-color: #007bff;
    }

    .form-label {
        color: #000;
        transition: all 0.3s ease;
        margin-bottom: 5px;
        display: block;
    }

    .form-control:focus ~ .form-label,
    .form-control:not(:placeholder-shown) .form-label:not(:focus)
    ~ .form-label {

```

```

    top: -10px;

    font-size: 12px;

    color: #007bff;
}

.btn-primary {
    background-color: #007bff;

    border: none;

    border-radius: 5px;

    width: 100%;

    padding: 15px;

    color: #fff;

    cursor: pointer;

    transition: background-color 0.3s ease;
}

.btn-primary:hover {
    background-color: #0056b3;
}

.alert {
    border-radius: 5px;

    padding: 15px;

    margin-bottom: 30px;
}

```

```

        .alert-success {
            background-color: #d4edda;

            color: #155724;

            border-color: #c3e6cb;
        }

        .alert-danger {
            background-color: #f8d7da;

            color: #721c24;

            border-color: #f5c6cb;
        }
    </style>
</head>

```

pada baris-baris tersebut merupakan bagian HTML dimulai dengan deklarasi dokumen HTML5 dan menetapkan bahasa halaman menjadi bahasa Inggris. Metadata seperti pengaturan karakter encoding UTF-8, kompatibilitas mode rendering untuk Internet Explorer, dan viewport yang responsif juga ditetapkan. Tag <title> menentukan judul halaman yang akan ditampilkan di tab browser. Terdapat juga penggunaan bagian CSS mendefinisikan tampilan halaman, dimulai dari properti global untuk seluruh body seperti jenis font, warna latar belakang, dan display flex untuk penataan elemen. .card mengatur kotak formulir dengan properti seperti lebar, padding, latar belakang, border-radius, dan box-shadow. Pengaturan juga diberikan untuk elemen-elemen seperti .form-group, .form-control untuk input, .form-label untuk label, .btn-primary untuk tombol "Register", serta .alert, .alert-success, dan .alert-danger untuk pesan peringatan dengan warna dan gaya yang sesuai.

Pada baris [92] – [131], syntax

```

<body>

    <div class="card">

        <?php if (session()->getFlashdata('success')) : ?>

            <div class="alert alert-success"><?=
session()->getFlashdata('success') ?></div>

        <?php endif; ?>

        <?php if (session()->getFlashdata('errors')) : ?>

            <div class="alert alert-danger">

                <?php foreach (session()-
>getFlashdata('errors') as $error) : ?>

                    <p><?= $error ?></p>

                <?php endforeach; ?>

            </div>

        <?php endif; ?>

        <form action="<?= base_url('/register') ?>"
method="post">

            <div class="form-group">

                <label class="form-label"
for="username">Username</label>

                <input id="username" class="form-control"
name="username" value="<?= old('username') ?>" required />

            </div>

            <div class="form-group">

```



```

        <label                                class="form-label"
for="email">Email address</label>

        <input    id="email"    class="form-control"
name="email" value="<?= old('email') ?>" required />

    </div>

    <div class="form-group">

        <label                                class="form-label"
for="password">Password</label>

        <input    type="password"    id="password"
name="password" class="form-control" required />

    </div>

    <div class="form-group">

        <label                                class="form-label"
for="confirm_password">Confirm Password</label>

        <input                                type="password"
id="confirm_password"                        name="confirm_password"
class="form-control" required />

    </div>

    <button    type="submit"    class="btn    btn-
primary">Register</button>

</form>

</div>

</body>

</html>

```

pada baris-baris tersebut merupakan bagian penting dari halaman HTML untuk menampilkan formulir pendaftaran. Terdapat juga penggunaan PHP untuk menangani pesan-pesan sukses atau error setelah pengguna mengirimkan formulir. Misalnya, jika ada pesan 'success', akan ditampilkan dalam sebuah div dengan kelas 'alert alert-success'. Begitu juga dengan pesan 'errors', di mana setiap kesalahan ditampilkan sebagai paragraf terpisah di dalam div dengan kelas 'alert alert-danger'. Semua pesan ini disimpan dalam session menggunakan framework PHP seperti CodeIgniter. Formulir pendaftaran tersebut terdiri dari empat bagian utama: input untuk username, email, password, dan konfirmasi password. Setiap input dilengkapi dengan label yang jelas, yaitu "Username" dan "Email address" untuk memudahkan pengguna saat mengisi formulir. Input untuk password diatur dengan tipe 'password' untuk mengamankan masukan pengguna. Selanjutnya terdapat tombol "Register" digunakan untuk mengirimkan data formulir ke endpoint '/register' menggunakan metode POST, yang berarti data akan dikirimkan secara aman ke server untuk diproses lebih lanjut.

DD. Source Code welcome_mesaage.php

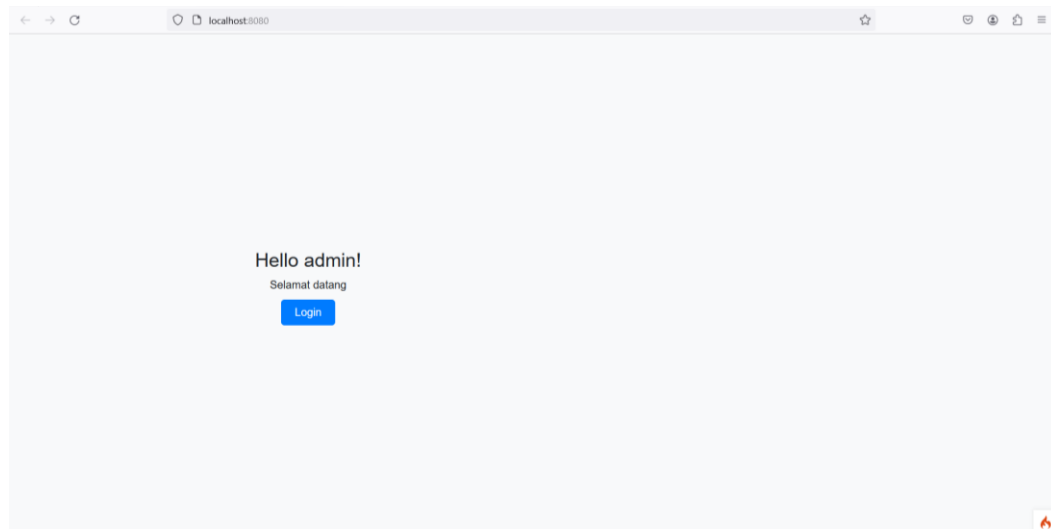
Table 39.Source Code Modul 6 *welcome_message.php*

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<title>PRAKTIKUM 6</title>
7	<meta name="description" content="The small framework with powerful features">
8	<meta name="viewport" content="width=device- width, initial-scale=1.0">
9	<link rel="shortcut icon" type="image/png" href="/favicon.ico">
10	

	<pre> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0- alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384- KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p 5ILy+dN9+nJOZ" crossorigin="anonymous"> </pre>
11	
12	
13	<pre> <style> </pre>
14	<pre> body { </pre>
15	<pre> font-family: Arial, sans-serif; </pre>
16	<pre> background-color: #f8f9fa; </pre>
17	<pre> margin: 0; </pre>
18	<pre> padding: 0; </pre>
19	<pre> display: flex; </pre>
20	<pre> justify-content: center; </pre>
21	<pre> align-items: center; </pre>
22	<pre> height: 100vh; </pre>
23	<pre> } </pre>
24	
25	<pre> .container { </pre>
26	<pre> text-align: center; </pre>
27	<pre> } </pre>
28	
29	<pre> .btn-primary { </pre>
30	<pre> background-color: #007bff; </pre>
31	<pre> color: #fff; </pre>
32	<pre> padding: 10px 20px; </pre>
33	<pre> font-size: 16px; </pre>
34	<pre> border: none; </pre>
35	<pre> border-radius: 5px; </pre>

36	cursor: pointer;
37	text-decoration: none;
38	transition: background-color 0.3s ease;
39	}
40	
41	.btn-primary:hover {
42	background-color: #0056b3;
43	}
44	</style>
45	</head>
46	
47	<body>
48	<div class="container">
49	<div class="row">
50	<div class="col-md-6">
	<h3 class="h3">Hello <?php if
	(session()->get('logged_in') === TRUE) {
51	echo session()-
	>get('username');
52	} else {
53	echo "Guest";
54	} ?>!/h3>
55	<p>Selamat datang</p>
56	<a href="<?= base_url('login')
	?>" class="btn-primary">Login
57	</div>
58	</div>
59	</div>
60	</body>
61	</html>

EE. Output Program welcome_mesaage.php



Gambar 37. Screenshot Hasil Jawaban Modul 6 *welcome_message.php*

FF. Pembahasan welcome_mesaage.php

Pada baris [1] – [44], syntax

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>PRAKTIKUM 6</title>

    <meta name="description" content="The small framework
with powerful features">

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

    <link rel="shortcut icon" type="image/png"
href="/favicon.ico">
```

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css"          rel="stylesheet"
integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+d
N9+nJOZ" crossorigin="anonymous">
```

```
<style>
```

```
body {

    font-family: Arial, sans-serif;

    background-color: #f8f9fa;

    margin: 0;

    padding: 0;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

}

.container {

    text-align: center;

}

.btn-primary {

    background-color: #007bff;

    color: #fff;
```

```

        padding: 10px 20px;

        font-size: 16px;

        border: none;

        border-radius: 5px;

        cursor: pointer;

        text-decoration: none;

        transition: background-color 0.3s ease;
    }

    .btn-primary:hover {

        background-color: #0056b3;

    }

</style>

</head>

```

pada baris-baris tersebut merupakan kode untuk sebuah halaman HTML yang mencakup deklarasi dokumen dan meta tags untuk mengatur karakter set, judul halaman, deskripsi singkat, serta pengaturan viewport untuk responsifitas. Selain itu, terdapat link untuk favicon dan penggunaan Bootstrap melalui CDN untuk mengambil file CSS Bootstrap versi 5.3.0-alpha3. Bagian CSS internalnya menentukan gaya halaman dengan font Arial atau sans-serif, latar belakang abu-abu muda, dan konten yang terpusat secara vertikal dan horizontal di tengah halaman. Class `.container` digunakan untuk meratakan teks ke tengah, sementara tombol `.btn-primary` diberi warna latar belakang biru dengan transisi warna saat digulirkan untuk efek visual yang lebih halus.

Pada baris [46] – [61], syntax

```

<body>

    <div class="container">

        <div class="row">

            <div class="col-md-6">

                <h3 class="h3">Hello <?php if (session()-
>get('logged_in') === TRUE) {

                    echo session()->get('username');

                } else {

                    echo "Guest";

                } ?>!/h3>

                <p>Selamat datang</p>

                <a href="<?= base_url('login') ?>"
class="btn-primary">Login</a>

            </div>

        </div>

    </div>

</body>

</html>

```

pada baris-baris tersebut merupakan halaman HTML yang menampilkan pesan sambutan dinamis berdasarkan status login pengguna dan menyediakan tombol untuk login. Struktur dasar HTML menggunakan elemen .container dari Bootstrap, dengan satu baris dan satu kolom besar di dalamnya untuk menampilkan konten. Di dalam elemen <h3>, terdapat kondisi PHP untuk menampilkan nama pengguna jika sudah login, atau menampilkan "Guest" jika belum. Elemen <p> menampilkan pesan

"Selamat datang" sebagai tambahan informasi setelah pesan sambutan. Tombol login (<a>) dengan class btn-primary mengarahkan pengguna ke halaman login.

GG. Source Code BukuModel.php

Table 40. Source Code Modul 6 BukuModelphp

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class BukuModel extends Model
8	{
9	protected \$DBGroup = 'default';
10	protected \$table = 'buku';
11	protected \$primaryKey = 'id';
12	protected \$useAutoIncrement = true;
13	protected \$returnType = 'array';
14	protected \$useSoftDeletes = false;
15	protected \$protectFields = true;
16	protected \$allowedFields = [
17	'judul',
18	'penulis',
19	'penerbit',
20	'tahun_terbit'
21];
22	
23	// Dates
24	protected \$useTimestamps = false;
25	protected \$dateFormat = 'datetime';

26	protected \$createdField = 'created_at';
27	protected \$updatedField = 'updated_at';
28	protected \$deletedField = 'deleted_at';
29	
30	// Validation
31	protected \$validationRules = [];
32	protected \$validationMessages = [];
33	protected \$skipValidation = false;
34	protected \$cleanValidationRules = true;
35	
36	// Callbacks
37	protected \$allowCallbacks = true;
38	protected \$beforeInsert = [];
39	protected \$afterInsert = [];
40	protected \$beforeUpdate = [];
41	protected \$afterUpdate = [];
42	protected \$beforeFind = [];
43	protected \$afterFind = [];
44	protected \$beforeDelete = [];
45	protected \$afterDelete = [];
46	}

HH. Pembahasan BukuModel.php

Pada baris [1] [2] [3] [4] [5], syntax

```
<?php
```

```
namespace App\Models;
```

```
use CodeIgniter\Model;
```

pada baris-baris tersebut mendefinisikan namespace App\Models dan mengimpor kelas Model dari framework CodeIgniter.

Pada baris [7], syntax `class BukuModel extends Model` baris tersebut mendefinisikan kelas BukuModel merupakan turunan dari kelas Model dari CodeIgniter.

Pada baris [8] -[21], syntax

```
{  
  
    protected $DBGroup          = 'default';  
  
    protected $table            = 'buku';  
  
    protected $primaryKey       = 'id';  
  
    protected $useAutoIncrement = true;  
  
    protected $returnType       = 'array';  
  
    protected $useSoftDeletes    = false;  
  
    protected $protectFields     = true;  
  
    protected $allowedFields     = [  
  
        'judul',  
  
        'penulis',  
  
        'penerbit',  
  
        'tahun_terbit'  
  
    ];
```

pada baris-baris tersebut digunakan untuk mengkonfigurasi perilaku model dalam berinteraksi dengan tabel database buku. Hal tersebut termasuk pengaturan seperti koneksi basis data (\$DBGroup), nama tabel (\$table), primary key (\$primaryKey),

penggunaan auto increment untuk primary key (\$useAutoIncrement), tipe data yang dikembalikan (\$returnType), dan sebagainya.

Pada baris [24] – [28], syntax

```
protected $useTimestamps = false;

protected $dateFormat      = 'datetime';
protected $createdField    = 'created_at';
protected $updatedField    = 'updated_at';
protected $deletedField    = 'deleted_at';
```

pada baris-baris tersebut merupakan bagian yang mengatur apakah model akan menggunakan timestamps, misalnya, created_at dan updated_at. Penggunaan \$useTimestamps akan mengontrol penggunaan timestamps, sedangkan \$dateFormat menentukan format tanggal yang digunakan.

Pada baris [31] – [34], syntax

```
protected $validationRules      = [];

protected $validationMessages   = [];
protected $skipValidation       = false;
protected $cleanValidationRules = true;
```

pada abris-baris tersebut merupakan properti-properti yang digunakan untuk mengatur aturan validasi data. \$validationRules berisi aturan-aturan validasi, \$validationMessages berisi pesan-pesan kesalahan yang terkait dengan validasi, dan \$skipValidation menentukan apakah validasi harus dilewati atau tidak.

Pada baris [37] – [46], syntax

```
protected $allowCallbacks = true;

protected $beforeInsert   = [];
```

```

        protected $afterInsert      = [];

        protected $beforeUpdate     = [];

        protected $afterUpdate      = [];

        protected $beforeFind       = [];

        protected $afterFind        = [];

        protected $beforeDelete     = [];

        protected $afterDelete      = [];

    }

```

Pada baris-baris tersebut merupakan properti-properti yang digunakan untuk mendefinisikan callback-callback yang akan dieksekusi sebelum atau sesudah operasi-insert, update, find (query), atau delete. Callback-callback ini memungkinkan untuk mengeksekusi kode tambahan sebelum atau sesudah melakukan operasi terkait pada data.

II. Source Code UserModel.php

Table 41. Source Code Modul 6 *UserModel.php*

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class UserModel extends Model
8	{
9	protected \$DBGroup = 'default';
10	protected \$table = 'user';
11	protected \$primaryKey = 'id';

```

12     protected $useAutoIncrement = true;
13     protected $returnType       = 'array';
14     protected $useSoftDeletes   = false;
15     protected $protectFields    = true;
16     protected $allowedFields    = ['username', 'email',
    'password'];
17
18     protected $useTimestamps = false;
19     protected $dateFormat    = 'datetime';
20     protected $createdField   = 'created_at';
21     protected $updatedField   = 'updated_at';
22     protected $deletedField   = 'deleted_at';
23
24     protected $validationRules = [
25         'username' =>
    'required|min_length[3]|max_length[20]|is_unique[user.username]',
26         'email'    =>
    'required|valid_email|is_unique[user.email]',
27         'password' => 'required|min_length[6]',
28     ];
29     protected $validationMessages = [
30         'username' => [
31             'required'    => 'Username is required',
32             'min_length' => 'Username must be at least 3
characters long',
33             'max_length' => 'Username cannot exceed 20
characters',
34             'is_unique'  => 'Username already exists',
35         ],
36         'email' => [

```

37	'required' => 'Email is required',
38	'valid_email' => 'You must provide a valid email
	address',
39	'is_unique' => 'Email already exists',
40],
41	'password' => [
42	'required' => 'Password is required',
43	'min_length' => 'Password must be at least 6
	characters long',
44],
45];
46	protected \$skipValidation = false;
47	protected \$cleanValidationRules = true;
48	
49	protected \$allowCallbacks = true;
50	protected \$beforeInsert = [];
51	protected \$afterInsert = [];
52	protected \$beforeUpdate = [];
53	protected \$afterUpdate = [];
54	protected \$beforeFind = [];
55	protected \$afterFind = [];
56	protected \$beforeDelete = [];
57	protected \$afterDelete = [];
58	}

JJ. Pembahasan UserModel.php

Pada baris [1] – [5], syntax

```
<?php
```

```
namespace App\Models;
```

```
use CodeIgniter\Model;
```

pada baris-baris tersebut men deklarasikan namespace App\Models. Hal tersebut menandakan bahwa kelas UserModel berada di direktori App\Models dalam struktur proyek CodeIgniter.

Pada baris [7], syntax `class UserModel extends Model` baris tersebut merupakan kelas UserModel yang mewarisi semua fitur dan metode dari kelas Model dari CodeIgniter.

Pada baris [8] – [16], syntax

```
{  
  
    protected $DBGroup          = 'default';  
  
    protected $table            = 'user';  
  
    protected $primaryKey       = 'id';  
  
    protected $useAutoIncrement = true;  
  
    protected $returnType       = 'array';  
  
    protected $useSoftDeletes   = false;  
  
    protected $protectFields    = true;  
  
    protected $allowedFields     = ['username', 'email',  
    'password'];
```

pada baris-baris tersebut merupakan konfigurasi utama untuk sebuah model dalam framework PHP seperti CodeIgniter. Properti \$DBGroup menentukan grup koneksi basis data yang digunakan, dengan 'default' menunjukkan penggunaan koneksi standar. \$table menentukan nama tabel dalam basis data yang digunakan oleh model ini, dalam

kasus ini adalah 'user'. \$primaryKey menentukan kolom yang berfungsi sebagai primary key dalam tabel ('id'), sementara \$useAutoIncrement mengatur apakah primary key menggunakan increment otomatis. \$returnType menentukan jenis data yang dikembalikan oleh model, dalam hal ini 'array'. \$useSoftDeletes mengatur apakah model mendukung penghapusan lembut (soft deletes), sedangkan \$protectFields mengontrol apakah field-field tertentu dilindungi dari asignasi secara massal. Selanjutnya, \$allowedFields menentukan field-field yang diizinkan untuk asignasi secara massal, yang dalam kode tersebut mencakup 'username', 'email', dan 'password'.

Pada baris [18] – [22], syntax

```
protected $useTimestamps = false;

protected $dateFormat    = 'datetime';

protected $createdField  = 'created_at';

protected $updatedField  = 'updated_at';

protected $deletedField  = 'deleted_at';
```

pada abris-baris tersebut bertugas untuk mengatur pengelolaan timestamps dalam sebuah model dalam framework PHP seperti CodeIgniter. Penggunaan \$useTimestamps untuk mengontrol apakah timestamps seperti created_at dan updated_at digunakan pada kode tersbeut diatur sebagai false. \$dateFormat menentukan format tanggal yang digunakan untuk timestamps, dengan nilai 'datetime'. Selanjutnya, \$createdField, \$updatedField, dan \$deletedField menentukan nama kolom dalam tabel basis data yang digunakan untuk masing-masing timestamps, yaitu 'created_at', 'updated_at', dan 'deleted_at'.

Pada baris [24] – [47], syntax

```
protected $validationRules = [
```

```

        'username' =>
'required|min_length[3]|max_length[20]|is_unique[user.use
rname] ',

        'email' =>
'required|valid_email|is_unique[user.email] ',

        'password' => 'required|min_length[6] ',

];

protected $validationMessages = [

    'username' => [

        'required' => 'Username is required',

        'min_length' => 'Username must be at least 3
characters long',

        'max_length' => 'Username cannot exceed 20
characters',

        'is_unique' => 'Username already exists',

    ],

    'email' => [

        'required' => 'Email is required',

        'valid_email' => 'You must provide a valid
email address',

        'is_unique' => 'Email already exists',

    ],

    'password' => [

        'required' => 'Password is required',

```

```

        'min_length' => 'Password must be at least 6
characters long',

    ],

];

protected $skipValidation      = false;

protected $cleanValidationRules = true;

```

pada baris-baris tersebut digunakan untuk mengatur validasi data dalam sebuah model dalam framework PHP seperti CodeIgniter. Penggunaan `$validationRules` untuk menentukan aturan validasi untuk masing-masing field seperti username, email, dan password, termasuk keharusan (required), panjang minimal (`min_length`), panjang maksimal (`max_length`), dan unik (`is_unique` untuk memastikan nilai unik dalam basis data). Lalu penggunaan `$validationMessages` untuk menentukan pesan kesalahan yang ditampilkan jika validasi gagal untuk setiap aturan. `$skipValidation` mengontrol apakah validasi akan dilewati atau tidak (diatur sebagai `false` untuk melakukan validasi). Serta `$cleanValidationRules` mengontrol apakah aturan validasi harus dibersihkan sebelum digunakan.

Pada baris [49] – [58], syntax

```

protected $allowCallbacks = true;

protected $beforeInsert    = [];

protected $afterInsert     = [];

protected $beforeUpdate    = [];

protected $afterUpdate     = [];

protected $beforeFind      = [];

protected $afterFind       = [];

protected $beforeDelete    = [];

```

```

        protected $afterDelete    = [];
    }

```

pada baris-baris tersebut digunakan untuk mengatur penggunaan callback dalam sebuah model dalam framework PHP seperti CodeIgniter. Penggunaan \$allowCallbacks untuk menentukan apakah fungsi callback seperti sebelum insert atau setelah delete diizinkan, dimana pada kode tersebut diatur menjadi true. Penggunaan properti-properti seperti \$beforeInsert, \$afterInsert, \$beforeUpdate, \$afterUpdate, \$beforeFind, \$afterFind, \$beforeDelete, dan \$afterDelete digunakan untuk menentukan tindakan tertentu yang akan dijalankan sebelum atau setelah operasi dasar pada data, seperti penyisipan, pembaruan, pencarian, dan penghapusan.

KK. Source Code UserSeeder.php

Table 42. Source Code Modul 6 *UserSeeder.php*

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class UserSeeder extends Seeder
8	{
9	public function run()
10	{
11	\$this->db->table('user')->insert([
12	'username' => 'admin',
13	'email' => 'admin@mail.com',
14	'password' => password_hash('12345678',
	PASSWORD_DEFAULT),
15]);

16	}
17	}

LL. Pembahasan *UserSeeder.php*

Pada baris [1] – [5], syntax

```
<?php
```

```
namespace App\Database\Seeds;
```

```
use CodeIgniter\Database\Seeder;
```

pada baris-baris tersebut merupakan deklarasi namespace `App\Database\Seeds`, yang menunjukkan lokasi dari kelas `UserSeeder` di dalam struktur direktori proyek.

Pada abirs [7], syntax `class UserSeeder extends Seeder` baris tersebut merupakan kelas `UserSeeder` mewarisi semua fungsi dari kelas `Seeder` `CodeIgniter`, yang menyediakan metode untuk memasukkan data ke dalam basis data.

Pada baris [8] – [17], syntax

```
{
    public function run()
    {
        $this->db->table('user')->insert([
            'username' => 'admin',
            'email' => 'admin@mail.com',
            'password' => password_hash('12345678',
PASSWORD_DEFAULT),
        ]);
    }
}
```

```
}
```

pada baris-baris tersebut merupakan method `run()`, method khusus yang digunakan dalam proses seeding data pada basis data. Ketika method ini dieksekusi, kode yang terdapat di dalamnya akan menjalankan operasi untuk memasukkan data ke dalam tabel `user`. Pada kode tersebut, terdapat pemanggilan `$this->db->table('user')->insert([...])`, yang berfungsi untuk memasukkan sebuah array asosiatif ke dalam tabel `user`. Array tersebut berisi nilai untuk kolom-kolom seperti `'username'` dengan nilai `'admin'`, `'email'` dengan nilai `'admin@mail.com'`, dan `'password'` dengan hasil dari fungsi `password_hash('12345678', PASSWORD_DEFAULT)`. Fungsi `password_hash()` digunakan untuk mengamankan dan mengenkripsi password `'12345678'` menggunakan algoritma yang aman sebelum disimpan ke dalam basis data.

MM. Source Code 2024-06-12-202936_Buku

Table 43. Source Code Modul 6 2024-06-12-202936_Buku

1	<code><?php</code>
2	
3	<code>namespace App\Database\Migrations;</code>
4	
5	<code>use CodeIgniter\Database\Migration;</code>
6	
7	<code>class Buku extends Migration</code>
8	<code>{</code>
9	<code> public function up()</code>
10	<code> {</code>
11	<code> \$this->forge->addField([</code>
12	<code> 'id' => [</code>
13	<code> 'type' => 'INT',</code>
14	<code> 'constraint' => 5,</code>
15	<code> 'unsigned' => true,</code>
16	<code> 'auto_increment' => true</code>

```

17         ],
18         'judul' => [
19             'type' => 'VARCHAR',
20             'constraint' => 100,
21         ],
22         'penulis' => [
23             'type' => 'VARCHAR',
24             'constraint' => 100,
25         ],
26         'penerbit' => [
27             'type' => 'VARCHAR',
28             'constraint' => 100,
29         ],
30         'tahun_terbit' => [
31             'type' => 'YEAR',
32         ],
33     ]);
34
35     $this->forge->addKey('id', true);
36     $this->forge->createTable('buku');
37 }
38
39 public function down()
40 {
41     $this->forge->dropTable('buku', true);
42 }
43 }

```

NN. Pembahasan 2024-06-12-202936_Buku

Pada baris [1] – [5], syntax

```
?php
```

```
namespace App\Database\Migrations;
```

```
use CodeIgniter\Database\Migration;
```

pada baris-baris tersebut merupakan deklarasi namespace App\Database\Migrations, yang menunjukkan lokasi dari kelas Buku di dalam struktur direktori proyek.

Pada baris [7], syntax `class Buku extends Migration` baris tersebut merupakan kelas Buku mewarisi semua fungsi dari kelas Migration CodeIgniter, yang menyediakan metode untuk mengatur skema basis data.

Pada baris [8] – [37], syntax

```
{  
  
    public function up()  
    {  
  
        $this->forge->addField([  
  
            'id' => [  
  
                'type' => 'INT',  
  
                'constraint' => 5,  
  
                'unsigned' => true,  
  
                'auto_increment' => true  
  
            ],  
  
            'judul' => [  
  
                'type' => 'VARCHAR',
```



```

        'constraint' => 100,
    ],
    'penulis' => [
        'type' => 'VARCHAR',
        'constraint' => 100,
    ],
    'penerbit' => [
        'type' => 'VARCHAR',
        'constraint' => 100,
    ],
    'tahun_terbit' => [
        'type' => 'YEAR',
    ],
]);

$this->forge->addKey('id', true);

$this->forge->createTable('buku');

}

```

pada baris-baris tersebut merupakan method `up()`, sebuah method dalam kode migrasi yang bertanggung jawab untuk membuat struktur tabel baru dalam basis data. Pada kode tersebut, method ini digunakan untuk membuat tabel buku dengan menggunakan fungsi `$this->forge->addField([...])` untuk menambahkan definisi kolom ke dalam tabel. Di dalam method ini, terdapat definisi kolom seperti 'id' dengan tipe INT, panjang

maksimum 5 digit, dan memiliki atribut unsigned serta auto_increment untuk menghasilkan nilai yang otomatis bertambah. Selain itu, terdapat juga kolom 'judul', 'penulis', 'penerbit', dan 'tahun_terbit' dengan tipe VARCHAR dan YEAR yang masing-masing memiliki constraint panjang maksimum dan tipe data yang sesuai. Metode `$this->forge->addKey('id', true)` digunakan untuk menetapkan kolom 'id' sebagai primary key, sedangkan `$this->forge->createTable('buku')` digunakan untuk mengeksekusi pembuatan tabel 'buku' berdasarkan definisi kolom yang telah ditentukan sebelumnya.

Pada baris [39] – [41], syntax

```
public function down()
{
    $this->forge->dropTable('buku', true);
}
}
```

pada baris tersebut merupakan method `down()` yang digunakan dalam kode migrasi untuk melakukan rollback, khususnya untuk menghapus tabel yang telah dibuat sebelumnya. Dalam implementasinya, method ini menggunakan `$this->forge->dropTable('buku', true)` untuk menghapus tabel 'buku' dari basis data. Parameter kedua yang disetel sebagai `true` mengindikasikan agar SQL yang dihasilkan akan mencakup `IF EXISTS`, sehingga mengamankan proses penghapusan jika tabel tidak ada dalam basis data saat rollback dilakukan.

OO. Source Code 2024-06-12-202940_User

Table 44. Source Code Modul 6 2024-06-12-202940_User

1	<?php
2	

3	namespace App\Database\Migrations;
4	
5	use CodeIgniter\Database\Migration;
6	
7	class User extends Migration
8	{
9	public function up()
10	{
11	\$this->forge->addField([
12	'id'=> [
13	'type' => 'INT',
14	'constraint' => 5,
15	'unsigned' => true,
16	'auto_increment' => true
17],
18	'username' => [
19	'type' => 'VARCHAR',
20	'constraint' => 100,
21],
22	'email' => [
23	'type' => 'VARCHAR',
24	'constraint' => 100,
25],
26	'password' => [
27	'type' => 'TEXT',
28	
29],
30	'created_at' => [
31	'type' => 'DATETIME',
32],

33	'updated_at' => [
34	'type' => 'DATETIME',
35],
36	'deleted_at' => [
37	'type' => 'DATETIME',
38],
39]);
40	
41	\$this->forge->addKey('id', true);
42	\$this->forge->createTable('user', true);
43	}
44	
45	public function down()
46	{
47	\$this->forge->dropTable('user', true);
48	}
49	}

PP. Pembahasan 2024-06-12-202940_User

Pada baris [1] – [5], syntax

```
<?php
```

```
namespace App\Database\Migrations;
```

```
use CodeIgniter\Database\Migration;
```

pada baris tersebut digunakan untuk mendeklarasi namespace App\Database\Migrations, yang menunjukkan lokasi dari kelas User di dalam struktur direktori proyek.

Pada baris [7], syntax `class User extends Migration` baris tersebut merupakan kelas User yang mewarisi semua fungsi dari kelas Migration CodeIgniter, serta menyediakan metode untuk mengatur skema basis data.

Pada baris [8] – [43], syntax

```
{  
  
    public function up()  
    {  
  
        $this->forge->addField([  
  
            'id'=> [  
  
                'type' => 'INT',  
  
                'constraint' => 5,  
  
                'unsigned' => true,  
  
                'auto_increment' => true  
  
            ],  
  
            'username' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => 100,  
  
            ],  
  
            'email' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => 100,  
  
            ],  
  
        ],
```

```

        'password' => [
            'type' => 'TEXT',

        ],
        'created_at' => [
            'type' => 'DATETIME',
        ],
        'updated_at' => [
            'type' => 'DATETIME',
        ],
        'deleted_at' => [
            'type' => 'DATETIME',
        ],
    ]);

    $this->forge->addKey('id', true);

    $this->forge->createTable('user', true);
}

```

pada baris-baris tersebut merupakan method up() yang berisi kode untuk membuat tabel user dengan struktur dan properti kolom yang didefinisikan menggunakan \$this->forge->addField(...). \$this->forge->addField(...) menambahkan definisi kolom ke dalam tabel, di mana 'id' => [...], 'username' => [...], 'email' => [...], 'password' => [...], 'created_at' => [...], 'updated_at' => [...], dan 'deleted_at' => [...] mendefinisikan kolom-kolom tabel beserta tipe data, panjang maksimum, dan atribut lainnya seperti auto_increment untuk kolom id dan tipe data DATETIME untuk created_at,

updated_at, dan deleted_at. \$this->forge->addKey('id', true) menetapkan kolom id sebagai primary key. \$this->forge->createTable('user', true) membuat tabel dengan nama user berdasarkan definisi yang telah ditentukan, di mana parameter kedua true mengindikasikan untuk menambahkan IF NOT EXISTS pada perintah SQL yang dihasilkan.

Pada baris [45] – [49], syntax

```
public function down()
{
    $this->forge->dropTable('user', true);
}
}
```

pada baris-baris tersebut merupakan method down() yang digunakan untuk melakukan rollback, dalam hal ini menghapus tabel yang telah dibuat. \$this->forge->dropTable('user', true) menghapus tabel user dari basis data, di mana parameter kedua true mengindikasikan untuk menambahkan IF EXISTS pada perintah SQL yang dihasilkan.

QQ. Source Code buku.sql

Table 45.Source Code Modul 6 buku.sql

1	SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
2	START TRANSACTION;
3	SET time_zone = "+00:00";
4	
5	
6	CREATE TABLE `buku` (
7	`id` INT(5) UNSIGNED NOT NULL,
8	`judul` VARCHAR(100) NOT NULL,

9	<code>`penulis` VARCHAR(100) NOT NULL,</code>
10	<code>`penerbit` VARCHAR(100) NOT NULL,</code>
11	<code>`tahun_terbit` YEAR(4) NOT NULL</code>
12	<code>) ENGINE=INNODB DEFAULT CHARSET=utf8</code>
	<code>COLLATE=utf8_general_ci;</code>
13	
14	
15	<code>CREATE TABLE `crud` (</code>
16	<code> `id` INT(5) UNSIGNED NOT NULL,</code>
17	<code> `nama` VARCHAR(50) NOT NULL,</code>
18	<code> `nim` VARCHAR(13) NOT NULL,</code>
19	<code> `alamat` TEXT NOT NULL</code>
20	<code>) ENGINE=INNODB DEFAULT CHARSET=utf8</code>
	<code>COLLATE=utf8_general_ci;</code>
21	
22	<code>INSERT INTO `crud` (`id`, `nama`, `nim`, `alamat`)</code>
23	<code>VALUES</code>
24	<code>(1, 'syahna', '2210817120007', 'Banjarmasin'),</code>
25	<code>(4, 'puteri', '2210817120007', 'Rantau');</code>
26	
27	<code>CREATE TABLE `migrations` (</code>
28	<code> `id` BIGINT(20) UNSIGNED NOT NULL,</code>
29	<code> `version` VARCHAR(255) NOT NULL,</code>
30	<code> `class` VARCHAR(255) NOT NULL,</code>
31	<code> `group` VARCHAR(255) NOT NULL,</code>
32	<code> `namespace` VARCHAR(255) NOT NULL,</code>
33	<code> `time` INT(11) NOT NULL,</code>
34	<code> `batch` INT(11) UNSIGNED NOT NULL</code>
35	<code>) ENGINE=INNODB DEFAULT CHARSET=utf8</code>
	<code>COLLATE=utf8_general_ci;</code>

36	
37	INSERT INTO `migrations` (`id`, `version`, `class`,
	`group`, `namespace`, `time`, `batch`) VALUES
38	(1, '2024-06-12-080538',
	'App\\Database\\Migrations\\Crud', 'default', 'App',
	1684830849, 1),
39	(2, '2024-06-12-090624',
	'App\\Database\\Migrations\\User', 'default', 'App',
	1685438559, 2),
40	(3, '2024-06-12-015133',
	'App\\Database\\Migrations\\Buku', 'default', 'App',
	1686193775, 3),
41	(4, '2024-06-12-114209',
	'App\\Database\\Migrations\\User', 'default', 'App',
	1686193775, 3);
42	
43	CREATE TABLE `user` (
44	`id` INT(5) UNSIGNED NOT NULL,
45	`username` VARCHAR(100) NOT NULL,
46	`email` VARCHAR(100) NOT NULL,
47	`password` TEXT NOT NULL
48) ENGINE=INNODB DEFAULT CHARSET=utf8
	COLLATE=utf8_general_ci;
49	
50	
51	INSERT INTO `user` (`id`, `username`, `email`,
52	`password`) VALUES
53	

54	(1, 'user', 'user@gmail.com', '\$2y\$10\$uikyKAFYu5peatxETHuGq.t6wItNJidIoEy0P5pkBQyyMJD qRV390'), (2, 'admin', 'admin@mail.com', '\$2y\$10\$owPcuOliltHxCooZMtYEHOkEnURD39DSJ1/OQcdTKnvaBYJ bwT1D2');
57	
58	ALTER TABLE `buku`
59	ADD PRIMARY KEY (`id`);
60	
61	ALTER TABLE `crud`
62	ADD PRIMARY KEY (`id`);
63	
64	ALTER TABLE `migrations`
65	ADD PRIMARY KEY (`id`);
66	
67	ALTER TABLE `user`
68	ADD PRIMARY KEY (`id`);
69	
70	ALTER TABLE `buku`
71	MODIFY `id` INT(5) UNSIGNED NOT NULL AUTO_INCREMENT;
72	
	ALTER TABLE `crud`
73	MODIFY `id` INT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
74	AUTO_INCREMENT=5;
75	
	ALTER TABLE `migrations`
76	MODIFY `id` BIGINT(20) UNSIGNED NOT NULL
77	AUTO_INCREMENT, AUTO_INCREMENT=5;
78	

79	<pre>ALTER TABLE `user` MODIFY `id` INT(5) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3; COMMIT;</pre>
----	--

RR. Pembahasan buku.sql

Pada database yang dibuat oleh skrip sql tersebut digunakan untuk keperluan web. Setiap tabel dalam struktur database tersebut memiliki tujuan untuk mendukung fungsi keseluruhan sistem. Pada skrip tersebut terdapat pembuatan tabel buku yang berfungsi sebagai repositori untuk menyimpan informasi terkait buku, yaitu judul, penulis, penerbit, dan tahun terbit. Lalu terdapat tabel crud untuk menyimpan informasi dasar tentang pengguna atau mahasiswa, dengan kolom yang mencakup ID, nama, NIM (Nomor Induk Mahasiswa), dan alamat. Tabel migrations digunakan dalam mencatat dan melacak setiap perubahan struktur database. Informasi yang tersimpan di sini mencakup versi migrasi, kelas migrasi, grup, namespace, waktu migrasi, dan batch. Pada skrip tersebut juga terdapat pembuatan tabel user yang bertanggung jawab untuk penyimpanan data pengguna sistem, termasuk ID pengguna, nama pengguna (username), email, dan password yang tersimpan dengan metode enkripsi. Database tersebut dibuat untuk mendukung berbagai fungsi pada aplikasi web, seperti manajemen data buku, pengelolaan informasi pengguna, pelacakan perubahan pada struktur database, serta autentikasi dan manajemen keamanan pengguna.

TAUTAN GITHUB

Berikut adalah tautan untuk source code yang telah dibuat,

<https://github.com/syahputerii/Praktikum-Pemrograman-WEB-II.git> dan
<https://github.com/syahputerii/PRAK601-WEB2.git>