

**LAPORAN AKHIR PRAKTIKUM
PEMROGRAMAN WEB II**



Oleh:

Aditya Oktaviari

NIM. 2210817110008

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

LEMBAR PENGESAHAN

LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN WEB II

Laporan 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 II. Laporan Akhir Praktikum ini dikerjakan oleh:

Nama Praktikan : Aditya Oktaviari

NIM : 2210817110008

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

Ir. Eka Setya Wijaya, S.T., M.Kom.
NIP. 19820508200801 1 010

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI.....	3
DAFTAR TABEL.....	6
DAFTAR GAMBAR	8
MODUL 1: VARIABEL, OPERATOR, DAN ARRAY	10
SOAL 1	10
A. Source Code.....	10
B. Output Program	10
C. Pembahasan	10
SOAL 2	11
A. Source Code.....	11
B. Output Program	12
C. Pembahasan	12
SOAL 3	13
A. Source Code.....	13
B. Output Program	13
C. Pembahasan	14
SOAL 4	15
A. Source Code.....	15
B. Output Program	16
C. Pembahasan	17
SOAL 5	19
A. Source Code.....	19
B. Output Program	21
C. Pembahasan	21
MODUL 2: HTTP REQUEST DAN KONDISIONAL.....	23
SOAL 1	23
A. Source Code.....	23
B. Output Program	25

C. Pembahasan	25
SOAL 2	26
A. Source Code.....	27
B. Output Program	29
C. Pembahasan	29
SOAL 3	32
A. Source Code.....	32
B. Output Program	35
C. Pembahasan	35
SOAL 4	37
A. Source Code.....	37
B. Output Program	38
C. Pembahasan	39
MODUL 3: PERULANGAN DAN KONDISIONAL.....	41
SOAL 1	41
A. Source Code.....	42
B. Output Program	43
C. Pembahasan	43
SOAL 2	45
A. Source Code.....	46
B. Output Program	47
C. Pembahasan	47
SOAL 3	49
A. Source Code.....	50
B. Output Program	51
C. Pembahasan	51
SOAL 4	53
A. Source Code.....	53
B. Output Program	56
C. Pembahasan	56
SOAL 5	58

A. Source Code.....	58
B. Output Program	59
C. Pembahasan	59
MODUL 4: ARRAY	62
SOAL 1	62
A. Source Code.....	62
B. Output Program	64
C. Pembahasan	65
SOAL 2	66
A. Source Code.....	67
B. Output Program	70
C. Pembahasan	70
SOAL 3	72
A. Source Code.....	73
B. Output Program	77
C. Pembahasan	77
MODUL 5: FUNCTION DAN DATABASE.....	79
SOAL	79
A. Source Code.....	81
B. Output Program	97
C. Pembahasan	98
MODUL 6: CRUD DAN LOGIN PADA CODEIGNITER	104
SOAL	104
A. Source Code.....	105
B. Output Program	151
C. Pembahasan	153
TAUTAN GIT.....	161

DAFTAR TABEL

MODUL 1 : VARIABEL, OPERATOR, DAN ARRAY

Tabel 1. Soal 1.....	10
Tabel 2. Source Code Soal 1	10
Tabel 3. Soal 2.....	11
Tabel 4. Source Code Soal 2	11
Tabel 5. Soal 3.....	13
Tabel 6. Source Code Soal 3	13
Tabel 7. Source Code Soal 4.....	16
Tabel 8. Source Code Soal 5	20

MODUL 2 : HTTP REQUEST DAN KONDISIONAL

Tabel 9. Source Code Soal 1	24
Tabel 10. Source Code Soal 2	29
Tabel 11. Source Code Soal 3	34
Tabel 12. Source Code Soal 4	38

MODUL 3 : PERULANGAN DAN KONDISIONAL

Tabel 13. Source Code Soal 1	42
Tabel 14. Source Code Soal 2	46
Tabel 15. Source Code Soal 3	50
Tabel 16. Source Code Soal 4	55
Tabel 17. Source Code Soal 5	59

MODUL 4 : ARRAY

Tabel 18. Source Code Soal 1	64
Tabel 19. Source Code Soal 2	70
Tabel 20. Source Code Soal 3	77

MODUL 5 : FUNCTION DAN DATABASE

Tabel 21. Source Code Koneksi.php.....	81
Tabel 22. Source Code Model.php.....	86
Tabel 23. Source Code Member.php.....	88
Tabel 24. Source Code FormMember.php.....	89
Tabel 25. Source Code Buku.php	91
Tabel 26. Source Code FormBuku.php.....	93
Tabel 27. Source Code Peminjaman.php	95

Tabel 28. Source Code FormPeminjaman.php	96
--	----

MODUL 6 : CRUD DAN LOGIN PADA CODEIGNITER

Tabel 29. Source Code Routes.php	105
Tabel 30. Source Code Buku.php	109
Tabel 31. Source Code Home.php	109
Tabel 32. Source Code Login.php.....	111
Tabel 33. Source Code Register.php.....	113
Tabel 34. Source Code AuthFilter.php	114
Tabel 35. Source Code vw_add.php	118
Tabel 36. Source Code vw_buku.php	125
Tabel 37. Source Code vw_edit.php	129
Tabel 38. Source Code vw_login.php.....	134
Tabel 39. Source Code vw_register.php	139
Tabel 40. Source Code BukuModel.php.....	141
Tabel 41. Source Code UserModel.php.....	141
Tabel 42. Source Code 2024-06-03-103310_Users.php	143
Tabel 43. Source Code 2024-06-03-122804_Buku.php	145
Tabel 44. Source Code Seeds Buku.php	145
Tabel 45. Source Code UserSeeder.php.....	146
Tabel 46. Source Code perpustakaan.sql	150

DAFTAR GAMBAR

MODUL 1 : VARIABEL, OPERATOR, DAN ARRAY

Gambar 1. Screenshot Hasil Jawaban Soal 1	10
Gambar 2. Screenshot Hasil Jawaban Soal 2	12
Gambar 3. Screenshot Hasil Jawaban Soal 3	13
Gambar 4. Soal 4.....	15
Gambar 5. Screenshot Hasil Jawaban Soal 4	16
Gambar 6. Soal 5.....	19
Gambar 7. Screenshot Hasil Jawaban Soal 5	21

MODUL 2 : HTTP REQUEST DAN KONDISIONAL

Gambar 8. Screenshot Hasil Jawaban Soal 1	25
Gambar 9. Screenshot Hasil Jawaban Soal 2	29
Gambar 10. Screenshot Hasil Jawaban Soal 3	35
Gambar 11. Screenshot Hasil Jawaban Soal 4	38

MODUL 3 : PERULANGAN DAN KONDISIONAL

Gambar 12. Screenshot Hasil Jawaban Soal 1	43
Gambar 13. Screenshot Hasil Jawaban Soal 2	47
Gambar 14. Screenshot Hasil Jawaban Soal 3	51
Gambar 15. Screenshot Hasil Jawaban Soal 4	56
Gambar 16. Screenshot Hasil Jawaban Soal 5	59

MODUL 4 : ARRAY

Gambar 17 Screenshot Hasil Jawaban Soal 1 Output 1	64
Gambar 18 Screenshot Hasil Jawaban Soal 1 Output 2	64
Gambar 19 Screenshot Hasil Jawaban Soal 2	70
Gambar 20 Screenshot Hasil Jawaban Soal 3	77

MODUL 5 : FUNCTION DAN DATABASE

Gambar 21. Screenshot Hasil Jawaban Member.php.....	97
Gambar 22. Screenshot Hasil Jawaban FormMember.php	97
Gambar 23. Screenshot Hasil Jawaban Buku.php	97
Gambar 24. Screenshot Hasil Jawaban FormBuku.php	97
Gambar 25. Screenshot Hasil Jawaban Peminjaman.php	98
Gambar 26. Screenshot Hasil Jawaban FormPeminjaman.php	98

MODUL 6 : CRUD DAN LOGIN PADA CODEIGNITER

Gambar 27. Halaman Login.....	151
Gambar 28. Halaman Register	151
Gambar 29. Halaman Home.....	152
Gambar 30. Halaman Untuk Menambahkan Data	152
Gambar 31. Halaman Untuk Edit Data	153

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}

Tabel 1. Soal 1

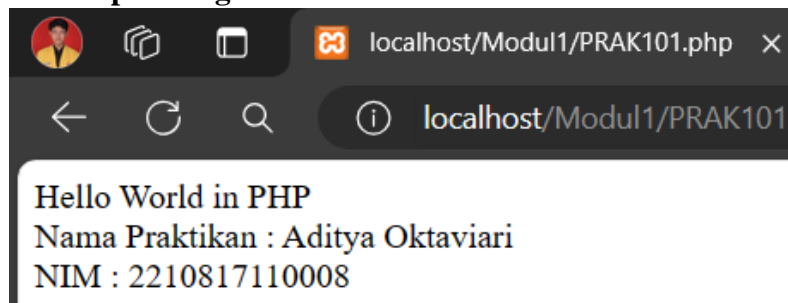
Simpan dengan nama file: PRAK101.php

A. Source Code

1	<?php
2	echo "Hello World in PHP ";
3	echo "Nama Praktikan : Aditya Oktaviari ";
4	echo "NIM : 2210817110008 ";
5	?>

Tabel 2. Source Code Soal 1

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

Pada baris [1], <?php digunakan untuk pembuka di setiap awal code PHP.

Pada baris [2-4], echo digunakan untuk mencetak output,
 digunakan untuk membuat baris baru dalam tampilan HTML.

Pada baris [5], ?> digunakan untuk penutup di setiap akhir code PHP.

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

Tabel 3. Soal 2

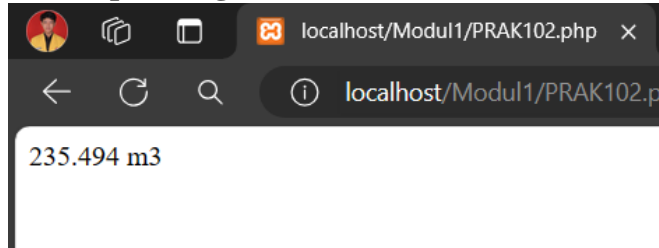
Simpan dengan nama file: PRAK102.php

A. Source Code

1	<?php
2	\$panjang = 8.9;
3	\$lebar = 14.7;
4	\$tinggi = 5.4;
5	\$sisi = 7.9;
6	
7	\$luas_alas = \$panjang * \$lebar;
8	\$volume = (1/3) * \$luas_alas * \$tinggi;
9	
10	echo number_format(\$volume, 3, '.', '') . " m3";
11	?>

Tabel 4. Source Code Soal 2

B. Output Program



Gambar 2. Screenshot Hasil Jawaban Soal 2

C. Pembahasan

Pada baris [1], `<?php` digunakan untuk pembuka di setiap awal code PHP.

Pada baris [2-10], `$panjang`, `$lebar`, `$tinggi`, `$sisi`, `$luas_alas`, `$volume` Merupakan sebuah variabel

Pada baris [10], `echo number_format($volume, 3, '.', '') . " m3";` Digunakan untuk mencetak volume bangun ruang dengan menggunakan fungsi `number_format()` untuk memformat angka dengan tiga angka di belakang koma dan tanpa pemisah ribuan.

Pada baris [11], `?>` digunakan untuk penutup di setiap akhir code PHP.

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

Tabel 5. Soal 3

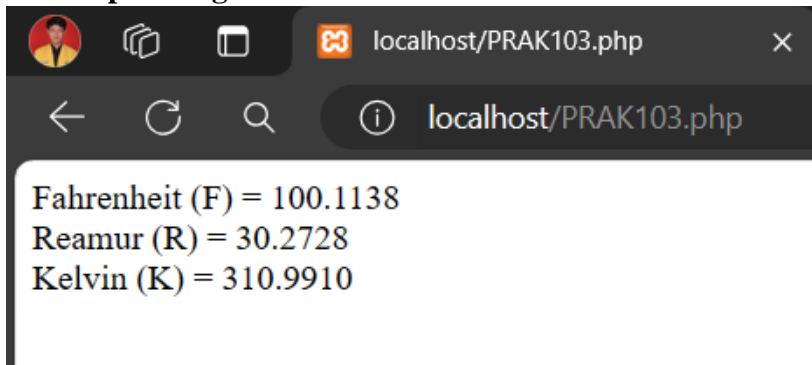
Simpan dengan nama file: PRAK103.php

A. Source Code

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

Tabel 6. Source Code Soal 3

B. Output Program



Gambar 3. Screenshot Hasil Jawaban Soal 3

C. Pembahasan

Pada baris [1], `<?php` digunakan untuk pembuka di setiap awal code PHP.

Pada baris [2-9], `$celcius`, `$fahrenheit`, `$reamur`, `$kelvin` Merupakan sebuah variabel

Pada baris [7-9] `echo""` Digunakan untuk mencetak nilai dalam variabel ke layar. `number_format()` digunakan untuk memformat nilai dengan empat angka di belakang koma tanpa pemisah ribuan. `
` digunakan untuk membuat baris baru dalam tampilan HTML.

Pada baris [10], `?>` digunakan untuk penutup di setiap akhir code PHP.

SOAL 4

Buatlah program dalam Bahasa pemrograman PHP yang menampilkan daftar data seperti ditunjukkan pada Gambar 1. Gunakan **indexed array** dalam implementasinya serta Gunakan kombinasi HTML, CSS dan PHP.

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

Gambar 4. Soal 4

Simpan dengan nama file: PRAK104.php

A. Source Code

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-
width, initial-scale=1.0">
6     <title>Daftar Smartphone Samsung</title>
7     <style>
8         body {
9             font-family:'Times New Roman', Times,
10 serif;
11         }
12
13         table {
14             border: 1px solid
15
16         }
17
18         th, td {
19             border: 1px solid ;
20             text-align: left;
21             padding: 2px;
22         }
23     </style>
24 </head>
```

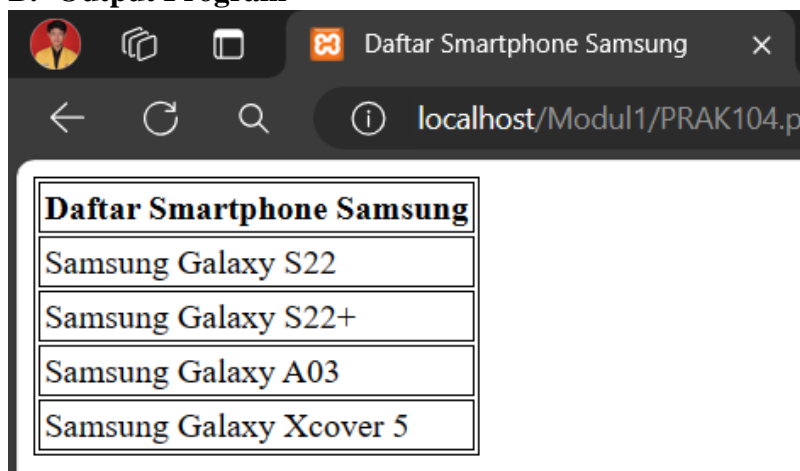
```

25 <body>
26     <div class="container">
27         <table>
28             <tr>
29                 <th>Daftar Smartphone Samsung
30                 <?php
31                     $daftarsmartphone = array(
32                         "Samsung Galaxy S22",
33                         "Samsung Galaxy S22+",
34                         "Samsung Galaxy A03",
35                         "Samsung Galaxy Xcover 5"
36                     );
37
38                 foreach ($daftarsmartphone as
39 $smartphone) {
40                     echo
41 " <tr><td>$smartphone</td></tr>";
42                 }
43             <?>
44         </th>
45     </tr>
46 </table>
47 </div>
48 </body>
49 </html>

```

Tabel 7. Source Code Soal 4

B. Output Program



Gambar 5. Screenshot Hasil Jawaban Soal 4

C. Pembahasan

Pada baris [1], `<!DOCTYPE html>` Digunakan untuk mendeklarasikan tipe dokumen bahwa halaman ini adalah HTML.

Pada baris [2], `<html lang="en">` Digunakan untuk menandakan bahwa bahasa dari dokumen HTML ini menggunakan bahasa inggris.

Pada baris [3], `<head>` Merupakan bagian kepala dari dokumen HTML yang berisi meta-informasi tentang dokumen, seperti karakter set, viewport, dan judul halaman.

Pada baris [4], `<meta charset="UTF-8">` Digunakan untuk mendefinisikan karakter set dokumen sebagai UTF-8, yang mendukung berbagai karakter internasional.

Pada baris [5], `<meta name="viewport" content="width=device-width, initial-scale=1.0">` Digunakan untuk mendefinisikan tampilan halaman agar sesuai dengan lebar perangkat dan awal skala 1:1.

Pada baris [6], `<title>` Digunakan untuk memberi dan menampilkan judul pada dokumen HTML

Pada baris[7], `<style>` Digunakan untuk memodifikasi style dari elemen HTML.

Pada baris [8-10] `body { font-family: 'Times New Roman', Times, serif; }` Digunakan untuk mengatur jenis font untuk teks dalam dokumen HTML.

Pada baris [12-15], `table { border: 1px solid; }` Digunakan untuk mengatur batas-batas tabel dengan garis hitam.

Pada baris [17-21] `th, td { border: 1px solid; text-align: left; padding: 2px; }` Digunakan untuk mengatur batas-batas sel header dan sel data dalam tabel dengan garis hitam berketebalan 1 piksel, mengatur teks dalam sel menjadi rata kiri, dan memberikan jarak antara teks dan batas sel sebesar 2 piksel.

Pada baris [24], `<body>` Merupakan elemen body dokumen HTML, yang berisi konten yang akan ditampilkan di browser.

Pada baris [25], `<div>` Digunakan untuk Memulai sebuah divisi (bagian) dalam dokumen.

Pada baris [26-43], `<table>` digunakan untuk mendeklarasikan pembuka dan penutup tabel.

Pada baris [27-42], `<tr>` Digunakan untuk membuat baris.

Pada baris [28], `<th>` Digunakan untuk mendefinisikan tiap nama kolom pada tabel.

Pada baris [29], `<?php` digunakan untuk pembuka di setiap awal code PHP.

Pada baris [30-35], `$daftarsmartphone = array("Samsung Galaxy S22", "Samsung Galaxy S22+", "Samsung Galaxy A03", "Samsung Galaxy Xcover 5");` Digunakan untuk menginisialisasi array dalam PHP dengan nama `$daftarsmartphone` yang berisi daftar model smartphone Samsung.

Pada baris [37], `foreach ($daftarsmartphone as $smartphone) { echo "<tr><td>$smartphone</td></tr>"; }` Digunakan untuk melooping melalui setiap elemen dalam array `$daftarsmartphone` dan mencetaknya dalam sel data tabel sebagai satu baris baru.

Pada baris [40], `?>` digunakan untuk penutup di setiap akhir code PHP.

SOAL 5

Buatlah program dalam Bahasa pemrograman PHP yang menampilkan daftar data seperti ditunjukkan pada Gambar 1. Gunakan **associative array** dalam implementasinya serta Gunakan kombinasi HTML, CSS dan PHP.

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

Gambar 6. Soal 5

Simpan dengan nama file: PRAK105.php

A. Source Code

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-
width, initial-scale=1.0">
6     <title>Daftar Smartphone Samsung</title>
7     <style>
8         body {
9             font-family:'Times New Roman', Times,
10 serif;
11         }
12
13         table {
14             border: 1px solid
15
16         }
17
18         th, td {
19             border: 1px solid ;
20             text-align: left;
21             padding: 2px;
22
23         }
24
```

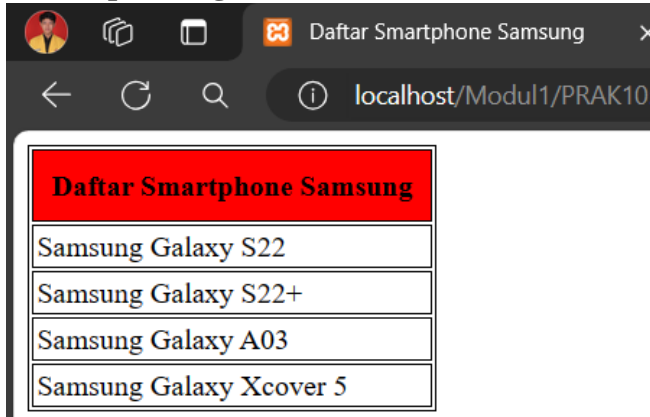
```

25         th.judul{
26             background-color: red;
27             padding: 10px;
28         }
29     </style>
30 </head>
31 <body>
32     <div class="container">
33         <table>
34             <tr>
35                 <th class="judul">Daftar Smartphone
36                 </th>
37             <tr>
38                 <td>
39                     <?php
40                         $daftarsmartphone = array(
41                             "Samsung Galaxy S22" => "Samsung
42                             Galaxy S22",
43                             "Samsung Galaxy S22+" => "Samsung
44                             Galaxy S22+",
45                             "Samsung Galaxy A03" => "Samsung
46                             Galaxy A03",
47                             "Samsung Galaxy Xcover 5" => "Samsung
48                             Galaxy Xcover 5"
49                         );
50                     foreach ($daftarsmartphone as $key =>
51                         $smartphone) {
52                         echo
53                         "<tr><td>$smartphone</td></tr>";
54                     }
55                 <?>
56             </table>
57         </div>
58     </body>
59 </html>

```

Tabel 8. Source Code Soal 5

B. Output Program



Gambar 7. Screenshot Hasil Jawaban Soal 5

C. Pembahasan

Pada baris [1], `<!DOCTYPE html>` Digunakan untuk mendeklarasikan tipe dokumen bahwa halaman ini adalah HTML.

Pada baris [2], `<html lang="en">` Digunakan untuk menandakan bahwa bahasa dari dokumen HTML ini menggunakan bahasa inggris.

Pada baris [3], `<head>` Merupakan bagian kepala dari dokumen HTML yang berisi meta-informasi tentang dokumen, seperti karakter set, viewport, dan judul halaman.

Pada baris [4], `<meta charset="UTF-8">` Digunakan untuk mendefinisikan karakter set dokumen sebagai UTF-8, yang mendukung berbagai karakter internasional.

Pada baris [5], `<meta name="viewport" content="width=device-width, initial-scale=1.0">` Digunakan untuk mendefinisikan tampilan halaman agar sesuai dengan lebar perangkat dan awal skala 1:1.

Pada baris [6], `<title>` Digunakan untuk memberi dan menampilkan judul pada dokumen HTML

Pada baris[7], `<style>` Digunakan untuk memodifikasi style dari elemen HTML.

Pada baris [8-10] `body { font-family: 'Times New Roman', Times, serif; }` Digunakan untuk mengatur jenis font untuk teks dalam dokumen HTML.

Pada baris [12-15], `table { border: 1px solid; }` Digunakan untuk mengatur batas-batas tabel dengan garis hitam.

Pada baris [17-21], `th, td { border: 1px solid; text-align: left; padding: 2px; }` Digunakan untuk mengatur batas-batas sel header dan sel data dalam tabel dengan garis hitam berketebalan 1 piksel, mengatur teks dalam sel menjadi rata kiri, dan memberikan jarak antara teks dan batas sel sebesar 2 piksel.

Pada baris [24], `th.judul{` Digunakan untuk mendefinisikan gaya untuk elemen `<th>` yang memiliki kelas CSS `judul`.

Pada baris [25-26], `background-color: red;` Digunakan untuk mengatur warna latar belakang elemen `<th>` dengan nilai warna merah. `padding: 10px;` Digunakan untuk mengatur padding (jarak antara konten dan tepi) untuk elemen `<th>` sebesar 10 piksel di semua sisi (atas, kanan, bawah, kiri).

Pada baris [30], `<body>` Merupakan elemen body dokumen HTML, yang berisi konten yang akan ditampilkan di browser.

Pada baris [31], `<div>` Digunakan untuk Memulai sebuah divisi (bagian) dalam dokumen.

Pada baris [32-48], `<table>` digunakan untuk mendeklarasikan pembuka dan penutup tabel.

Pada baris [33-35], `<tr>` Digunakan untuk membuat baris.

Pada baris [34], `<th>` Digunakan untuk mendefinisikan tiap nama kolom pada tabel.

Pada baris [36], `<?php` digunakan untuk pembuka di setiap awal code PHP.

Pada baris [30-35], `$daftarsmartphone = array("Samsung Galaxy S22" => "Samsung Galaxy S22", "Samsung Galaxy S22+" => "Samsung Galaxy S22+", "Samsung Galaxy A03" => "Samsung Galaxy A03", "Samsung Galaxy Xcover 5" => "Samsung Galaxy Xcover 5");` Digunakan untuk menginisialisasi sebuah array asosiatif dengan nama `$daftarsmartphone`.

Pada baris [44], `foreach ($daftarsmartphone as $key => $smartphone) {echo "<tr><td>$smartphone</td></tr>";}` Digunakan untuk melooping melalui setiap elemen dalam array `$daftarsmartphone` dan mencetaknya dalam sel data tabel sebagai satu baris baru.

Pada baris [47], `?>` digunakan untuk penutup di setiap akhir code PHP.

MODUL 2: HTTP 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

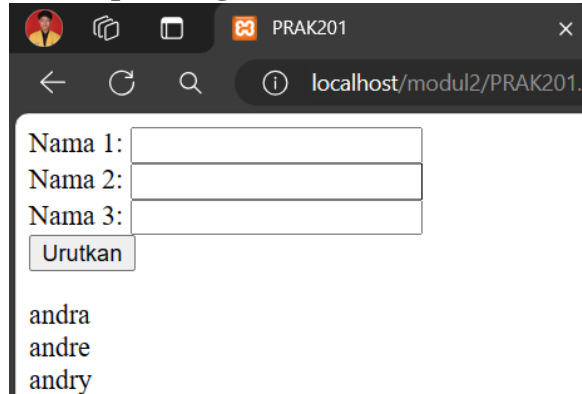
A. Source Code

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>PRAK201</title>
5 </head>
6 <body>
7
8 <form      method="post"      action="<?php      echo
$_SERVER['PHP_SELF']; ?>">
9     Nama 1: <input type="text" name="nama1"><br>
10    Nama 2: <input type="text" name="nama2"><br>
11    Nama 3: <input type="text" name="nama3"><br>
12    <input      type="submit"      name="submit"
13 value="Urutkan">
14 </form>
15
16 <?php
17 if ($_SERVER["REQUEST_METHOD"] == "POST") {
18     $nama1 = $_POST['nama1'];
```

19	\$nama2 = \$_POST['nama2'];
20	\$nama3 = \$_POST['nama3'];
21	
22	if (\$nama1 <= \$nama2 && \$nama1 <= \$nama3) {
23	if (\$nama2 <= \$nama3) {
24	\$hasilurutan = "\$nama1 \$nama2
25	\$nama3 ";
26	} else {
27	\$hasilurutan = "\$nama1 \$nama3
28	\$nama2 ";
29	} elseif (\$nama2 <= \$nama1 && \$nama2 <= \$nama3)
30	{
31	if (\$nama1 <= \$nama3) {
32	\$hasilurutan = "\$nama2 \$nama1
33	\$nama3 ";
34	} else {
35	\$hasilurutan = "\$nama2 \$nama3
36	\$nama1 ";
37	} else {
38	if (\$nama1 <= \$nama2) {
39	\$hasilurutan = "\$nama3 \$nama1
40	\$nama2 ";
41	} else {
42	\$hasilurutan = "\$nama3 \$nama2
43	\$nama1 ";
44	}
45	}
46	}
	echo "<p>\$hasilurutan</p>";
	}
	?>
	</body>
	</html>

Tabel 9. Source Code Soal 1

B. Output Program



Gambar 8. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

Pada baris [1] terdapat `<!DOCTYPE html>` yang digunakan untuk membangun dan menyusun halaman web.

Pada baris [2] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [3] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [4] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [5] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [6] terdapat `<body>` digunakan untuk memberikan instruksi atau perintah kepada browser bagaimana objek yang berupa teks, video, dan lain-lain akan ditampilkan berdasar tag yang digunakan.

Pada baris [8-13] Baris pertama berfungsi sebagai pembuka tag formulir HTML dengan atribut metode yang diatur sebagai "post". Metode HTTP POST digunakan untuk mengirimkan data ke server, dan atribut action menentukan URL atau skrip yang akan dipanggil saat formulir dikirim. Di sini, nilai tindakan diatur menjadi `$_SERVER['PHP_SELF']`, variabel lingkungan PHP yang memberi nama skrip PHP yang sedang dieksekusi saat ini. Input field dalam bentuk teks dibaris kedua hingga keempat memungkinkan pengguna memasukkan data; masing-masing field memiliki nama yang berbeda untuk mengidentifikasi data saat dikirim ke server.

Terakhir, di baris kelima terdapat tombol "submit", yang ketika ditekan akan mengirimkan data yang dimasukkan oleh pengguna ke skrip PHP sesuai dengan atribut action.

Pada baris [16-19] Dalam bahasa pemrograman PHP, blok if statement di atas digunakan untuk mengetahui apakah permintaan yang diterima oleh server merupakan metode POST. Pada baris pertama, variabel superglobal `$_SERVER["REQUEST_METHOD"]` == "POST" melakukan pengecekan apakah metode HTTP yang digunakan untuk permintaan saat ini sama dengan "POST". Jika kondisi ini benar, itu menunjukkan bahwa form sebelumnya telah dikirim menggunakan metode POST. Jika itu benar, blok kode dalam if statement akan dieksekusi. Pada baris kedua hingga keempat, nilai-nilai yang dikirim dari form HTML sebelumnya yang menggunakan metode POST disimpan dalam variabel PHP (`$nama1`, `$nama2`, `$nama3`) dengan menggunakan `$_POST['nama1']`, `$_POST['nama2']`, dan `$_POST['nama3']`. Oleh karena itu, variabel-variabel ini akan mengandung informasi yang telah dimasukkan oleh pengguna melalui form sebelumnya, sehingga dapat digunakan dalam proses selanjutnya.

Pada baris [21-39] Merupakan sebuah blok kontrol alur yang digunakan untuk mengurutkan tiga nilai yang disimpan dalam variabel `$nama1`, `$nama2`, dan `$nama3`. Blok if-elseif-else pertama memeriksa nilai-nilai tersebut untuk menentukan urutan mana yang merupakan nilai terkecil. Pada setiap kondisi if, perbandingan dilakukan antara nilai-nilai tersebut untuk menentukan urutan yang benar secara ascending. Jika kondisi pada if pertama terpenuhi, artinya `$nama1` merupakan nilai terkecil, maka dilakukan pengecekan lebih lanjut dengan if di dalamnya untuk memastikan urutan antara `$nama2` dan `$nama3`. Jika kondisi tersebut benar, maka variabel `$hasilurutan` akan berisi urutan `$nama1`, `$nama2`, dan `$nama3`. Jika tidak, urutan `$nama1` dan `$nama3` akan ditukar. Proses serupa dilakukan untuk kondisi elseif dan else, di mana masing-masing mengurutkan nilai-nilai yang tersisa setelah memeriksa nilai terkecil sebelumnya. Akhirnya, variabel `$hasilurutan` akan berisi urutan yang benar dari nilai-nilai yang dimasukkan oleh pengguna melalui form sebelumnya, dengan menggunakan tag HTML `
` untuk memisahkan setiap nama sehingga ditampilkan dalam baris yang berbeda.

Pada baris [41] `echo "<p>$hasilurutan</p>";` merupakan sebuah pernyataan echo dalam bahasa PHP yang digunakan untuk menampilkan isi dari variabel `$hasilurutan` ke dalam halaman web.

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 tampilan 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

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>PRAK202</title>
5 </head>
6 <body>
7     <form      method="post"      action="<?php      echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
8         Nama: <input type="text" name="nama">
9         <?php

```

```

10         if ($_SERVER["REQUEST_METHOD"] == "POST" &&
empty($_POST['nama'])) { ?>
11             <span style="color:red">*nama tidak
boleh kosong</span>
12             <?php }
13             ?>
14             <br>
15
16             NIM: <input type="text" name="nim">
17             <?php
18             if ($_SERVER["REQUEST_METHOD"] == "POST" &&
empty($_POST['nim'])) { ?>
19                 <span style="color:red">*nim tidak boleh
kosong</span>
20                 <?php }
21                 ?>
22                 <br>
23
24                 Jenis Kelamin: <?php if
($_SERVER["REQUEST_METHOD"] == "POST" &&
empty($_POST['jenis_kelamin'])) { ?>
25                     <span style="color:red">*jenis kelamin
tidak boleh kosong</span>
26                     <?php } ?><br>
27
28                     <input type="radio" name="jenis_kelamin"
value="Laki-laki">Laki-laki <br>
29                     <input type="radio" name="jenis_kelamin"
value="Perempuan">Perempuan
30                     <br>
31
32                     <input type="submit" name="submit"
value="Submit">
33                     </form>
34
35                     <?php
36                     if ($_SERVER["REQUEST_METHOD"] == "POST") {
37                         $nama = $_POST['nama'];
38                         $nim = $_POST['nim'];
39                         $jenis_kelamin = $_POST['jenis_kelamin'];
40
41                         if (!empty($nama) && !empty($nim) &&
!empty($jenis_kelamin)) {
42                             echo "<br><b>Output : </b><br>";
43                             echo "<br>$nama <br>";

```

44	echo "\$nim ";
45	echo "\$jenis_kelamin";
46	}
47	}
48	?>
49	</body>
50	</html>

Tabel 10. Source Code Soal 2

B. Output Program

Output :

Aditya Oktaviari
2210817110008
Laki-laki

Gambar 9. Screenshot Hasil Jawaban Soal 2

C. Pembahasan

Pada baris [1] terdapat `<!DOCTYPE html>` yang digunakan untuk membangun dan menyusun halaman web.

Pada baris [2] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [3] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [4] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [5] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [6] terdapat `<body>` digunakan untuk memberikan instruksi atau perintah kepada browser bagaimana objek yang berupa teks, video, dan lain-lain akan ditampilkan berdasar tag yang digunakan.

Pada baris [7-26] Baris ketujuh hingga kedua puluh enam adalah formulir HTML yang digunakan untuk mengumpulkan data pengguna seperti nama, NIM, dan jenis kelamin. Pada tag formulir, atribut metode diatur sebagai "post", yang menunjukkan bahwa data akan dikirimkan ke server melalui metode POST. Atribut aksi menunjuk ke `$_SERVER["PHP_SELF"]` yang akan menangani pengolahan data saat formulir diserahkan. Selain itu, ada kolom nama dan NIM di mana pengguna dapat memasukkan informasi tersebut. Setelah setiap kolom dimasukkan, terdapat blok PHP yang digunakan untuk mengetahui apakah data tersebut kosong setelah formulir dikirim dengan menggunakan `$_SERVER["REQUEST_METHOD"] == "POST"` dan `empty($_POST['nama'])`. Jika data tidak kosong, maka akan ditampilkan pesan kesalahan dalam warna merah menggunakan tag `span`, memberi tahu pengguna bahwa kolom tersebut tidak boleh kosong. Di antara kolom input NIM dan jenis kelamin, ada pemeriksaan khusus untuk jenis kelamin karena jenis kelamin mungkin memiliki opsi yang harus dipilih daripada diisi. Jika jenis kelamin belum dipilih saat mengisi formulir, pesan kesalahan akan muncul. Oleh karena itu, kode tersebut membuat formulir dengan validasi sederhana untuk memastikan bahwa informasi pengguna lengkap sebelum diproses lebih lanjut.

Pada baris [28-32] Menampilkan dua opsi radio button yang memungkinkan pengguna memilih jenis kelamin mereka, dengan opsi "Laki-laki" dan "Perempuan". Kedua opsi tersebut memiliki atribut `name` yang sama, `"jenis_kelamin"`, memastikan hanya satu opsi yang dapat dipilih. Setiap opsi memiliki atribut `value` yang berbeda, yang akan dikirim ke server saat form disubmit. Setelah opsi radio button, tag `
` digunakan untuk membuat baris baru. Input button bertipe "submit" juga ditampilkan, memungkinkan pengguna untuk mengirimkan form setelah memilih jenis kelamin mereka. Dengan demikian, form tersebut memungkinkan pengguna untuk memasukkan informasi jenis kelamin mereka dan kemudian mengirimkannya ke server untuk diproses lebih lanjut saat tombol "Submit" ditekan.

Pada baris [36-39] Merupakan sebuah blok `if` statement dalam bahasa PHP yang digunakan untuk memeriksa apakah request yang diterima oleh server merupakan metode POST. Jika kondisi tersebut terpenuhi, maka variabel `$nama`, `$nim`, dan `$jenis_kelamin` akan diinisialisasi dengan nilai-nilai yang dikirimkan dari form sebelumnya menggunakan metode POST. Variabel `$nama` akan berisi nama yang dimasukkan oleh pengguna, variabel `$nim` akan berisi NIM yang dimasukkan oleh pengguna, dan variabel `$jenis_kelamin` akan berisi jenis kelamin yang dipilih oleh pengguna melalui form sebelumnya. Dengan demikian, data yang dikirimkan dari form dapat diakses dan digunakan dalam proses selanjutnya dalam skrip PHP.

Pada baris [41-46] Blok code di atas merupakan sebuah kondisi `if` yang digunakan untuk memeriksa apakah variabel `$nama`, `$nim`, dan `$jenis_kelamin` memiliki nilai yang tidak kosong (`empty`). Jika semua variabel memiliki nilai yang

tidak kosong, maka blok kode di dalam if akan dieksekusi. Pada baris pertama, terdapat pernyataan echo untuk menampilkan teks "
Output :
", yang memberikan penanda bahwa output dari data yang dimasukkan oleh pengguna akan ditampilkan di bawahnya. Selanjutnya, nilai dari variabel \$nama, \$nim, dan \$jenis_kelamin akan ditampilkan masing-masing dalam baris baru menggunakan tag
, dan nama variabel juga akan ditebalkan dengan tag untuk menonjolkan informasi output kepada pengguna. Oleh karena itu, sebuah output yang terstruktur akan dihasilkan jika semua data yang dimasukkan pengguna tidak kosong.

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

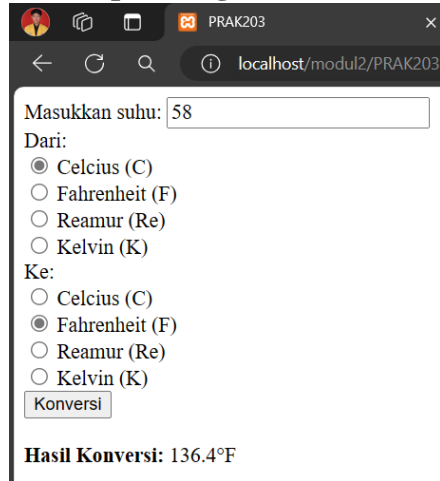
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>PRAK203</title>
5 </head>
6 <body>
7     <form method="post">
8         Nilai : <input type="text" name="suhu"
required><br>
9         Dari :<br>
10        <input type="radio" name="from_unit"
value="celcius" checked> Celcius (C)<br>
11        <input type="radio" name="from_unit"
value="fahrenheit"> Fahrenheit (F)<br>
12        <input type="radio" name="from_unit"
value="reamur"> Reamur (Re)<br>
13        <input type="radio" name="from_unit"
value="kelvin"> Kelvin (K)<br>
14        Ke :<br>
15        <input type="radio" name="to_unit"
value="celcius" checked> Celcius (C)<br>
```


16	<code><input type="radio" name="to_unit"</code>
	<code>value="fahrenheit"> Fahrenheit (F)
</code>
17	<code><input type="radio" name="to_unit"</code>
	<code>value="reamur"> Reamur (Re)
</code>
18	<code><input type="radio" name="to_unit"</code>
	<code>value="kelvin"> Kelvin (K)
</code>
19	<code><input type="submit" name="submit"</code>
	<code>value="Konversi"></code>
20	<code></form></code>
21	<code>
</code>
22	
23	<code><?php</code>
24	<code>if(isset(\$_POST['submit'])) {</code>
25	<code> \$suhu = \$_POST['suhu'];</code>
26	<code> \$from_unit = \$_POST['from_unit'];</code>
27	<code> \$to_unit = \$_POST['to_unit'];</code>
28	
29	<code> if(\$from_unit == \$to_unit) {</code>
30	<code> \$result = \$suhu;</code>
31	<code> echo "Hasil Konversi: " . \$result;</code>
32	<code> }elseif(\$from_unit == 'celcius' && \$to_unit</code>
	<code>== 'fahrenheit') {</code>
33	<code> \$result = (\$suhu * 9/5) + 32;</code>
34	<code> echo "Hasil Konversi: " . \$result</code>
	<code>. "°F";</code>
35	<code> }elseif(\$from_unit == 'celcius' && \$to_unit</code>
	<code>== 'reamur') {</code>
36	<code> \$result = \$suhu * 4/5;</code>
37	<code> echo "Hasil Konversi: " . \$result</code>
	<code>. "°R";</code>
38	<code> }elseif(\$from_unit == 'celcius' && \$to_unit</code>
	<code>== 'kelvin') {</code>
39	<code> \$result = \$suhu + 273.15;</code>
40	<code> echo "Hasil Konversi: " . \$result</code>
	<code>. "°K";</code>
41	<code> }elseif(\$from_unit == 'fahrenheit' &&</code>
	<code>\$to_unit == 'celcius') {</code>
42	<code> \$result = (\$suhu - 32) * 5/9;</code>
43	<code> echo "Hasil Konversi: " . \$result</code>
	<code>. "°C";</code>
44	<code> }elseif(\$from_unit == 'fahrenheit' &&</code>
	<code>\$to_unit == 'reamur') {</code>
45	<code> \$result = (\$suhu - 32) * 4/9;</code>
46	<code> echo "Hasil Konversi: " . \$result</code>
	<code>. "°R";</code>

47	}elseif(\$from_unit == 'fahrenheit' &&
	\$to_unit == 'kelvin'){
48	\$result = (\$suhu + 459.67) * 5/9;
49	echo "Hasil Konversi: " . \$result
	. "°K";
50	}elseif(\$from_unit == 'reamur' && \$to_unit ==
	'celcius'){
51	\$result = \$suhu * 5/4;
52	echo "Hasil Konversi: " . \$result
	. "°C";
53	}elseif(\$from_unit == 'reamur' && \$to_unit ==
	'fahrenheit'){
54	\$result = (\$suhu * 9/4) + 32;
55	echo "Hasil Konversi: " . \$result
	. "°F";
56	}elseif(\$from_unit == 'reamur' && \$to_unit ==
	'kelvin'){
57	\$result = (\$suhu * 5/4) + 273.15;
58	echo "Hasil Konversi: " . \$result
	. "°K";
59	}elseif(\$from_unit == 'kelvin' && \$to_unit ==
	'celcius'){
60	\$result = \$suhu - 273.15;
61	echo "Hasil Konversi: " . \$result
	. "°C";
62	}elseif(\$from_unit == 'kelvin' && \$to_unit ==
	'fahrenheit'){
63	\$result = (\$suhu * 9/5) - 459.67;
64	echo "Hasil Konversi: " . \$result
	. "°F";
65	}elseif(\$from_unit == 'kelvin' && \$to_unit ==
	'reamur'){
66	\$result = (\$suhu - 273.15) * 4/5;
67	echo "Hasil Konversi: " . \$result
	. "°R";
68	}
69	}
70	?>
71	</body>
72	</html>

Tabel 11. Source Code Soal 3

B. Output Program



Masukkan suhu:

Dari:

☒ Celcius (C)
☐ Fahrenheit (F)
☐ Reamur (Re)
☐ Kelvin (K)

Ke:

☐ Celcius (C)
☒ Fahrenheit (F)
☐ Reamur (Re)
☐ Kelvin (K)

Hasil Konversi: 136.4°F

Gambar 10. Screenshot Hasil Jawaban Soal 3

C. Pembahasan

Pada baris [1] terdapat `<!DOCTYPE html>` yang digunakan untuk membangun dan menyusun halaman web.

Pada baris [2] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [3] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [4] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [5] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [6] terdapat `<body>` digunakan untuk memberikan instruksi atau perintah kepada browser bagaimana objek yang berupa teks, video, dan lain-lain akan ditampilkan berdasar tag yang digunakan.

Pada baris [7-20] Merupakan sebuah HTML form yang memungkinkan pengguna untuk melakukan konversi suhu dari satu unit ke unit lainnya. Form ini menggunakan metode POST untuk mengirimkan data ke server. Di bagian pertama, pengguna diminta untuk memasukkan nilai suhu dalam input teks dengan nama "suhu", yang diberi atribut "required" untuk memastikan bahwa input tersebut tidak boleh kosong. Selanjutnya, terdapat opsi untuk memilih unit suhu awal dari radio button dengan

atribut name "from_unit" dan opsi untuk memilih unit suhu tujuan dari radio button dengan atribut name "to_unit". Setiap opsi memiliki nilai yang berbeda (Celcius, Fahrenheit, Reamur, Kelvin) yang akan dikirim ke server sebagai unit suhu yang dipilih oleh pengguna. Radio button untuk Celcius (C) diatur sebagai default untuk kedua unit suhu (from_unit dan to_unit) dengan menggunakan atribut "checked". Terakhir, terdapat tombol "Konversi" yang akan memicu pengiriman data ke server saat ditekan. Dengan demikian, form tersebut memberikan pengguna kemampuan untuk memasukkan nilai suhu, memilih unit suhu awal dan tujuan, dan melakukan konversi suhu sesuai dengan preferensi mereka.

Pada baris [23-70] Merupakan sebuah blok PHP yang akan dieksekusi saat form dikirimkan (dengan mengecek apakah tombol "submit" telah ditekan). Pertama, kode tersebut mengambil nilai suhu yang dimasukkan oleh pengguna, serta unit suhu awal dan tujuan yang dipilih dari form menggunakan `$_POST`. Selanjutnya, blok kode tersebut menggunakan serangkaian kondisional if-elseif untuk melakukan konversi suhu sesuai dengan unit suhu yang dipilih oleh pengguna. Setiap kondisi menghitung nilai suhu yang dikonversi sesuai dengan rumus konversi yang sesuai dengan unit suhu awal dan tujuan yang dipilih. Hasil konversi kemudian disimpan dalam variabel `$result` dan ditampilkan kepada pengguna menggunakan pernyataan echo, disertai dengan pesan yang menunjukkan unit suhu hasil konversi. Dengan demikian, blok kode tersebut menghasilkan output yang menampilkan hasil konversi suhu sesuai dengan pilihan unit suhu yang dimasukkan oleh pengguna.

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 yang merupakan Ejaan dari bilangan tersebut atau batas limit bilangan.

Input	Output
3	Satuan
0	Nol
1000	Anda Menginput Melebihi Limit Bilangan
632	Ratusan
13	Belasan

Output yang diinginkan :

Nilai :

Hasil: belasan

Simpan dengan nama file: PRAK204.php

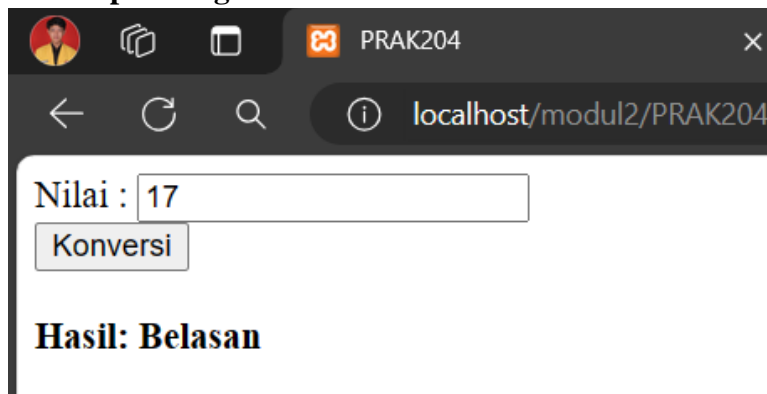
A. Source Code

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>membaca ejaan dari bilangan cacah</title>
5 </head>
6 <body>
7     <form method="post" action="<?php echo
  $_SERVER['PHP_SELF']; ?>">
8         Nilai : <input type="text" name="angka"><br>
9         <button type="submit">Konversi</button>
10    </form>
11
12    <?php
13        function konversiAngkaKeHuruf($angka) {
14            if ($angka == 0) {
15                return 'Nol';
16            } elseif ($angka >= 1 && $angka <= 10) {
```

17	return 'Satuan';
18	} elseif (\$angka >= 11 && \$angka <= 19) {
19	return 'Belasan';
20	} elseif (\$angka >= 20 && \$angka <= 99) {
21	return 'Puluhan';
22	} elseif (\$angka >= 100 && \$angka <= 999) {
23	return 'Ratusan';
24	} else {
25	return 'Anda Menginput Melebihi Limit
	Bilangan';
26	}
27	}
28	
29	\$angka = isset(\$_POST['angka']) ? \$_POST['angka']
30	: '';
31	
32	if (!empty(\$angka)) {
	\$hasilKonversi =
33	konversiAngkaKeHuruf(\$angka);
34	}
35	
36	echo "Hasil: \$hasilKonversi"
37	?>
38	
39	</body>
	</html>

Tabel 12. Source Code Soal 4

B. Output Program



Gambar 11. Screenshot Hasil Jawaban Soal 4

C. Pembahasan

Pada baris [1] terdapat `<!DOCTYPE html>` yang digunakan untuk membangun dan menyusun halaman web.

Pada baris [2] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [3] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [4] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [5] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [6] terdapat `<body>` digunakan untuk memberikan instruksi atau perintah kepada browser bagaimana objek yang berupa teks, video, dan lain-lain akan ditampilkan berdasar tag yang digunakan.

Pada baris [7-10] Merupakan sebuah HTML form yang memungkinkan pengguna untuk memasukkan nilai yang akan dikonversi. Form ini menggunakan metode POST untuk mengirimkan data ke halaman PHP yang sama (menggunakan `$_SERVER['PHP_SELF']`) saat tombol "Konversi" ditekan. Di dalam form, terdapat input teks dengan nama "angka" yang memungkinkan pengguna memasukkan nilai yang akan dikonversi. Selain itu, terdapat tombol submit yang akan memicu pengiriman data form ke server saat ditekan oleh pengguna. Dengan demikian, form tersebut memberikan pengguna kemampuan untuk memasukkan nilai yang akan dikonversi, dan ketika tombol "Konversi" ditekan, data tersebut akan dikirimkan ke halaman PHP yang sama untuk diproses lebih lanjut.

Pada baris [14-29] Merupakan sebuah fungsi PHP yang disebut `konversiAngkaKeHuruf()`. Fungsi ini menerima satu parameter, yaitu `$angka`, yang merupakan nilai numerik yang akan dikonversi menjadi kategori berdasarkan rentang nilai tertentu. Di dalam fungsi, terdapat serangkaian kondisional if-elseif yang memeriksa nilai `$angka` dan mengembalikan string yang sesuai dengan kategori nilai tersebut. Jika `$angka` sama dengan 0, fungsi akan mengembalikan string 'Nol'. Selanjutnya, jika `$angka` berada dalam rentang 1 hingga 10, fungsi akan mengembalikan string 'Satuan'. Begitu pula untuk rentang nilai 11 hingga 19 (mengembalikan 'Belasan'), 20 hingga 99 (mengembalikan 'Puluhan'), dan 100 hingga

999 (mengembalikan 'Ratusan'). Terakhir, jika nilai `$angka` melebihi 999, fungsi akan mengembalikan string 'Anda Menginput Melebihi Limit Bilangan'. Kemudian, diluar fungsi, nilai `$angka` diambil dari data yang dikirimkan oleh form sebelumnya menggunakan metode POST. Variabel `$angka` diinisialisasi dengan nilai dari input 'angka' jika data tersebut ada, dan jika tidak, nilai variabel tersebut diatur sebagai string kosong. Dengan demikian, fungsi tersebut dapat digunakan untuk mengonversi nilai numerik menjadi kategori berdasarkan rentang nilai tertentu, dan variabel `$angka` siap digunakan dalam pemrosesan lebih lanjut, seperti menampilkan hasil konversi ke dalam halaman web.

Pada baris [31-35] Merupakan kondisi if yang memeriksa apakah variabel `$angka` memiliki nilai yang tidak kosong. Jika `$angka` tidak kosong, maka blok kode di dalam if akan dieksekusi. Di dalam blok tersebut, variabel `$hasilKonversi` diinisialisasi dengan hasil pemanggilan fungsi `konversiAngkaKeHuruf()` dengan argumen nilai dari `$angka`, yang akan mengonversi nilai angka menjadi kategori berdasarkan rentang nilai tertentu, seperti 'Nol', 'Satuan', 'Belasan', 'Puluhan', 'Ratusan', atau 'Anda Menginput Melebihi Limit Bilangan'. Baris kedua menampilkan hasil konversi yang disimpan dalam variabel `$hasilKonversi` dengan menggunakan tag `` untuk memberikan efek teks yang tebal. Pemanggilan echo ini akan menampilkan hasil konversi ke dalam halaman web setelah pengguna memasukkan nilai angka melalui form dan tombol "Konversi" ditekan.

.

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 output 2:

Jumlah Peserta :

Peserta ke-1

Peserta ke-2

Peserta ke-3

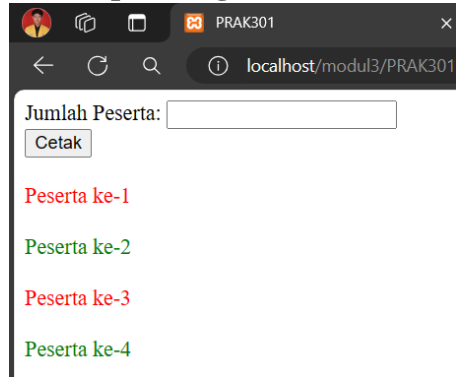
Simpan dengan nama file: PRAK301.php

A. Source Code

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>PRAK301</title>
5 </head>
6 <body>
7
8 <form method="post" action="">
9     <label                for="jumlah_peserta">Jumlah
10 Peserta:</label>
11     <input                type="text"            id="jumlah_peserta"
12 name="jumlah_peserta" min="1" required><br>
13     <button type="submit">Cetak</button><br>
14 </form>
15 <?php
16 if ($_SERVER["REQUEST_METHOD"] == "POST") {
17     $jumlah_peserta = $_POST['jumlah_peserta'];
18     $counter = 1;
19
20     while ($counter <= $jumlah_peserta) {
21         $color = ($counter % 2 != 0) ? 'red' :
22 'green';
23         echo '<span style="color: ' . $color . ' ;
24 font-family: \\'Times New Roman\\', Times,
25 serif;"><br>Peserta ke-' . $counter . '</span><br>';
26         $counter++;
27     }
28 }
29 ?>
30 </body>
31 </html>
```

Tabel 13. Source Code Soal 1

B. Output Program



Gambar 12. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

Pada baris [1] terdapat `<!DOCTYPE html>` yang digunakan untuk membangun dan menyusun halaman web.

Pada baris [2] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [3] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [4] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [5] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [6] terdapat `<body>` yang digunakan untuk mendefinisikan isi atau badan suatu dokumen.

Pada baris [8-12] Baris ini merupakan sebuah formulir HTML yang menggunakan metode POST dan akan mengirimkan data ke halaman itu sendiri saat tombol "Cetak" ditekan, karena nilai atribut action kosong (`action=""`). Formulir ini memiliki satu input teks (`<input type="text">`) dengan id `jumlah_peserta` yang meminta pengguna untuk memasukkan jumlah peserta. Atribut `min="1"` menetapkan nilai minimum yang dapat dimasukkan ke dalam input tersebut, dan `required` memastikan bahwa pengguna harus mengisi input sebelum mengirimkan formulir. Setelah input, terdapat sebuah tombol "Cetak" (`<button type="submit">`) yang digunakan

untuk mengirimkan formulir. Saat formulir dikirimkan, nilai dari input jumlah peserta akan dikirimkan dalam permintaan HTTP POST.

Pada baris [14-25] Baris ini adalah sebuah skrip PHP yang menangani data yang dikirim melalui metode POST dari sebuah formulir HTML. Ketika sebuah permintaan POST diterima, nilai dari input dengan nama 'jumlah_peserta' diambil dan disimpan dalam variabel `$jumlah_peserta`. Selanjutnya, sebuah perulangan while dilakukan untuk mencetak pesan yang menampilkan urutan peserta dengan mengiterasi dari 1 hingga jumlah peserta yang diinput. Di dalam perulangan, warna teks ditentukan berdasarkan sisa hasil bagi dari nilai `$counter`, dengan menggunakan operator ternary. Jika `$counter` ganjil, warna teks akan merah ('red'), dan jika genap, warna teks akan hijau ('green'). Setiap pesan yang dicetak mengandung nomor peserta dan warna teks yang sesuai. Setelah itu, nilai `$counter` ditingkatkan untuk iterasi berikutnya. Dengan demikian, setiap peserta dipresentasikan dengan nomor urutnya dan warna teks yang bergantian antara merah dan hijau.

SOAL 2

Buatlah program yang dapat menampilkan bentuk segitiga dari gambar yang dimasukkan. Input baris pertama adalah tinggi segitiga. Input kedua adalah alamat 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

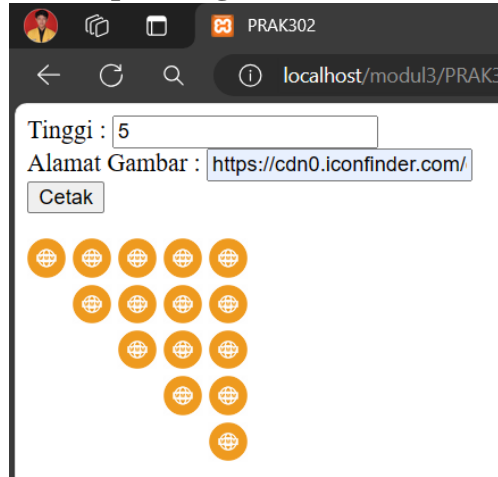
```

1 <html>
2 <head>
3 <title>PRAK302</title>
4 </head>
5 <body>
6 <form action="" method="post">
7 <label for="jumlah">Tinggi :</label>
8 <input type="text" name="jumlah"> <br>
9 <label for="alamat">Alamat Gambar :</label>
10 <input type="text" name="alamat"> <br>
11 <button type="submit"
name="submit">Cetak</button>
12 </form>
13
14 <?php
15 if( isset($_POST['submit']) ) {
16 $max = $_POST['jumlah'];
17 $i = 1; $j = 1; $k = $max;
18 $gambar = $_POST['alamat'];
19 } ?>
20 <?php if( isset($_POST['submit']) ) : ?>
21 <?php while($i <= $max) { ?>
22 <?php while($j < $i ) { ?>
23 <img style="width: 25px; opacity: 0;"
src=<?= "$gambar"; ?> alt="">
24 <?php $j = $j + 1; ?>
25 <?php } ?>
26 <?php while($k >= $i) { ?>
27 <img style="width : 25px" src=<?=
"$gambar"; ?> alt="">
28 <?php $k = $k - 1; ?>
29 <?php } ?>
30 <br>
31 <?php
32 $j = 1; $k = $max;
33 $i = $i + 1;
34 ?>
35 <?php } ?>
36 <?php endif; ?>
37 </body>
38 </html>

```

Tabel 14. Source Code Soal 2

B. Output Program



Gambar 13. Screenshot Hasil Jawaban Soal 2

C. Pembahasan

Pada baris [1] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [2] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [3] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [4] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [5] terdapat `<body>` yang digunakan untuk mendefinisikan isi atau badan suatu dokumen.

Pada baris [6-12] Baris ini adalah sebuah form HTML dengan metode POST yang memiliki dua input field dan satu tombol submit. Pertama, ada input field dengan label "Tinggi : " yang meminta pengguna untuk memasukkan nilai tinggi. Input ini akan dikirimkan sebagai data dengan nama "jumlah" saat formulir dikirimkan. Kemudian, ada input field dengan label "Alamat Gambar : " yang meminta pengguna untuk memasukkan alamat gambar yang ingin dicetak. Input ini akan dikirimkan sebagai data dengan nama "alamat" saat formulir dikirimkan. Terakhir,

ada tombol submit dengan label "Cetak" yang akan mengirimkan formulir saat ditekan dan akan memicu pengiriman data ke alamat yang sama, karena nilai atribut action dikosongkan (""). Semua elemen form ini berada dalam sebuah tag `<form>`, yang berfungsi untuk mengelompokkan elemen-elemen tersebut menjadi satu unit yang akan dikirimkan ke server saat formulir disubmit.

Pada baris [14-19] Baris ini merupakan sebuah blok PHP yang akan dieksekusi jika tombol submit pada suatu form diklik. Pertama, kode memeriksa apakah tombol submit telah ditekan dengan menggunakan fungsi `isset()` pada variabel `$_POST['submit']`. Jika iya, maka variabel `$max` akan diinisialisasi dengan nilai yang diterima dari form dengan nama 'jumlah'. Kemudian, variabel `$i`, `$j`, dan `$k` diinisialisasi dengan nilai 1 dan `$max`. Selanjutnya, variabel `$gambar` diinisialisasi dengan nilai yang diterima dari form dengan nama 'alamat'. Dengan demikian, blok kode ini akan menyiapkan variabel yang diperlukan untuk penggunaan lebih lanjut berdasarkan input yang diterima dari form.

Pada baris [20-36] Baris ini merupakan bagian dari struktur kontrol `if` yang bertujuan untuk mengeksekusi kode di dalamnya jika tombol submit pada formulir telah ditekan. Di dalam blok `if`, terdapat tiga loop `while` bersarang. Pertama, terdapat loop `while ($i <= $max)` yang akan dieksekusi selama nilai `$i` tidak melebihi atau sama dengan nilai `$max`. Di dalam loop ini, terdapat dua loop `while` lagi.

Loop kedua, `while ($j < $i)`, bertujuan untuk mencetak gambar dengan lebar 25px dan opacity 0 (transparan) sebanyak `$i - 1` kali. Variabel `$j` akan bertambah satu setiap iterasi.

Loop ketiga, `while ($k >= $i)`, mencetak gambar dengan lebar 25px sebanyak `$max - $i + 1` kali. Variabel `$k` akan dikurangi satu setiap iterasi.

Setelah kedua loop selesai, sebuah tag `
` ditambahkan untuk membuat baris baru dalam output. Selanjutnya, variabel `$j` diatur kembali menjadi 1, variabel `$k` diatur menjadi `$max`, dan variabel `$i` ditambah satu untuk mempersiapkan iterasi berikutnya.

Dengan demikian, kode ini menghasilkan pola cetak gambar yang berbeda-beda tergantung pada nilai dari `$max` dan mengatur variabel agar sesuai untuk iterasi selanjutnya.

SOAL 3

Buatlah sebuah program untuk mengecek 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

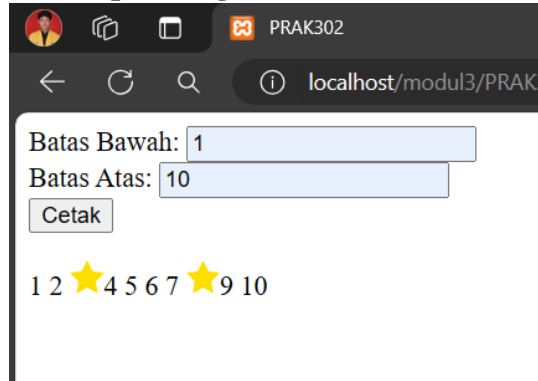
Simpan dengan nama file: PRAK303.php

A. Source Code

```
1 <html>
2 <head>
3   <title>PRAK302</title>
4 </head>
5 <body>
6   <form method="post">
7     Batas Bawah: <input type="text"
name="batas_bawah" required><br>
8     Batas Atas: <input type="text"
name="batas_atas" required><br>
9     <input type="submit" name="submit"
value="Cetak">
10   </form>
11
12   <?php
13     if ($_SERVER["REQUEST_METHOD"] == "POST") {
14       $batas_bawah = $_POST['batas_bawah'];
15       $batas_atas = $_POST['batas_atas'];
16
17       $bilangan = $batas_bawah;
18       do {
19         if (($bilangan + 7) % 5 == 0) {
20           echo '';
21         } else {
22           echo $bilangan . " ";
23         }
24         $bilangan++;
25       } while ($bilangan <= $batas_atas);
26     }
27   ?>
28 </body>
29 </html>
```

Tabel 15. Source Code Soal 3

B. Output Program



Batas Bawah: 1

Batas Atas: 10

Cetak

1 2 ★ 4 5 6 7 ★ 9 10

Gambar 14. Screenshot Hasil Jawaban Soal 3

C. Pembahasan

Pada baris [1] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [2] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [3] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [4] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [5] terdapat `<body>` yang digunakan untuk mendefinisikan isi atau badan suatu dokumen.

Pada baris [6-10] Baris ini merupakan sebuah formulir HTML yang menggunakan metode HTTP POST untuk mengirimkan data ke server. Formulir ini memiliki dua input teks yang meminta pengguna untuk memasukkan "Batas Bawah" dan "Batas Atas". Kedua input ini diberi atribut "required" untuk memastikan bahwa pengguna harus memasukkan nilai sebelum mengirimkan formulir. Terakhir, terdapat sebuah tombol "Cetak" yang bertipe "submit" yang akan mengirimkan data formulir ke server ketika ditekan. Saat formulir dikirim, data yang dimasukkan oleh pengguna akan dikirimkan dalam bentuk pasangan nama-nilai, dengan nama-nama yang ditentukan oleh atribut "name" pada masing-masing input.

Pada baris [12-27] Baris ini merupakan PHP script yang menangani permintaan POST dari sebuah form. Saat form tersebut disubmit, script ini mengambil dua nilai yang dikirimkan melalui form, yaitu batas bawah (`$batas_bawah`) dan batas atas (`$batas_atas`). Kemudian, script melakukan iterasi melalui semua bilangan dari batas bawah hingga batas atas menggunakan loop do-while. Setiap bilangan diuji dengan kondisi, jika hasil dari penambahan bilangan dengan 7 modulo 5 adalah 0 (artinya bilangan tersebut ketika ditambah 7 kemudian dibagi 5 akan menghasilkan sisa 0), maka script akan menampilkan sebuah gambar bintang dengan menggunakan tag ``. Jika tidak memenuhi kondisi tersebut, maka script akan menampilkan bilangan tersebut dengan diikuti spasi. Proses iterasi dilakukan hingga bilangan mencapai atau melebihi batas atas yang telah ditentukan dalam form.

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

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>PRAK304</title>
5  </head>
6  <body>
7      <?php
8          session_start();
9
10         if (!isset($_SESSION['jumlah_bintang'])) {
11             $_SESSION['jumlah_bintang'] = 1;
12         }
13
14         function tampilkanBintang($jumlah) {
15             for ($i = 0; $i < $jumlah; $i++) {
16                 echo      '';
17             }
18         }
19
20         if (isset($_POST['submit'])) {
21             $_SESSION['jumlah_bintang'] =
$_POST['jumlah_bintang'];
22         }
23
24         if (isset($_POST['tambah'])) {
25             $_SESSION['jumlah_bintang']++;
26         }
27
28         if (isset($_POST['kurang']) &&
$_SESSION['jumlah_bintang'] > 1) {
29             $_SESSION['jumlah_bintang']--;
30         }
31     ?>
32
33     <?php
34         if (!isset($_POST['submit']) &&
!isset($_POST['tambah']) &&
35 !isset($_POST['kurang'])) {
36             ?>
37             <form method="post">
38                 <label      for="jumlah_bintang">Jumlah
Bintang:</label>

```

```

39         <input type="text" id="jumlah_bintang"
name="jumlah_bintang" min="1" value="php echo
40 $_SESSION['jumlah_bintang']; ?&gt;"&gt;
41         &lt;button type="submit"
42 name="submit"&gt;Tampilkan&lt;/button&gt;
43     &lt;/form&gt;
44     &lt;?php
45     }
46     ?&gt;

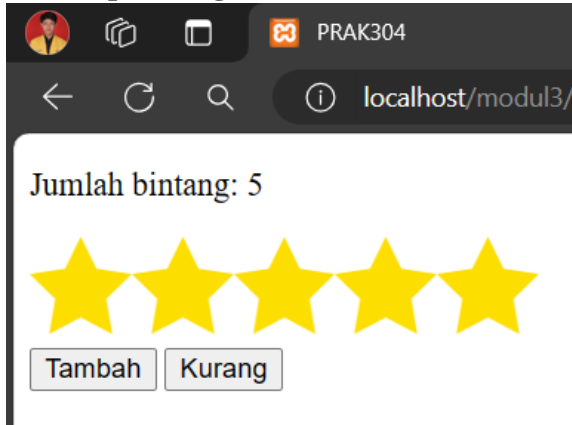
47     &lt;?php
        if (isset($_POST['submit']) ||
48 isset($_POST['tambah']) || isset($_POST['kurang']))
49     {
50         echo "&lt;p&gt;Jumlah bintang: " .
51 $_SESSION['jumlah_bintang'] . "&lt;/p&gt;";
52
53 tampilkanBintang($_SESSION['jumlah_bintang']);
54     }
55     ?&gt;

56     &lt;?php
57     if (isset($_POST['submit']) ||
isset($_POST['tambah']) || isset($_POST['kurang']))
58     {
59         &lt;form method="post"&gt;
60             &lt;button type="submit"
61 name="tambah"&gt;Tambah&lt;/button&gt;
62             &lt;button type="submit"
63 name="kurang"&gt;Kurang&lt;/button&gt;
        &lt;/form&gt;
        &lt;?php
        }
        ?&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre

```

Tabel 16. Source Code Soal 4

B. Output Program



Gambar 15. Screenshot Hasil Jawaban Soal 4

C. Pembahasan

Pada baris [1] terdapat `<!DOCTYPE html>` yang digunakan untuk membangun dan menyusun halaman web.

Pada baris [2] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [3] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [4] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [5] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [6] terdapat `<body>` yang digunakan untuk mendefinisikan isi atau badan suatu dokumen.

Pada baris [7-31] Baris ini merupakan script PHP yang digunakan untuk mengelola jumlah bintang yang ditampilkan dalam sebuah halaman web. Pertama, dilakukan panggilan fungsi `session_start()` untuk memulai sesi PHP. Kemudian, dilakukan pengecekan apakah variabel `$_SESSION['jumlah_bintang']` sudah tersedia dalam sesi. Jika belum, maka variabel tersebut diinisialisasi dengan nilai awal 1. Selanjutnya, terdapat fungsi `tampilkanBintang($jumlah)` yang digunakan untuk menampilkan gambar bintang sesuai dengan jumlah yang diberikan. Fungsi ini menggunakan loop for untuk mencetak gambar bintang sebanyak jumlah yang diinput.

Setelahnya, dilakukan pengecekan terhadap data yang dikirim melalui metode POST. Jika tombol submit ditekan, maka nilai `$_SESSION['jumlah_bintang']` akan diupdate dengan nilai yang baru dari input form. Jika tombol tambah ditekan, nilai `$_SESSION['jumlah_bintang']` akan ditambah 1. Dan jika tombol kurang ditekan dan nilai `$_SESSION['jumlah_bintang']` masih lebih dari 1, maka nilai tersebut akan dikurangi 1.

Pada baris [33-43] Baris ini merupakan kode PHP di atas memeriksa apakah tidak ada tombol submit, tambah, atau kurang yang ditekan. Jika kondisi tersebut terpenuhi, maka baris-baris di dalam blok if akan dieksekusi. Di dalam blok tersebut, sebuah form HTML dengan method post didefinisikan. Form tersebut memiliki sebuah input untuk jumlah bintang dengan tipe text, memiliki ID `jumlah_bintang`, dan nama `jumlah_bintang`. Input tersebut memiliki batas minimum 1 dan nilai awalnya adalah nilai dari variabel `$_SESSION['jumlah_bintang']`. Selanjutnya, terdapat sebuah tombol submit dengan nama `submit`. Jadi, keseluruhan blok tersebut bertujuan untuk menampilkan sebuah form input untuk mengatur jumlah bintang yang akan ditampilkan jika tidak ada aksi form sebelumnya yang dilakukan oleh pengguna.

Pada baris [45-50] Baris ini melakukan pengecekan apakah salah satu tombol telah ditekan. Jika ya, maka blok kode di dalamnya dieksekusi. Dalam blok kode tersebut, terdapat perintah untuk menampilkan jumlah bintang yang disimpan dalam sesi dengan menggunakan tag `<p>` untuk menampilkan paragraf dan mengakses nilai jumlah bintang melalui variabel `$_SESSION['jumlah_bintang']`. Selanjutnya, fungsi `tampilkanBintang()` dipanggil dengan parameter jumlah bintang yang sama untuk menampilkan gambar bintang sesuai dengan jumlahnya. Jadi, jika salah satu tombol ditekan, jumlah bintang dan gambar bintang akan ditampilkan.

Pada baris [52-61] Baris ini merupakan sebuah blok PHP yang memeriksa apakah salah satu dari tombol dengan atribut name `'submit'`, `'tambah'`, atau `'kurang'` ditekan. Jika salah satu tombol tersebut ditekan, maka blok kode tersebut akan mengeksekusi. Di dalam blok tersebut, terdapat sebuah form HTML yang memiliki metode post. Form tersebut memiliki dua tombol, yaitu tombol `'Tambah'` dan tombol `'Kurang'`, yang akan mengirimkan data ke halaman yang sama saat ditekan. Setiap tombol memiliki atribut name yang sesuai dengan kondisi yang diperiksa di dalam blok PHP, yaitu `'tambah'` dan `'kurang'`. Dengan demikian, ketika salah satu tombol ditekan, form ini akan ditampilkan, memberikan opsi kepada pengguna untuk melakukan aksi tambah atau kurang tergantung pada tombol yang ditekan 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 proram:

Contoh 1

Input:

BanDunG

Output:

BbbbbbbAaaaaaaNnnnnnnDddddddUuuuuuuNnnnnnnGgggggg

Contoh 2

Input:

BANJAR

Output:

BbbbbbbAaaaaaNnnnnnJjjjjAaaaaRrrrr

Simpan dengan nama file: PRAK305.php

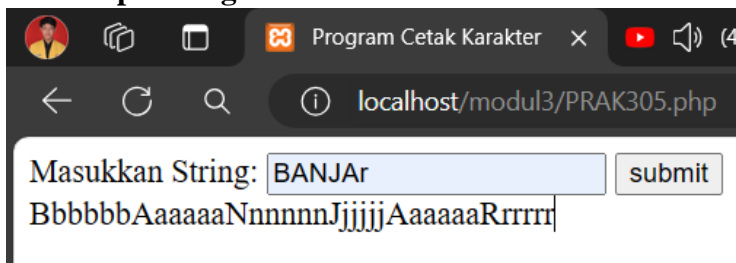
A. Source Code

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Program Cetak Karakter</title>
5 </head>
6 <body>
7     <form method="post">
8         <label                for="input_string">Masukkan
String:</label>
```

9	<code><input type="text" id="input_string"</code>
	<code>name="input_string" required></code>
10	<code><button type="submit"</code>
	<code>name="submit">submit</button></code>
11	<code></form></code>
12	
13	<code><?php</code>
14	<code>function cetakKarakter(\$string) {</code>
15	<code> \$string = strtoupper(\$string);</code>
16	<code> \$panjang = strlen(\$string);</code>
17	
18	<code> for (\$i = 0; \$i < \$panjang; \$i++) {</code>
19	<code> \$karakter = \$string[\$i];</code>
20	<code> echo strtoupper(\$karakter);</code>
21	<code> for (\$j = 1; \$j < \$panjang; \$j++) {</code>
22	<code> echo strtolower(\$karakter);</code>
23	<code> }</code>
24	<code> }</code>
25	<code>}</code>
26	
27	<code>if (isset(\$_POST['submit'])) {</code>
28	<code> \$input_string = \$_POST['input_string'];</code>
29	<code> cetakKarakter(\$input_string);</code>
30	<code>}</code>
31	<code>?></code>
32	<code></body></code>
33	<code></html></code>

Tabel 17. Source Code Soal 5

B. Output Program



Gambar 16. Screenshot Hasil Jawaban Soal 5

C. Pembahasan

Pada baris [1] terdapat `<!DOCTYPE html>` yang digunakan untuk membangun dan menyusun halaman web.

Pada baris [2] terdapat `<html>` yang digunakan untuk mendefinisikan sebuah dokumen html.

Pada baris [3] terdapat `<head>` yang digunakan untuk memberikan informasi tentang suatu dokumen html yang akan digunakan dalam halaman web.

Pada baris [4] terdapat `<title>` yang digunakan untuk mendefinisikan sebuah judul atau nama halaman web yang sedang ditampilkan di browser.

Terdapat juga `</title>` yang digunakan untuk menutup sebuah judul dari dokumen yang ditampilkan dan menandakan bahwa teks yang terdapat di dalam elemen tersebut telah selesai.

Pada baris [5] terdapat `</head>` yang digunakan sebagai sebuah tag penutup pada dokumen html yang mengidentifikasi akhir dari bagian dokumen html.

Pada baris [6] terdapat `<body>` yang digunakan untuk mendefinisikan isi atau badan suatu dokumen.

Pada baris [7-11] Baris ini merupakan sebuah formulir HTML yang menggunakan metode POST untuk mengirimkan data. Di dalam formulir tersebut, terdapat sebuah label yang berisi teks "Masukkan String:", yang diikuti oleh sebuah input teks dengan id `input_string` dan nama `input_string`. Input ini diperlukan (*required*) sehingga pengguna harus memasukkan nilai sebelum formulir dapat dikirimkan. Selain itu, terdapat sebuah tombol submit yang digunakan untuk mengirimkan formulir. Ketika tombol ini ditekan, data yang dimasukkan ke dalam input teks akan dikirimkan ke server untuk diproses lebih lanjut.

Pada baris [13-25] sebuah fungsi PHP bernama `cetakKarakter` yang menerima satu parameter, yaitu `$string`. Pada baris pertama, fungsi ini mengonversi seluruh huruf dalam string input menjadi huruf kapital menggunakan fungsi `strtoupper()`. Selanjutnya, panjang string dihitung dengan menggunakan fungsi `strlen()` dan disimpan dalam variabel `$panjang`. Kemudian, dilakukan iterasi melalui setiap karakter dalam string menggunakan loop `for`. Di dalam loop tersebut, setiap karakter dicetak dalam huruf kapital menggunakan `echo strtoupper($karakter)`. Kemudian, terdapat loop nested lainnya, dimulai dari `$j = 1`, yang mencetak karakter yang sama dalam huruf kecil menggunakan `echo strtolower($karakter)` untuk setiap karakter dalam string. Ini berarti setiap karakter akan dicetak sebanyak panjang string yang sesuai.

Pada baris [27-31] Baris ini memeriksa apakah tombol submit pada form telah ditekan. Jika ya, maka kontrol beralih ke dalam blok `if`. Di dalam blok `if`, nilai dari input string yang dikirim melalui form disimpan ke dalam variabel `$input_string` menggunakan `$_POST['input_string']`. Selanjutnya, fungsi

`cetakKarakter($input_string)` dipanggil dengan parameter string input sebagai argumen. Ini mengarah pada eksekusi fungsi `cetakKarakter`, yang akan mencetak karakter-karakter sesuai dengan logika yang didefinisikan di dalamnya.

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 yang 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 output 2:

Panjang :

Lebar :

Nilai :

Panjang nilai tidak sesuai dengan ukuran matriks

Simpan dengan nama file: PRAK401.php

A. Source Code

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-
width, initial-scale=1.0">
6      <title>PRAK401</title>
7      <style>
8          table {
9              border-collapse: collapse;
10             margin-top: 10px;
11         }
12         td {
13             padding: 5px;
14             border: 1px solid black;
15             text-align: center;
16             font-size: 15px;
17             width: 30px;
18             height: 30px;
19         }
20     </style>
21 </head>
22 <body>
23     <form method="post">
24         Panjang : <input type="text" name="panjang"
required><br>
25         Lebar : <input type="text" name="lebar"
required><br>
26         Nilai : <input type="text" name="nilai"
required><br>
27         <button type="submit"
name="cetak">Cetak</button>
28     </form>
29
30     <?php
31     if (isset($_POST['cetak'])) {
32         $panjang = $_POST['panjang'];
33         $lebar = $_POST['lebar'];
34         $nilai = $_POST['nilai'];
35
36         $values = explode(" ", $nilai);
37         $totalValues = count($values);
38
39         if ($totalValues != $panjang * $lebar) {
40

```

```

41         echo "<p>Panjang nilai tidak sesuai
42 dengan ukuran matriks</p>";
43     } else {
44         echo "<table>";
45         for ($i = 0; $i < $panjang; $i++) {
46             echo "<tr>";
47             for ($j = 0; $j < $lebar; $j++) {
48                 echo "<td>" . $values[$i * $lebar
49 + $j] . "</td>";
50             }
51             echo "</tr>";
52         }
53         echo "</table>";
54     }
55     ?>
</body>
</html>

```

Tabel 18. Source Code Soal 1

B. Output Program

Panjang :

Lebar :

Nilai :

34	56	41
45	36	37
51	32	46

Gambar 17 Screenshot Hasil Jawaban Soal 1 Output 1

Panjang :

Lebar :

Nilai :

Panjang nilai tidak sesuai dengan ukuran matriks

Gambar 18 Screenshot Hasil Jawaban Soal 1 Output 2

C. Pembahasan

Kode ini adalah sebuah halaman HTML yang memiliki form untuk memasukkan panjang, lebar, dan nilai-nilai matriks. Setelah pengguna mengisi form dan mengklik tombol "Cetak", nilai-nilai matriks akan ditampilkan dalam bentuk tabel HTML.

Pada bagian PHP, setelah form dikirimkan, nilai-nilai matriks dipisahkan menjadi array menggunakan fungsi `explode()`, dengan pemisah berupa spasi. Selanjutnya, dilakukan perhitungan untuk memastikan bahwa jumlah nilai dalam matriks sesuai dengan panjang dan lebar matriks yang diinginkan. Jika jumlah nilai tidak sesuai, pesan kesalahan akan ditampilkan. Jika jumlah nilai sesuai, nilai-nilai tersebut akan ditampilkan dalam bentuk tabel HTML.

Pada tabel HTML, setiap nilai dalam matriks diposisikan dalam sel-sel tabel. Dengan menggunakan loop bersarang, nilai-nilai tersebut diatur dalam baris dan kolom yang sesuai dengan panjang dan lebar matriks yang dimasukkan pengguna sebelumnya. Setiap baris matriks direpresentasikan sebagai satu baris dalam tabel HTML, sedangkan setiap nilai dalam baris tersebut direpresentasikan sebagai satu sel tabel.

Selain itu, terdapat validasi panjang nilai matriks dengan ukuran matriks yang ditentukan, sehingga memberikan pesan kesalahan jika jumlah nilai yang dimasukkan tidak sesuai dengan ukuran matriks yang diinginkan. Hal ini membantu memastikan bahwa matriks yang ditampilkan dalam tabel memiliki dimensi yang konsisten dan sesuai dengan yang diharapkan.

SOAL 2

Diberikan 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 berikut:

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

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

Sehingga didapatkan output seperti berikut

Nama	NIM	Nilai UTS	Nilai UAS	Nilai Akhir	Huruf
Andi	2101001	87	87	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

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-
width, initial-scale=1.0">
6      <title>Student Scores</title>
7      <style>
8          table {
9              border-collapse: collapse;
10             width: 40%;
11             margin-top: 20px;
12         }
13         th, td {
14             padding: 8px;
15             border: 1px solid black;
16             text-align: left;
17             font-size: 14px;
18         }
19         th {
20             background-color: #BFBEBE;
21         }
22     </style>
23 </head>
24 <body>
25     <?php
26         $students = [
27             ["Nama" => "Andi", "NIM" => "2101001", "Nilai
UTS" => 87, "Nilai UAS" => 65],
28             ["Nama" => "Budi", "NIM" => "2101002", "Nilai
UTS" => 76, "Nilai UAS" => 79],
29             ["Nama" => "Tono", "NIM" => "2101003", "Nilai
UTS" => 50, "Nilai UAS" => 41],
30             ["Nama" => "Jessica", "NIM" => "2101004",
"Nilai UTS" => 60, "Nilai UAS" => 75],
31         ];
32
33         function calculateFinalScore($uts, $uas) {
34             return ($uts * 0.4) + ($uas * 0.6);
35         }
36
37         function getGrade($finalScore) {
38             if ($finalScore >= 80) {
39                 return "A";
40             } elseif ($finalScore >= 70) {

```

```

41         return "B";
42     } elseif ($finalScore >= 60) {
43         return "C";
44     } elseif ($finalScore >= 50) {
45         return "D";
46     } else {
47         return "E";
48     }
49 }
50
51 foreach ($students as $key => $student) {
52     $finalScore =
53     calculateFinalScore($student["Nilai UTS"],
54     $student["Nilai UAS"]);
55     $students[$key]["Nilai Akhir"] =
56     $finalScore;
57     $students[$key]["Huruf"] =
58     getGrade($finalScore);
59 }
60
61 <table>
62     <thead>
63         <tr>
64             <th>Nama</th>
65             <th>NIM</th>
66             <th>Nilai UTS</th>
67             <th>Nilai UAS</th>
68             <th>Nilai Akhir</th>
69             <th>Huruf</th>
70         </tr>
71     </thead>
72     <tbody>
73         <?php foreach ($students as $student): ?>
74             <tr>
75                 <td><?php echo $student["Nama"];
76                 ?></td>
77                 <td><?php echo $student["NIM"];
78                 ?></td>
79                 <td><?php echo $student["Nilai
80                 UTS"]; ?></td>
81                 <td><?php echo $student["Nilai
82                 UAS"]; ?></td>
83                 <td><?php
84                 echo
85                 number_format($student["Nilai Akhir"], 1); ?></td>

```

79	<code><td><?php</code>	<code>echo</code>
80	<code>\$student["Huruf"]; ?></td></code>	
81	<code></tr></code>	
82	<code><?php endforeach; ?></code>	
83	<code></tbody></code>	
	<code></table></code>	
	<code></body></code>	
	<code></html></code>	

Tabel 19. Source Code Soal 2

B. Output Program

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	69.0	C

Gambar 19 Screenshot Hasil Jawaban Soal 2

C. Pembahasan

Kode ini merupakan halaman HTML yang menampilkan daftar nilai mahasiswa beserta nilai akhir dan huruf yang sesuai dengan skema penilaian yang ditentukan. Data mahasiswa disimpan dalam bentuk array asosiatif di dalam variabel `$students`, di mana setiap elemen array mewakili satu mahasiswa dengan informasi nama, NIM, nilai UTS, dan nilai UAS.

Ada dua buah fungsi yang digunakan dalam kode tersebut. Pertama, fungsi `calculateFinalScore()` untuk menghitung nilai akhir mahasiswa berdasarkan bobot nilai UTS (40%) dan nilai UAS (60%). Kedua, fungsi `getGrade()` untuk menentukan huruf nilai berdasarkan nilai akhir yang didapat.

Selanjutnya, dilakukan iterasi pada array `$students` menggunakan loop `foreach`. Pada setiap iterasi, nilai akhir dihitung dengan memanggil fungsi `calculateFinalScore()` dan disimpan kembali ke dalam array asosiatif `$students` bersama dengan nilai huruf yang sesuai, menggunakan fungsi `getGrade()`.

Hasil dari perhitungan tersebut kemudian ditampilkan dalam sebuah tabel HTML. Pada bagian header tabel, ditampilkan judul kolom sesuai dengan informasi yang ditampilkan, yaitu Nama, NIM, Nilai UTS, Nilai UAS, Nilai Akhir, dan Huruf.

Sedangkan pada bagian body tabel, setiap baris mewakili satu mahasiswa dengan kolom-kolom yang berisi informasi sesuai dengan data yang disimpan dalam array `$students`.

Tampilan tabel diatur dengan menggunakan CSS yang terdapat di dalam tag `<style>`, seperti setting border-collapse, warna latar belakang header tabel, dan lainnya.

SOAL 3

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

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-
width, initial-scale=1.0">
6      <title>Course Load</title>
7      <style>
8          table {
9              border-collapse: collapse;
10             width: 70%;
11             margin-top: 20px;
12         }
13         th, td {
14             padding: 8px;
15             border: 1px solid black;
16             text-align: left;
17             font-size: 14px;
18         }
19         th {
20             background-color: #e0e0e0;
21         }
22         .revisi {
23             background-color: red;
24             color: white;
25         }
26         .tidak-revisi {
27             background-color: green;
28             color: white;
29         }
30     </style>
31 </head>
32 <body>
33 <?php
34 $students = [
35     [
36         "No" => 1,
37         "Nama" => "Ridho",
38         "Mata Kuliah" => [
39             ["nama" => "Pemrograman I", "SKS" => 2],
40             ["nama" => "Praktikum Pemrograman I",
"SKS" => 1],
41             ["nama" => "Pengantar Lingkungan Lahan
Basah", "SKS" => 2],
42

```

```

43         ["nama" => "Arsitektur Komputer", "SKS"
=> 3],
44     ]
45 ],
46 [
47     "No" => 2,
48     "Nama" => "Ratna",
49     "Mata Kuliah" => [
50         ["nama" => "Basis Data I", "SKS" => 2],
51         ["nama" => "Praktikum Basis Data I",
"SKS" => 1],
52         ["nama" => "Kalkulus", "SKS" => 3],
53     ]
54 ],
55 [
56     "No" => 3,
57     "Nama" => "Tono",
58     "Mata Kuliah" => [
59         ["nama" => "Rekayasa Perangkat Lunak",
"SKS" => 3],
60         ["nama" => "Analisis dan Perancangan
Sistem", "SKS" => 3],
61         ["nama" => "Komputasi Awan", "SKS" =>
3],
62         ["nama" => "Koorporasi Basis Data",
"SKS" => 3],
63     ]
64 ],
65 ],
66 ];
67
68 function calculateTotalSKS($courses) {
69     $total = 0;
70     foreach ($courses as $course) {
71         $total += $course["SKS"];
72     }
73     return $total;
74 }
75
76 foreach ($students as $key => $student) {
77     $totalSKS = calculateTotalSKS($student["Mata
Kuliah"]);
78     $students[$key]["Total SKS"] = $totalSKS;
79     $students[$key]["Keterangan"] = $totalSKS < 7 ?
"Revisi KRS" : "Tidak Revisi";
80 }

```

```

81  ?>
82
83  <table>
84      <thead>
85          <tr>
86              <th>No</th>
87              <th>Nama</th>
88              <th>Mata Kuliah diambil</th>
89              <th>SKS</th>
90              <th>Total SKS</th>
91              <th>Keterangan</th>
92          </tr>
93      </thead>
94      <tbody>
95          <?php foreach ($students as $student): ?>
96              <tr>
97                  <td
98                      <?php echo
count($student["Mata Kuliah"]); ?>"><?php echo
99                      $student["No"]; ?></td>
100                  <td
101                      <?php echo
count($student["Mata Kuliah"]); ?>"><?php echo
102                      $student["Nama"]; ?></td>
103                      <?php foreach ($student["Mata Kuliah"]
as $index => $course): ?>
104                          <?php if ($index == 0): ?>
105                              <td><?php echo $course["nama"];
?></td>
106                              <td><?php echo $course["SKS"];
?></td>
107                              <td
108                                  <?php echo
count($student["Mata Kuliah"]); ?>"><?php echo
109                                  $student["Total SKS"]; ?></td>
110                                  <td
111                                      <?php echo
count($student["Mata Kuliah"]); ?>" class="<?php
112                                      echo $student["Keterangan"] == 'Revisi KRS' ?
113

```

114	'revisi' : 'tidak-revisi'; ?>"<?php echo
115	\$student["Keterangan"]; ?></td>
	<?php else: ?>
	</tr><tr>
	<td><?php echo \$course["nama"];
	?></td>
	<td><?php echo \$course["SKS"];
	?></td>
	<?php endif; ?>
	<?php endforeach; ?>
	</tr>
	<?php endforeach; ?>
	</tbody>
	</table>
	</body>
	</html>

Tabel 20. Source Code Soal 3

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		
		Korporasi Basis Data	3		

Gambar 20 Screenshot Hasil Jawaban Soal 3

C. Pembahasan

Kode di atas adalah sebuah halaman HTML yang menampilkan daftar mahasiswa beserta mata kuliah yang diambil oleh masing-masing mahasiswa beserta detail SKS (Satuan Kredit Semester) yang diperoleh. Setiap mahasiswa dapat mengambil beberapa mata kuliah, yang direpresentasikan dalam tabel dengan kolom-kolom No, Nama, Mata Kuliah diambil, SKS, Total SKS, dan Keterangan.

Data mahasiswa dan mata kuliah disimpan dalam bentuk array asosiatif di dalam variabel \$students. Setiap elemen array mewakili satu mahasiswa dengan

informasi seperti nomor, nama, dan daftar mata kuliah yang diambil. Setiap mata kuliah memiliki atribut nama dan SKS.

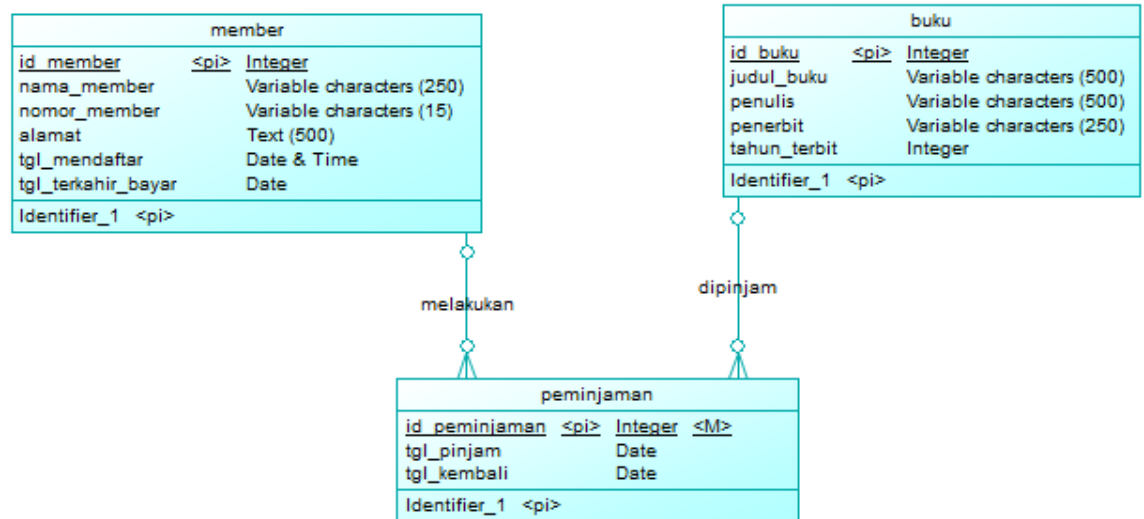
Pada bagian PHP, terdapat fungsi `calculateTotalSKS()` yang digunakan untuk menghitung total SKS dari semua mata kuliah yang diambil oleh seorang mahasiswa. Setelah itu, dilakukan iterasi pada array `$students` menggunakan loop `foreach`. Pada setiap iterasi, total SKS dihitung dan disimpan kembali ke dalam array `$students` bersama dengan keterangan apakah total SKS kurang dari 7 (yang menandakan perlu revisi KRS) atau tidak.

Hasil perhitungan tersebut kemudian ditampilkan dalam sebuah tabel HTML. Setiap mahasiswa direpresentasikan dalam sebuah baris tabel, dengan menggunakan atribut `rowspan` pada kolom No dan Nama agar informasi tersebut hanya ditampilkan sekali untuk setiap mahasiswa. Mata kuliah yang diambil oleh mahasiswa juga ditampilkan dalam baris-baris yang terpisah, dengan menggunakan loop `foreach` bersarang. Selain itu, kolom Keterangan diberi warna latar belakang merah jika mahasiswa memerlukan revisi KRS, dan hijau jika tidak. Ini ditentukan melalui pengecekan pada atribut Keterangan pada setiap mahasiswa.

MODUL 5: FUNCTION DAN DATABASE

SOAL

- 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:

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

```
1 <?php
2 class Koneksi {
3     private $host = "localhost";
4     private $dbname = "perpustakaan";
5     private $username = "root";
6     private $password = "";
7     public $conn;
8
9     public function getConnection() {
10         $this->conn = null;
11
12         try {
13             $this->conn = new PDO("mysql:host=" .
14 $this->host . ";dbname=" . $this->dbname, $this-
15 >username, $this->password);
16             $this->conn->exec("set names utf8");
17         } catch(PDOException $exception) {
18             echo "Connection error: " . $exception-
19 >getMessage();
20         }
21     }
22     return $this->conn;
23 }
```

Tabel 21. Source Code Koneksi.php

```

1  <?php
2  include_once 'Koneksi.php';
3
4  class Model {
5      private $conn;
6
7      public function __construct() {
8          $database = new Koneksi();
9          $this->conn = $database->getConnection();
10     }
11
12     private function executeQuery($query, $params =
13     []) {
14         $stmt = $this->conn->prepare($query);
15         foreach ($params as $param => $value) {
16             $stmt->bindValue($param, $value);
17         }
18         return $stmt->execute();
19     }
20
21     public function getMembers() {
22         $query = "SELECT * FROM member";
23         $stmt = $this->conn->prepare($query);
24         $stmt->execute();
25         return $stmt->fetchAll(PDO::FETCH_ASSOC);
26     }
27
28     public function addMember($nama_member,
29     $nomor_member, $tgl_mendaftar, $tgl_terakhir_bayar)
30     {
31         $query = "INSERT INTO member (nama_member,
32     nomor_member, tgl_mendaftar, tgl_terakhir_bayar)
33
34
35
36
37
38

```

	VALUES (:nama_member, :nomor_member, :tgl_mendaftar, :tgl_terakhir_bayar)";
	return \$this->executeQuery(\$query, [
39	':nama_member' => \$nama_member,
40	':nomor_member' => \$nomor_member,
41	':tgl_mendaftar' => \$tgl_mendaftar,
42	':tgl_terakhir_bayar' =>
43	\$tgl_terakhir_bayar
44]);
45	}
46	
47	public function updateMember(\$id_member,
48	\$nama_member, \$nomor_member, \$tgl_mendaftar,
49	\$tgl_terakhir_bayar) {
	\$query = "UPDATE member SET nama_member =
50	:nama_member, nomor_member = :nomor_member,
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	

```

73     $tgl_mendaftar = :tgl_mendaftar, tgl_terakhir_bayar =
74     :tgl_terakhir_bayar WHERE id_member = :id_member";
75     return $this->executeQuery($query, [
76         ':id_member' => $id_member,
77         ':nama_member' => $nama_member,
78         ':nomor_member' => $nomor_member,
79         ':tgl_mendaftar' => $tgl_mendaftar,
80         ':tgl_terakhir_bayar' =>
81         $tgl_terakhir_bayar
82     ]);
83     }
84     public function deleteMember($id_member) {
85         $query = "DELETE FROM member WHERE id_member
86         = :id_member";
87         return $this->executeQuery($query,
88         [':id_member' => $id_member]);
89     }
90     public function getBooks() {
91         $query = "SELECT * FROM buku";
92         $stmt = $this->conn->prepare($query);
93         $stmt->execute();
94         return $stmt->fetchAll(PDO::FETCH_ASSOC);
95     }
96     public function addBook($judul_buku, $penulis,
97     $tahun_terbit) {
98         $query = "INSERT INTO buku (judul_buku,
99     penulis, tahun_terbit) VALUES (:judul_buku,
100     :penulis, :tahun_terbit)";
101         return $this->executeQuery($query, [
102             ':judul_buku' => $judul_buku,
103             ':penulis' => $penulis,
104             ':tahun_terbit' => $tahun_terbit
105         ]);
106     }
107     public function updateBook($id_buku,
108     $judul_buku, $penulis, $tahun_terbit) {
109         $query = "UPDATE buku SET judul_buku =
110     :judul_buku, penulis = :penulis, tahun_terbit =
111     :tahun_terbit WHERE id_buku = :id_buku";
112         return $this->executeQuery($query, [
113             ':id_buku' => $id_buku,

```

```

109         ':judul_buku' => $judul_buku,
110         ':penulis' => $penulis,
111         ':tahun_terbit' => $tahun_terbit
112     });
113     }

114     public function deleteBook($id_buku) {
115         $query = "DELETE FROM buku WHERE id_buku =
116 :id_buku";
117         return $this->executeQuery($query,
118 [':id_buku' => $id_buku]);
119     }

    public function getLoans() {
        $query = "SELECT * FROM peminjaman";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }

    public function addLoan($id_member, $id_buku,
$ttl_pinjam, $ttl_kembali) {
        $query = "INSERT INTO peminjaman (id_member,
id_buku,      ttl_pinjam,      ttl_kembali) VALUES
(:id_member, :id_buku, :ttl_pinjam, :ttl_kembali)";
        return $this->executeQuery($query, [
            ':id_member' => $id_member,
            ':id_buku' => $id_buku,
            ':ttl_pinjam' => $ttl_pinjam,
            ':ttl_kembali' => $ttl_kembali
        ]);
    }

    public function updateLoan($id_peminjaman,
$id_member, $id_buku, $ttl_pinjam, $ttl_kembali) {
        $query = "UPDATE peminjaman SET id_member =
:id member,  id buku  =  :id buku,  ttl pinjam  =

```

	<pre> :tgl_pinjam, tgl_kembali = :tgl_kembali WHERE id_peminjaman = :id_peminjaman"; return \$this->executeQuery(\$query, [':id_peminjaman' => \$id_peminjaman, ':id_member' => \$id_member, ':id_buku' => \$id_buku, ':tgl_pinjam' => \$tgl_pinjam, ':tgl_kembali' => \$tgl_kembali]); } public function deleteLoan(\$id_peminjaman) { \$query = "DELETE FROM peminjaman WHERE id_peminjaman = :id_peminjaman"; return \$this->executeQuery(\$query, [':id_peminjaman' => \$id_peminjaman]); } } ?> </pre>
--	---

Tabel 22. Source Code Model.php

```

1  <?php
2  include_once 'Model.php';
3  $model = new Model();
4  $members = $model->getMembers();
5  ?>
6
7  <!DOCTYPE html>
8  <html>
9  <head>
10     <title>Member</title>
11 </head>
12 <body>
13     <h1>Data Member</h1>
14     <table border="1">
15         <tr>
16             <th>ID Member</th>
17             <th>Nama Member</th>
18             <th>Nomor Member</th>
19             <th>Tanggal Mendaftar</th>
20             <th>Tanggal Terakhir Bayar</th>
21             <th>Aksi</th>
22         </tr>
23         <?php foreach ($members as $member) { ?>
24             <tr>
25                 <td><?php echo $member['id_member'];
26                 ?></td>
27                 <td><?php echo
28                 $member['nama_member']; ?></td>
29                 <td><?php echo
30                 $member['nomor_member']; ?></td>
31                 <td><?php echo
32                 $member['tgl_mendaftar']; ?></td>
33                 <td><?php echo
34                 $member['tgl_terakhir_bayar']; ?></td>
35                 <td>
36                     <a href="FormMember.php?id=<?php
37                     echo $member['id_member']; ?>">Edit</a>
38                     <a
39                     href="deleteMember.php?id=<?php
40                     echo

```

38	\$member['id_member']; ?>" onclick="return
39	confirm('Yakin ingin menghapus data?')">Hapus
40	</td>
	</tr>
	<?php } ?>
	</table>
	Tambah Member
	</body>
	</html>

Tabel 23. Source Code Member.php

1	<?php
2	include_once 'Model.php';
3	\$model = new Model();
4	
5	\$id_member = '';
6	\$nama_member = '';
7	\$nomor_member = '';
8	\$tgl_mendaftar = '';
9	\$tgl_terakhir_bayar = '';
10	
11	if (isset(\$_GET['id'])) {
12	\$id_member = \$_GET['id'];
13	\$member = \$model->getMemberById(\$id_member);
14	if (\$member) {
15	\$nama_member = \$member['nama_member'];
16	\$nomor_member = \$member['nomor_member'];
17	\$tgl_mendaftar
18	\$member['tgl_mendaftar'];
	\$tgl_terakhir_bayar
19	\$member['tgl_terakhir_bayar'];
20	}
21	}
22	
23	if (\$_POST) {
24	if (\$id_member) {
	\$model->
	updateMember(\$_POST['id_member'],
25	\$_POST['nama_member'], \$_POST['nomor_member'],
26	
27	
28	
29	


```

30 $_POST['tgl_mendaftar'],
31 $_POST['tgl_terakhir_bayar']);
32     } else {
33         $model->addMember($_POST['nama_member'],
34 $_POST['nomor_member'], $_POST['tgl_mendaftar'],
35 $_POST['tgl_terakhir_bayar']);
36     }
37     header('Location: Member.php');
38 }
39 ?>
40 <!DOCTYPE html>
41 <html>
42 <head>
43     <title>Form Member</title>
44 </head>
45 <body>
46     <h1><?php echo $id_member ? 'Edit' : 'Tambah'; ?>
47 Member</h1>
48     <form method="post">
49         <input type="hidden" name="id_member"
50 value="<?php echo $id_member; ?>">
51         <label>Nama Member:</label>
52         <input type="text" name="nama_member"
53 value="<?php echo $nama_member; ?>" required><br>
54         <label>Nomor Member:</label>
55         <input type="text" name="nomor_member"
56 value="<?php echo $nomor_member; ?>" required><br>
57         <label>Tanggal Mendaftar:</label>
58         <input type="date" name="tgl_mendaftar"
59 value="<?php echo $tgl_mendaftar; ?>" required><br>
60         <label>Tanggal Terakhir Bayar:</label>
61         <input type="date" name="tgl_terakhir_bayar"
62 value="<?php echo $tgl_terakhir_bayar; ?>"
63 required><br>
64         <input type="submit" value="Simpan">
65     </form>
66     <br>
67     <a href="Member.php">Kembali</a>
68 </body>
69 </html>

```

Tabel 24. Source Code FormMember.php

```

1  <?php
2  include_once 'Model.php';
3  $model = new Model();
4  $books = $model->getBooks();
5  ?>
6
7  <!DOCTYPE html>
8  <html>
9  <head>
10     <title>Buku</title>
11 </head>
12 <body>
13     <h1>Data Buku</h1>
14     <table border="1">
15         <tr>
16             <th>ID Buku</th>
17             <th>Judul Buku</th>
18             <th>Penulis</th>
19             <th>Tahun Terbit</th>
20             <th>Aksi</th>
21         </tr>
22         <?php foreach ($books as $book) { ?>
23             <tr>
24                 <td><?php echo $book['id_buku'];
25 ?></td>
26                 <td><?php echo $book['judul_buku'];
27 ?></td>
28                 <td><?php echo $book['penulis'];
29 ?></td>
30                 <td><?php echo
$book['tahun_terbit']; ?></td>
31                 <td>
32                     <a href="FormBuku.php?id=<?php
echo $book['id_buku']; ?>">Edit</a>
33                     <a href="deleteBuku.php?id=<?php
echo $book['id_buku']; ?>" onclick="return
34 confirm('Yakin ingin menghapus data?')">Hapus</a>
35                 </td>
36             </tr>
37         <?php } ?>
38     </table>
39     <br>
40     <a href="FormBuku.php">Tambah Buku</a>
41 </body>
42 </html>

```

Tabel 25. Source Code Buku.php

```

1  <?php
2      include_once 'Model.php';
3      $model = new Model();
4
5      $id_buku = '';
6      $judul_buku = '';
7      $penulis = '';
8      $tahun_terbit = '';
9
10     if (isset($_GET['id'])) {
11         $id_buku = $_GET['id'];
12         $book = $model->getBookById($id_buku);
13         if ($book) {
14             $judul_buku = $book['judul_buku'];
15             $penulis = $book['penulis'];
16             $tahun_terbit = $book['tahun_terbit'];
17         }
18     }
19
20     if ($_POST) {
21         if ($id_buku) {
22             $model->updateBook($_POST['id_buku'],
23 $_POST['judul_buku'],
24 $_POST['penulis'],
25 $_POST['tahun_terbit']);
26         } else {
27             $model->addBook($_POST['judul_buku'],
28 $_POST['penulis'], $_POST['tahun_terbit']);
29         }
30         header('Location: Buku.php');
31     }
32 ?>
33
34 <!DOCTYPE html>
35 <html>
36 <head>
37     <title>Form Buku</title>
38 </head>
39 <body>
40     <h1><?php echo $id_buku ? 'Edit' : 'Tambah'; ?>
41     Buku</h1>
42     <form method="post">
43         <input type="hidden" name="id_buku"
44 value="<?php echo $id_buku; ?>">
45         <label>Judul Buku:</label>

```

41	<code><input type="text" name="judul_buku"</code>
42	<code>value="<?php echo \$judul_buku; ?>" required>
</code>
43	<code><label>Penulis:</label></code>
44	<code><input type="text" name="penulis"</code>
45	<code>value="<?php echo \$penulis; ?>" required>
</code>
46	<code><label>Tahun Terbit:</label></code>
47	<code><input type="text" name="tahun_terbit"</code>
48	<code>value="<?php echo \$tahun_terbit; ?>" required>
</code>
49	<code><input type="submit" value="Simpan"></code>
50	<code></form></code>
	<code>
</code>
	<code>Kembali</code>
	<code></body></code>
	<code></html></code>

Tabel 26. Source Code FormBuku.php

```

1  <?php
2  include_once 'Model.php';
3  $model = new Model();
4  $loans = $model->getLoans();
5  ?>
6
7  <!DOCTYPE html>
8  <html>
9  <head>
10     <title>Peminjaman</title>
11 </head>
12 <body>
13     <h1>Data Peminjaman</h1>
14     <table border="1">
15         <tr>
16             <th>ID Peminjaman</th>
17             <th>ID Member</th>
18             <th>ID Buku</th>
19             <th>Tanggal Pinjam</th>
20             <th>Tanggal Kembali</th>
21             <th>Aksi</th>
22         </tr>
23         <?php foreach ($loans as $loan) { ?>
24             <tr>
25                 <td><?php echo
$loan['id_peminjaman']; ?></td>
26                 <td><?php echo $loan['id_member'];
?></td>
27                 <td><?php echo $loan['id_buku'];
28                 ?></td>
                <td><?php echo $loan['tgl_pinjam'];
                ?></td>
29                 <td><?php echo $loan['tgl_kembali'];
30                 ?></td>
31                 <td>
                    <a
32 href="FormPeminjaman.php?id=<?php echo
$loan['id_peminjaman']; ?>">Edit</a>
                    <a
33 href="deletePeminjaman.php?id=<?php echo
34
35
36
37

```

38	\$loan['id_peminjaman']; ?>" onclick="return
39	confirm('Yakin ingin menghapus data?')">Hapus
40	</td>
	</tr>
	<?php } ?>
	</table>
	Tambah
	Peminjaman
	</body>
	</html>

Tabel 27. Source Code Peminjaman.php

1	<?php
2	include_once 'Model.php';
3	\$model = new Model();
4	
5	\$id_peminjaman = '';
6	\$id_member = '';
7	\$id_buku = '';
8	\$tgl_pinjam = '';
9	\$tgl_kembali = '';
10	
11	if (isset(\$_GET['id'])) {
12	\$id_peminjaman = \$_GET['id'];
13	\$loan = \$model->getLoanById(\$id_peminjaman);
14	if (\$loan) {
15	\$id_member = \$loan['id_member'];
16	\$id_buku = \$loan['id_buku'];
17	\$tgl_pinjam = \$loan['tgl_pinjam'];
18	\$tgl_kembali = \$loan['tgl_kembali'];
19	}
20	}
21	
22	if (\$_POST) {
23	if (\$id_peminjaman) {
24	\$model->
	updateLoan(\$_POST['id_peminjaman'],
25	
26	
27	
28	
29	

```

30 $_POST['id_member'],          $_POST['id_buku'],
31 $_POST['tgl_pinjam'], $_POST['tgl_kembali']);
32     } else {
33         $model->addLoan($_POST['id_member'],
34 $_POST['id_buku'],          $_POST['tgl_pinjam'],
35 $_POST['tgl_kembali']);
36     }
37     header('Location: Peminjaman.php');
38 }
39 ?>
40 <!DOCTYPE html>
41 <html>
42 <head>
43     <title>Form Peminjaman</title>
44 </head>
45 <body>
46     <h1><?php echo $id_peminjaman ? 'Edit' :
47 'Tambah'; ?> Peminjaman</h1>
48     <form method="post">
49         <input type="hidden" name="id_peminjaman"
50 value="<?php echo $id_peminjaman; ?>">
51         <label>ID Member:</label>
52         <input type="number" name="id_member"
53 value="<?php echo $id_member; ?>" required><br>
54         <label>ID Buku:</label>
55         <input type="number" name="id_buku"
56 value="<?php echo $id_buku; ?>" required><br>
57         <label>Tanggal Pinjam:</label>
58         <input type="date" name="tgl_pinjam"
59 value="<?php echo $tgl_pinjam; ?>" required><br>
60         <label>Tanggal Kembali:</label>
61         <input type="date" name="tgl_kembali"
62 value="<?php echo $tgl_kembali; ?>" required><br>
63         <input type="submit" value="Simpan">
64     </form>
65     <br>
66     <a href="Peminjaman.php">Kembali</a>
67 </body>
68 </html>

```

Tabel 28. Source Code FormPeminjaman.php

B. Output Program

Data Member

ID Member	Nama Member	Nomor Member	Tanggal Mendaftar	Tanggal Terakhir Bayar	Aksi
1	eva	5	2024-05-28	2024-06-28	Edit Hapus

[Tambah Member](#)

Gambar 21. Screenshot Hasil Jawaban Member.php

Tambah Member

Nama Member:

Nomor Member:

Tanggal Mendaftar:

Tanggal Terakhir Bayar:

[Kembali](#)

Gambar 22. Screenshot Hasil Jawaban FormMember.php

Data Buku

ID Buku	Judul Buku	Penulis	Tahun Terbit	Aksi
1	Investasi	adit	2024	Edit Hapus

[Tambah Buku](#)

Gambar 23. Screenshot Hasil Jawaban Buku.php

Tambah Buku

Judul Buku:

Penulis:

Tahun Terbit:

[Kembali](#)

Gambar 24. Screenshot Hasil Jawaban FormBuku.php

Data Peminjaman

ID Peminjaman	ID Member	ID Buku	Tanggal Pinjam	Tanggal Kembali	Aksi
1	1	1	2024-05-28	2024-06-28	Edit Hapus

[Tambah Peminjaman](#)

Gambar 25. Screenshot Hasil Jawaban Peminjaman.php

Tambah Peminjaman

ID Member:

ID Buku:

Tanggal Pinjam:

Tanggal Kembali:

[Kembali](#)

Gambar 26. Screenshot Hasil Jawaban FormPeminjaman.php

C. Pembahasan

File Koneksi.php

Kode PHP ini mendefinisikan sebuah kelas bernama Koneksi yang digunakan untuk menghubungkan ke database MySQL. Pada awalnya, empat properti privat (`$host`, `$dbname`, `$username`, dan `$password`) dideklarasikan untuk menyimpan detail koneksi database seperti alamat host, nama database, username, dan password. Properti publik `$conn` dideklarasikan untuk menyimpan objek koneksi nantinya. Metode publik `getConnection` didefinisikan untuk membuat koneksi ke database. Pada awal metode ini, `$conn` diinisialisasi dengan null untuk memastikan tidak ada koneksi yang terbuka sebelumnya. Kemudian, blok try digunakan untuk mencoba membuat koneksi baru dengan memanggil konstruktor PDO menggunakan informasi koneksi yang telah didefinisikan sebelumnya, dan menetapkan karakter set ke utf8 dengan `exec`. Jika terjadi kesalahan selama proses koneksi, blok catch akan menangkap `PDOException` dan mencetak pesan kesalahan yang relevan. Akhirnya, metode ini mengembalikan objek koneksi `$conn` baik jika berhasil ataupun tidak.

File Model.php

Kode PHP ini mendefinisikan sebuah kelas bernama Model yang mengelola operasi CRUD (Create, Read, Update, Delete) pada tabel-tabel member, buku, dan peminjaman di database. Pada bagian awal, file Koneksi.php di-include untuk mendapatkan koneksi database. Kelas Model memiliki properti privat `$conn` yang

menyimpan koneksi database. Di dalam konstruktor, sebuah objek Koneksi dibuat dan metode `getConnection` dipanggil untuk menginisialisasi properti `$conn`.

Metode privat `executeQuery` digunakan untuk menjalankan query dengan parameter opsional. Query dipersiapkan menggunakan `prepare` dan parameter diikat dengan `bindValue`, lalu dieksekusi dengan `execute`. Metode ini mengembalikan hasil eksekusi query.

Metode `getMembers` digunakan untuk mengambil semua data dari tabel member. Query dipersiapkan, dieksekusi, dan hasilnya diambil dalam bentuk array asosiatif. Metode `addMember`, `updateMember`, dan `deleteMember` digunakan untuk menambah, memperbarui, dan menghapus data anggota, masing-masing. Setiap metode ini mempersiapkan query, mengikat parameter, dan memanggil `executeQuery`.

Metode `getBooks`, `addBook`, `updateBook`, dan `deleteBook` memiliki fungsi serupa namun untuk tabel buku. Demikian pula, metode `getLoans`, `addLoan`, `updateLoan`, dan `deleteLoan` menangani operasi CRUD pada tabel peminjaman.

Secara keseluruhan, kelas `Model` menyediakan antarmuka terstruktur untuk berinteraksi dengan tabel-tabel dalam database, mengabstraksi operasi database dengan metode yang mudah digunakan untuk mengambil, menambah, memperbarui, dan menghapus data dari tabel member, buku, dan peminjaman.

File Member.php

Kode PHP ini berfungsi untuk menampilkan data anggota (member) dalam format HTML. Pertama, file `Model.php` di-include untuk mendapatkan akses ke kelas `Model`. Kemudian, sebuah objek `Model` dibuat dan metode `getMembers` dipanggil untuk mengambil semua data anggota dari database dan menyimpannya dalam variabel `$members`. Kode HTML berikutnya digunakan untuk membangun struktur halaman web. Tag `<!DOCTYPE html>` mendefinisikan tipe dokumen HTML. Dalam tag `<html>`, terdapat bagian `<head>` yang berisi judul halaman "Member". Bagian `<body>` berisi konten utama, dimulai dengan judul "Data Member". Sebuah tabel dengan border ditampilkan, dengan baris pertama sebagai header tabel yang terdiri dari kolom "ID Member", "Nama Member", "Nomor Member", "Tanggal Mendaftar", "Tanggal Terakhir Bayar", dan "Aksi". Kemudian, dengan menggunakan loop `foreach`, setiap elemen dari array `$members` diiterasi untuk membuat baris baru dalam tabel. Data anggota ditampilkan dalam masing-masing kolom menggunakan tag `<td>`. Di kolom terakhir, dua link disediakan untuk setiap anggota: satu untuk mengedit data anggota (dengan mengarahkan ke `FormMember.php` dan menyertakan `id_member` sebagai parameter) dan satu lagi

untuk menghapus data anggota (mengarah ke `deleteMember.php` dengan `id_member` sebagai parameter, serta konfirmasi sebelum menghapus). Setelah tabel, sebuah link disediakan untuk menambah anggota baru, yang mengarah ke `FormMember.php`. Kode ini memberikan antarmuka sederhana untuk melihat, mengedit, dan menghapus data anggota, serta menambah anggota baru.

File FormMember.php

Kode PHP ini digunakan untuk menangani formulir tambah dan edit data anggota (member). Pada bagian awal, file `Model.php` di-include untuk mendapatkan akses ke kelas `Model`. Sebuah objek `Model` dibuat dan beberapa variabel inisialisasi untuk menyimpan informasi anggota. Jika parameter `id` tersedia di URL, maka nilai `id_member` diambil dari parameter tersebut, dan metode `getMemberById` dipanggil untuk mendapatkan data anggota dengan ID tersebut. Jika data anggota ditemukan, nilai-nilai dari anggota tersebut disimpan ke dalam variabel yang sesuai (`$nama_member`, `$nomor_member`, `$tgl_mendaftar`, `$tgl_terakhir_bayar`).

Pada bagian berikutnya, jika formulir dikirimkan (`$_POST`), data yang diterima dari formulir akan diperiksa. Jika ada `id_member`, maka data tersebut akan diperbarui menggunakan metode `updateMember`; jika tidak ada `id_member`, data baru akan ditambahkan menggunakan metode `addMember`. Setelah operasi selesai, halaman akan diarahkan kembali ke `Member.php`.

Bagian HTML dari kode ini mendefinisikan struktur formulir dalam dokumen HTML. Tag `<!DOCTYPE html>` mendefinisikan tipe dokumen HTML. Dalam tag `<html>`, terdapat bagian `<head>` yang berisi judul halaman "Form Member". Bagian `<body>` berisi konten utama, dimulai dengan judul yang menampilkan "Edit Member" jika `id_member` ada, atau "Tambah Member" jika tidak ada. Sebuah formulir HTML dibuat menggunakan tag `<form>` dengan metode pengiriman `post`. Formulir ini berisi beberapa input untuk mengisi informasi anggota seperti "Nama Member", "Nomor Member", "Tanggal Mendaftar", dan "Tanggal Terakhir Bayar". Nilai dari setiap input diambil dari variabel yang telah diinisialisasi sebelumnya. Setelah semua input, sebuah tombol submit disediakan untuk mengirimkan formulir. Sebuah link juga disediakan untuk kembali ke halaman `Member.php`.

Secara keseluruhan, kode ini memungkinkan pengguna untuk menambah anggota baru atau mengedit anggota yang sudah ada dengan mudah melalui formulir yang sama, tergantung pada keberadaan parameter `id` di URL.

File Buku.php

Kode PHP ini digunakan untuk menampilkan data buku dalam format HTML. File Model.php di-include untuk mendapatkan akses ke kelas Model, kemudian sebuah objek Model dibuat dan metode getBooks dipanggil untuk mengambil semua data buku dari database, yang disimpan dalam variabel \$books. Struktur halaman web didefinisikan dalam HTML, dimulai dengan judul "Buku" di dalam tag <head> dan judul "Data Buku" di dalam tag <body>. Sebuah tabel dengan border ditampilkan, dengan header yang terdiri dari kolom "ID Buku", "Judul Buku", "Penulis", "Tahun Terbit", dan "Aksi". Menggunakan loop foreach, setiap elemen dari array \$books diiterasi untuk membuat baris baru dalam tabel, dengan data buku ditampilkan dalam kolom yang sesuai. Di kolom "Aksi", dua link disediakan untuk setiap buku: satu untuk mengedit data buku yang mengarahkan ke FormBuku.php dengan id_buku sebagai parameter, dan satu lagi untuk menghapus data buku yang mengarahkan ke deleteBuku.php dengan id_buku sebagai parameter serta konfirmasi sebelum menghapus. Setelah tabel, sebuah link disediakan untuk menambah buku baru yang mengarahkan ke FormBuku.php. Secara keseluruhan, kode ini memberikan antarmuka sederhana untuk melihat, mengedit, dan menghapus data buku, serta menambah buku baru.

File FormBuku.php

Kode PHP ini digunakan untuk menampilkan formulir untuk menambah atau mengedit data buku. Pertama, file Model.php di-include untuk mendapatkan akses ke kelas Model, kemudian sebuah objek Model dibuat. Variabel-variabel untuk menyimpan informasi buku (\$id_buku, \$judul_buku, \$penulis, \$tahun_terbit) diinisialisasi dengan nilai kosong. Jika parameter id tersedia di URL, nilai id_buku diambil dan metode getBookById dipanggil untuk mendapatkan data buku berdasarkan ID tersebut. Jika data buku ditemukan, nilai-nilai dari buku tersebut disimpan ke variabel yang sesuai. Jika formulir dikirimkan (\$_POST), data yang diterima dari formulir akan diperiksa. Jika id_buku ada, data buku diperbarui menggunakan metode updateBook, jika tidak, data baru ditambahkan menggunakan metode addBook. Setelah operasi selesai, halaman akan diarahkan kembali ke Buku.php. Bagian HTML dari kode ini mendefinisikan struktur formulir dalam dokumen HTML. Tag <!DOCTYPE html> mendefinisikan tipe dokumen HTML. Dalam tag <html>, terdapat bagian <head> yang berisi judul halaman "Form Buku". Bagian <body> berisi konten utama, dimulai dengan judul yang menampilkan "Edit Buku" jika id_buku ada, atau "Tambah Buku" jika tidak ada. Formulir HTML dibuat menggunakan tag <form> dengan metode pengiriman post, berisi input untuk mengisi informasi buku seperti "Judul Buku", "Penulis", dan "Tahun Terbit". Nilai dari setiap input diambil dari variabel yang telah

diinisialisasi sebelumnya. Setelah semua input, sebuah tombol submit disediakan untuk mengirimkan formulir. Sebuah link juga disediakan untuk kembali ke halaman Buku.php. Secara keseluruhan, kode ini memungkinkan pengguna untuk menambah buku baru atau mengedit buku yang sudah ada dengan mudah melalui formulir yang sama, tergantung pada keberadaan parameter id di URL.

File Peminjaman.php

Kode PHP ini digunakan untuk menampilkan data peminjaman dalam format HTML. File Model.php di-include untuk mendapatkan akses ke kelas Model, kemudian objek Model dibuat dan metode getLoans dipanggil untuk mengambil semua data peminjaman dari database, yang disimpan dalam variabel \$loans. Bagian HTML mendefinisikan struktur halaman web dengan judul "Peminjaman" dan konten utama berupa tabel yang berisi data peminjaman. Tabel tersebut memiliki kolom "ID Peminjaman", "ID Member", "ID Buku", "Tanggal Pinjam", "Tanggal Kembali", dan "Aksi". Dengan menggunakan loop foreach, setiap elemen dari array \$loans diiterasi untuk membuat baris baru dalam tabel dengan data peminjaman yang sesuai. Kolom "Aksi" menyediakan dua link untuk setiap peminjaman: satu untuk mengedit data yang mengarah ke FormPeminjaman.php dengan id_peminjaman sebagai parameter, dan satu lagi untuk menghapus data yang mengarah ke deletePeminjaman.php dengan id_peminjaman sebagai parameter serta konfirmasi sebelum menghapus. Setelah tabel, sebuah link disediakan untuk menambah peminjaman baru yang mengarah ke FormPeminjaman.php. Kode ini memberikan antarmuka sederhana untuk melihat, mengedit, dan menghapus data peminjaman, serta menambah peminjaman baru.

File FormPeminjaman.php

Kode PHP ini mengelola formulir untuk menambah atau mengedit data peminjaman dalam aplikasi. Pertama, file Model.php di-include untuk mengakses kelas Model, dan objek \$model dibuat dari kelas tersebut. Variabel \$id_peminjaman, \$id_member, \$id_buku, \$tgl_pinjam, dan \$tgl_kembali diinisialisasi dengan nilai kosong. Jika parameter id tersedia dalam URL (\$_GET['id']), maka nilai dari \$id_peminjaman diambil dari URL dan metode \$model->getLoanById(\$id_peminjaman) dipanggil untuk mengambil data peminjaman yang sesuai. Jika data peminjaman ditemukan, nilai-nilai dari kolom-kolom tersebut diisi dengan data yang sesuai.

Ketika formulir dikirimkan (\$_POST terisi), kode akan memeriksa apakah \$id_peminjaman sudah ada. Jika ya, maka metode \$model->updateLoan dipanggil untuk memperbarui data peminjaman dengan nilai yang diperbarui dari formulir. Jika tidak, metode \$model->addLoan dipanggil untuk menambahkan data peminjaman baru menggunakan nilai-nilai dari formulir. Setelah itu, pengguna akan diarahkan kembali

ke halaman Peminjaman.php menggunakan fungsi `header('Location: Peminjaman.php')`.

Bagian HTML dari kode ini mendefinisikan struktur halaman web dengan judul "Form Peminjaman". Konten utama halaman ini berupa formulir menggunakan tag `<form>` dengan metode post. Terdapat input hidden untuk menyimpan nilai `$id_peminjaman`. Formulir ini memungkinkan pengguna untuk memasukkan atau mengedit informasi peminjaman, termasuk ID Member, ID Buku, Tanggal Pinjam, dan Tanggal Kembali. Nilai-nilai dari setiap input diambil dari variabel yang telah diinisialisasi sebelumnya. Sebuah tombol submit disediakan untuk mengirimkan formulir, serta link untuk kembali ke halaman Peminjaman.php setelah pengguna selesai melakukan operasi..

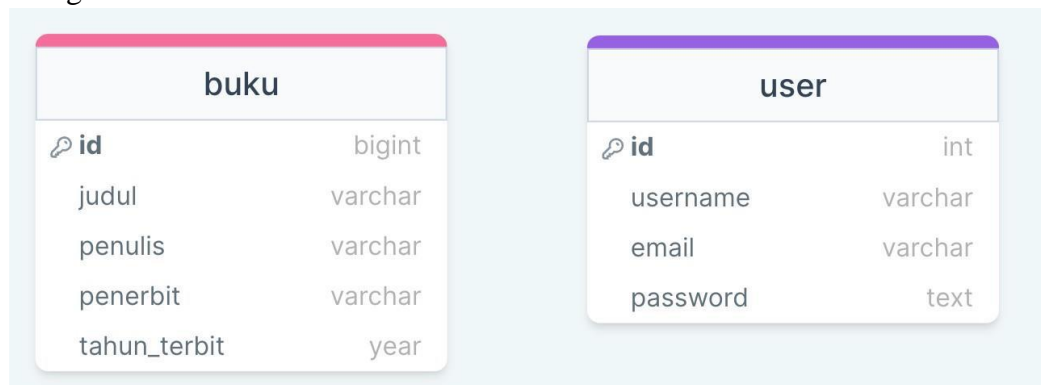
MODUL 6: CRUD DAN LOGIN PADA CODEIGNITER

SOAL

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 tampilan 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 besar dari 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 dalam folder Github PRAK601

A. Source Code

1	<?php
2	
3	use CodeIgniter\Router\RouteCollection;
4	
5	/**
6	* @var RouteCollection \$routes
7	*/
8	\$routes->setAutoRoute(true);
9	\$routes->get('/', 'Buku::login');
10	\$routes->get('/home', 'Buku::index');
11	\$routes->get('/register', 'Register::index');
12	\$routes->post('/register/process',
13	'Register::process');
14	\$routes->get('/login', 'Login::index');
15	\$routes->post('/login/process', 'Login::process');
16	\$routes->get('/logout', 'Login::logout');
17	\$routes->get('/buku/create', 'Buku::create');
18	\$routes->post('buku/store', 'Buku::store');
	\$routes->post('buku/update/', 'Buku::update',
19	['filter' => 'auth']);
	\$routes->get('buku/edit/', 'Buku::edit', ['filter'
20	=> 'auth']);
	\$routes->get('buku/delete/', 'Buku::delete',
	['filter' => 'auth']);

Tabel 29. Source Code Routes.php

```

1  <?php namespace App\Controllers;
2
3  use App\Models\BukuModel;
4  use CodeIgniter\Controller;
5  use App\Models\PostModel;
6
7  class Buku extends Controller
8  {
9      /**
10       *
11       * @var Model
12       */
13     protected $model;
14
15     public function __construct()
16     {
17         $this->model = new BukuModel();
18         $this->helpers = ['form', 'url'];
19     }
20
21
22     public function login()
23     {
24         $data = [
25             'posts' => $this->model->paginate(10),
26             'pager' => $this->model->pager,
27             'title' => 'POST LIST'
28         ];
29
30         return view('vw_login', $data);
31     }
32
33     public function index()
34     {
35         if (!session()->get('logged_in')) {
36             return redirect()-
37 >to(base_url('/login'));
38         }
39         $data = [
40             'posts' => $this->model->paginate(10),
41             'pager' => $this->model->pager,
42             'title' => 'POST LIST'
43         ];
44
45         return view('vw_buku', $data);

```

```

46     }
47
48     public function create()
49     {
50         // Check if user is authenticated
51         if (!session()->get('logged_in')) {
52             return redirect()->to(base_url('/login'));
53         }
54         $data = ['title' => 'Create new post'];
55
56         return view('vw_add', $data);
57     }
58
59     public function store()
60     {
61         $data = $this->request->getPost(['judul',
62 'penulis', 'penerbit', 'tahun_terbit']);
63
64         if (! $this->validateData($data, $this-
65 >model->validationRules)) {
66             return $this->create();
67         }
68
69         $post = $this->validator->getValidated();
70
71         $save = $this->model->save([
72             'judul' => $post['judul'],
73             'penulis' => $post['penulis'],
74             'penerbit' => $post['penerbit'],
75             'tahun_terbit' => $post['tahun_terbit'],
76             ]);
77
78         session()->setFlashdata('success', 'Post
79 has been added successfully.');
```

```

80         return redirect()->to(base_url('home'));
81     }
82
83     public function edit($id)
84     {
85         // Check if user is authenticated
86         if (!session()->get('logged_in')) {
87             return redirect()->to(base_url('/login'));
88         }
89         $post = $this->model->find($id);

```

```

88         if (empty($post)) {
89             session()->setFlashdata('error','Post not
found');
90             return redirect()->back();
91         }
92         $data = [
93             'title' => 'Edit Post',
94             'post' => $post
95         ];
96
97         return view('vw_edit', $data);
98     }
99
100     public function update($id)
101     {
102         // Check if user is authenticated
103         if (!session()->get('logged_in')) {
104             return redirect()->to(base_url('/login'));
105         }
106         $post = $this->model->find($id);
107
108         if (empty($post)) {
109             session()->setFlashdata('error','Post not
found');
110             return redirect()->back();
111         }
112
113         $data = $this->request->getPost(['judul',
'penulis', 'penerbit', 'tahun_terbit']);
114
115         if (! $this->validateData($data, $this->model-
>validationRules)) {
116             return $this->create();
117         }
118
119         $updatedPost = $this->validator-
>getValidated();
120
121         $update = $this->model->update($post['id'],
$updatedPost);
122
123         if ($update) {
124             session()->setFlashdata('success', 'Post
has been updated successfully');
125             return redirect()->to(base_url('home'));

```

126	} else {
127	session()->setFlashdata('error', 'Some
128	problems occured, please try again.');
129	return redirect()->back();
130	}
131	}
132	
133	public function delete(\$id)
134	{
135	// Check if user is authenticated
136	if (!session()->get('logged_in')) {
137	return redirect()->to(base_url('home'));
138	}
139	
140	if (empty(\$id)) {
141	return redirect()->to(base_url('home'));
142	}
143	
144	/delete = \$this->model->delete(\$id);
145	if (\$delete) {
146	session()->setFlashdata('success', 'Post
147	has been removed successfully.');
148	return redirect()->to(base_url('home'));
149	} else {
150	session()->setFlashdata('error', 'Some
151	problems occured, please try again.');
	return redirect()->to(base_url('home'));
	}
	}

Tabel 30. Source Code Buku.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	class Home extends BaseController
6	{
7	public function index(): string
8	{
9	return view('vw_buku');
10	}
11	}

Tabel 31. Source Code Home.php

```

1  <?php
2
3  namespace App\Controllers;
4
5  use App\Models\UsersModel;
6
7  class Login extends BaseController
8  {
9      public function index()
10     {
11         return view('vw_login');
12     }
13
14     public function process()
15     {
16         $users = new UsersModel();
17         $username = $this->request-
18 >getVar('username');
19         $password = $this->request-
20 >getVar('password');
21         $dataUser = $users->where('username',
22 $username)->first();
23
24         if ($dataUser) {
25             // Check the password using object
26             property syntax
27             if (password_verify($password,
28 $dataUser->password)) {
29                 session()->set([
30                     'username' => $dataUser-
31 >username,
32                     'logged_in' => true
33                 ]);
34                 return redirect()-
35 >to(base_url('home'));
36             } else {
37                 session()->setFlashdata('error',
38 'Username & Password Salah');
39                 return redirect()-
40 >to(base_url('login'));
41             }
42         } else {
43             session()->setFlashdata('error',
44 'Username & Password Salah');
45         }
46     }
47 }

```

41	return	redirect()-
42	>to(base_url('login'));	
43	}	
44	}	
	public function logout()	
	{	
	session()->destroy();	
	return redirect()->to(base_url('login'));	
	}	
	}	

Tabel 32. Source Code Login.php

```

1  <?php
2
3  namespace App\Controllers;
4
5  use App\Models\UsersModel;
6
7  class Register extends BaseController
8  {
9      public function index()
10     {
11         return view('vw_register');
12     }
13
14     public function process()
15     {
16         // Validate input
17         if (!$this->validate([
18             'username' => [
19                 'rules' =>
20                 'required|min_length[4]|max_length[20]|is_unique[users.us
21                 ername]',
22                 'errors' => [
23                     'required' => '{field} Harus diisi',
24                     'min_length' => '{field} Minimal 4
25                     Karakter',
26                     'max_length' => '{field} Maksimal 20
27                     Karakter',
28                     'is_unique' => 'Username sudah
29                     digunakan sebelumnya'
30                 ]
31             ],
32             'password' => [
33                 'rules' =>
34                 'required|min_length[0]|max_length[50]',
35                 'errors' => [
36                     'required' => '{field} Harus diisi',
37                     'min_length' => '{field} Minimal 0
38                     Karakter',
39                     'max_length' => '{field} Maksimal 50
40                     Karakter',
41                 ]
42             ],
43             'password_conf' => [
44                 'rules' => 'matches[password]',
45                 'errors' => [

```



```

38         'matches' => 'Konfirmasi Password
tidak sesuai dengan password',
39     ]
40 ],
41     'email' => [
42         'rules' =>
'required|min_length[4]|max_length[100]|valid_email',
43         'errors' => [
44             'required' => '{field} Harus diisi',
45             'min_length' => '{field} Minimal 4
Karakter',
46             'max_length' => '{field} Maksimal 100
Karakter',
47             'valid_email' => 'Email tidak valid'
48         ]
49     ],
50 ])) {
51     session()->setFlashdata('error', $this-
>validator->listErrors());
52     return redirect()->back()->withInput();
53 }
54
55 $users = new UsersModel();
56
57 // Hash the password before storing it
58 $hashedPassword = password_hash($this->request-
>getVar('password'), PASSWORD_DEFAULT);
59
60 $users->insert([
61     'username' => $this->request-
>getVar('username'),
62     'password' => $hashedPassword,
63     'email' => $this->request->getVar('email')
64 ]);
65
66 return redirect()->to('/login');
67 }
68 }

```

Tabel 33. Source Code Register.php

1	<?php
2	// app/Filters/AuthFilter.php
3	
4	namespace App\Filters;
5	
6	use CodeIgniter\HTTP\RequestInterface;
7	use CodeIgniter\HTTP\ResponseInterface;
8	use CodeIgniter\Filters\FilterInterface;
9	
10	class AuthFilter implements FilterInterface
11	{
12	public function before(RequestInterface
	\$request, \$arguments = null)
13	{
14	// Check if the user is logged in
15	if (!session()->has('user_id')) {
16	// If not logged in, redirect to the login
	page
17	return redirect()->to('/login');
18	}
19	}
20	
21	public function after(RequestInterface \$request,
	ResponseInterface \$response, \$arguments = null)
22	{
23	// Do nothing after the request
24	}
25	}

Tabel 34. Source Code AuthFilter.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body {
6	background-image:
	url('https://images.unsplash.com/photo-1661097410573-
	16d926ec1ec7?q=80&w=2070&auto=format&fit=crop&ixlib=rb-
7	
8	
9	
10	
11	
12	

13	4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8f
14	A%3D%3D'); background-repeat: no-repeat; background-attachment: fixed; background-position: center; background-size: cover; color: #ececec; }
15	
16	
17	
18	
19	
20	
21	.card { background: rgba(31, 31, 31); /* Opasitas background 20% */ color: #ececec; box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15); border-radius: 10px; }
22	
23	
24	
25	
26	
27	
28	
29	.card-header { background-color: #00bfa5; border-bottom: 1px solid #37474f; }
30	
31	
32	
33	
34	.btn-success { background-color: #00bfa5; border: none; }
35	
36	
37	
38	
39	.btn-success:hover { background-color: #008e76; }
40	
41	
42	
43	.btn-secondary { background-color: #000000; border: none; }
44	
45	
46	
47	
48	.btn-secondary:hover { background-color: #333333; }
49	
50	
51	
52	.alert { background-color: rgba(255, 138, 101, 0.1); color: #ffab40; border: 1px solid #ffab40; }
53	
54	
55	
56	

57	
58	.form-control {
59	background-color: #2e2e2e;
60	color: #ececce;
61	border: 1px solid #37474f;
62	}
63	
64	.form-control::placeholder {
65	color: #bdbdbd;
66	}
67	
68	label {
69	color: #ececce;
70	}
71	</style>
72	
73	<div class="container mt-5">
	<div class="row justify-content-center">
74	<div class="col-md-8">
75	<div class="card shadow-sm">
	<div class="card-header text-white">
76	<h4>Tambah</h4>
77	</div>
78	<div class="card-body">
79	<?php if (session()-
80	>getFlashdata('error')) : ?>
81	<div class="alert alert-danger">
82	<?= session()-
83	>getFlashdata('error'); ?>
84	</div>
	<?php endif; ?>
85	<?= validation_list_errors() ?>
86	
87	<?= form_open('buku/store'); ?>
	<div class="form-group">
88	<label for="judul">Judul</label>
	<input type="text" class="form-
	control" id="judul" name="judul" placeholder="Masukkan
89	Judul">
90	</div>
91	<div class="form-group">
	<label
92	for="penulis">Penulis</label>

93	control" id="penulis" name="penulis"
94	placeholder="Masukkan Penulis">
95	</div>
	<div class="form-group">
96	<label
	for="penerbit">Penerbit</label>
	<input type="text" class="form-
97	control" id="penerbit" name="penerbit"
98	placeholder="Masukkan Penerbit">
99	</div>
	<div class="form-group">
100	<label for="tahun_terbit">Tahun
	Terbit</label>
101	<input type="number" class="form-
102	control" id="tahun_terbit" name="tahun_terbit"
103	placeholder="Masukkan Tahun Terbit">
104	</div>
105	<div class="form-group text-right">
106	<button class="btn btn-
107	success">Simpan</button>
108	<a href="<?= base_url('home') ?>"
109	class="btn btn-secondary">Kembali
110	</div>
111	<?= form_close(); ?>
112	</div>
113	</div>
	</div>
114	</div>
	<?= \$this->endSection() ?>
115	
116	<?= \$this->section('extra-js') ?>
117	<!-- include summernote css/js -->
118	<link
119	href="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dis
120	t/summernote-bs4.min.css" rel="stylesheet">
121	<script
122	src="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dist
123	/summernote-bs4.min.js"></script>
	<script>
	\$(document).ready(function() {
	\$('#post_content').summernote({
	tabsize: 2,

	<pre> height: 500 }); }); </script> <?= \$this->endSection() ?> </pre>
--	---

Tabel 35. Source Code vw_add.php

1	<pre><?= \$this->extend('layouts/main') ?></pre>
2	
3	<pre><?= \$this->section('content') ?></pre>
4	<pre><style></pre>
5	<pre> body,</pre>
6	<pre> html {</pre>
7	<pre> margin: 0;</pre>
8	<pre> padding: 0;</pre>
9	<pre> font-family: Arial, sans-serif;</pre>
10	<pre> height: 100%;</pre>
11	<pre> display: flex;</pre>
12	<pre> flex-direction: column;</pre>
13	<pre> background-color: #121212;</pre>
14	<pre> color: #ececec;</pre>
15	<pre> position: relative;</pre>
16	<pre> overflow: hidden;</pre>
17	<pre> }</pre>
18	
19	<pre> body::before {</pre>
20	<pre> content: "";</pre>
21	<pre> position: absolute;</pre>
22	<pre> top: 0;</pre>
23	<pre> left: 0;</pre>
24	<pre> width: 100%;</pre>
25	<pre> height: 100%;</pre>
26	<pre> background-image:</pre>
	<pre>url('https://images.unsplash.com/photo-1661097410573-16d926ec1ec7?q=80&w=2070&auto=format&fit=crop&ixlib=rb-</pre>
27	
28	
29	
30	
31	
32	
33	
34	
35	

36	4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8f
37	A%3D%3D'); background-size: cover;
38	background-position: center;
39	opacity: 0.1;
40	z-index: 1;
41	}
42	
43	.navbar {
44	z-index: 2;
45	position: relative;
46	}
47	
48	.container,
49	.card {
50	z-index: 2;
51	position: relative;
52	}
53	
54	.navbar {
55	background-color: #1f1f1f !important;
56	border-bottom: 1px solid #37474f;
57	}
58	
59	.navbar-brand,
60	.nav-link {
61	color: #00bfa5 !important;
62	}
63	
64	.card {
65	background: #1f1f1f;
66	color: #ececce;
67	box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
68	border-radius: 10px;
69	}
70	
71	.card-header {
72	background-color: #00bfa5;
73	border-bottom: 1px solid #37474f;
74	}
75	
76	.btn {
77	background-color: #00bfa5;
78	color: white;
79	border: none;
80	

```

81     }
82
83     .btn:hover {
84         background-color: #008e76;
85     }
86
87     .table thead {
88         background-color: #37474f;
89     }
90
91     .table th,
92     .table td {
93         border: 1px solid #37474f;
94         color: #fff; /* Change table headers and content
95         text color to white */
96     }
97
98     .pagination li a {
99         background-color: #1f1f1f !important;
100        color: #00bfa5 !important;
101        border: 1px solid #00bfa5 !important;
102    }
103
104    .pagination li a:hover {
105        background-color: #00bfa5 !important;
106        color: #1f1f1f !important;
107    }
108
109    .alert {
110        background-color: rgba(255, 138, 101, 0.1);
111        color: #ffab40;
112        border: 1px solid #ffab40;
113    }
114</style>
115
116<!-- Navbar -->
117<nav class="navbar navbar-expand-lg navbar-light shadow-
118sm">
119    <div class="container">
120        <a class="navbar-brand" href="<?= base_url('/')
121        ?>">My Library</a>
122        <button class="navbar-toggler" type="button"
123        data-toggle="collapse" data-target="#navbarNav" aria-

```



```

117 controls="navbarNav"      aria-expanded="false"      aria-
118 label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
119 </button>
120 <div      class="collapse      navbar-collapse"
121 id="navbarNav">
122     <ul class="navbar-nav ml-auto">
123         <?php if (session()->get('logged_in')) :
124         <?>
125             <li class="nav-item">
126                 <a      class="nav-link"
127 href="#">Welcome, <?= session()->get('username'); ?></a>
128             </li>
129             <li class="nav-item">
130                 <a      class="nav-link"      href="<?=
131 base_url('login/logout'); ?>">Logout</a>
132             </li>
133             <?php else : ?>
134                 <li class="nav-item">
135                     <a      class="nav-link"      href="<?=
136 base_url('login'); ?>">Login</a>
137                 </li>
138                 <li class="nav-item">
139                     <a      class="nav-link"      href="<?=
140 base_url('register'); ?>">Register</a>
141                 </li>
142             <?php endif; ?>
143         </ul>
144     </div>
145 </div>
146 </nav>
147
148 <div class="container mt-5">
149     <div class="row justify-content-center">
150     <div class="col-md-10">
151         <div class="card shadow-sm">
152             <div class="card-header text-white">
153                 <h5 class="mb-0">List Buku</h5>
154                 <a href="<?= base_url('buku/create');
155 ?>" class="btn btn-sm float-right">Tambah Data</a>
156             </div>
157             <div class="card-body">
158                 <?php      if      (session()-
159 >getFlashdata('success')) : ?>
160                 <div class="alert alert-success">

```

```

152                                     <?= session()-
153 >getFlashdata('success'); ?>
154                                     </div>
155                                     <?php endif; ?>
156                                     <?php if (session()-
157 >getFlashdata('error')) : ?>
158                                     <div class="alert alert-danger">
159                                     <?= session()-
160 >getFlashdata('error'); ?>
161                                     </div>
162                                     <?php endif; ?>
163
164                                     <table class="table table-bordered
165 table-striped table-hover">
166                                     <thead class="text-center">
167                                     <tr>
168                                     <th
169 scope="col">Judul</th>
170                                     <th
171 scope="col">Penulis</th>
172                                     <th
173 scope="col">Penerbit</th>
174                                     <th scope="col">Tahun
175 Terbit</th>
176                                     <th
177 scope="col">Action</th>
178                                     </tr>
179                                     </thead>
180                                     <tbody>
181                                     <?php if (!empty($posts) &&
182 is_array($posts)) : ?>
183                                     <?php foreach ($posts as
184 $row) : ?>
185                                     <tr>
186                                     <td><?=
187 $row['judul']; ?></td>
188                                     <td><?=
189 $row['penulis']; ?></td>
190                                     <td><?=
191 $row['penerbit']; ?></td>
192                                     <td><?=
193 $row['tahun_terbit']; ?></td>
194                                     <td class="text-
195 center">

```

177	<form
	onsubmit="return confirm('Apakah Anda Yakin ?');"
178	
179	
180	
181	
182	
183	
184	
185	
186	
187	
188	
189	
190	
191	
192	
193	
194	
195	
196	
197	
198	
199	
200	
201	
202	
203	
204	
205	
206	
207	
208	

	<pre> action="<?= base_url('buku/delete/' . \$row['id']); ?>" method="POST"> <input type="hidden" name="{csrf_token}" value="{csrf_hash}"> <input type="hidden" name="_method" value="DELETE"> <a href="<?= base_url('buku/edit/' . \$row['id']); ?>" class="btn btn-warning btn-sm">Edit <button type="submit" class="btn btn-danger btn- sm">Delete</button> </form> </td> </tr> <?php endforeach; ?> <?php else : ?> <tr> <td colspan="5" class="text-center">No post found.</td> </tr> <?php endif; ?> </tbody> </table> <div class="d-flex justify-content- center"> <?= \$pager->links(); ?> </div> </div> </div> </div> </div> </div> </div> <?= \$this->endSection() ?> <?= \$this->section('extra-js') ?> <script> \$(document).ready(function() { \$('.pagination li').addClass('page-item'); \$('.pagination li a').addClass('page-link'); }); </script> <?= \$this->endSection() ?> </pre>
--	---

Tabel 36. Source Code vw_buku.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body {
6	background-image:
	url('https://images.unsplash.com/photo-1661097410573-
	16d926ec1ec7?q=80&w=2070&auto=format&fit=crop&ixlib=rb-
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	
32	
33	
34	
35	
36	
37	
38	
39	
40	

41	4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8f
42	A%3D%3D'); background-repeat: no-repeat;
43	background-attachment: fixed;
44	background-position: center;
45	background-size: cover;
46	color: #ececec;
47	}
48	
49	.card {
50	background: #1f1f1f;
51	color: #ececec;
52	box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
53	border-radius: 10px;
54	}
55	
56	.card-header {
57	background-color: #00bfa5;
58	border-bottom: 1px solid #37474f;
59	}
60	
61	.btn-success {
62	background-color: #00bfa5;
63	border: none;
64	}
65	
66	.btn-success:hover {
67	background-color: #008e76;
68	}
69	
70	.btn-secondary {
71	background-color: #000000;
72	border: none;
73	}
74	
75	.btn-secondary:hover {
76	background-color: #333333;
77	}
78	.alert {
79	background-color: rgba(255, 138, 101, 0.1);
80	color: #ffab40;
81	border: 1px solid #ffab40;
82	}

83	.form-control {
	background-color: #2e2e2e;
84	color: #ececec;
85	border: 1px solid #37474f;
86	}
87	
	.form-control::placeholder {
	color: #bdbdbd;
88	}
89	
90	label {
	color: #ececec;
91	}
	</style>
	<div class="container mt-5">
92	<div class="row justify-content-center">
93	<div class="col-md-8">
94	<div class="card shadow-sm">
	<div class="card-header text-white">
95	<h4>Edit</h4>
	</div>
	<div class="card-body">
96	<?php if (session()-
97	>getFlashdata('error')) : ?>
98	<div class="alert alert-danger">
	<?= session()-
99	>getFlashdata('error'); ?>
	</div>
	<?php endif; ?>
100	<?= validation_list_errors() ?>
101	
102	<?= form_open('buku/update/' .
	\$post['id']); ?>
103	<input type="hidden" name="_method"
	value="PUT">
104	
105	<div class="form-group">
106	<label for="judul">Judul</label>
107	<input type="text" class="form-
108	control" id="judul" name="judul" placeholder="Masukkan
109	Judul" value="<?= \$post['judul'] ?>">
110	</div>

111	<div class="form-group">
112	<label
113	for="penulis">Penulis</label>
114	<input type="text" class="form-
115	control" id="penulis" name="penulis"
116	placeholder="Masukkan Penulis" value="<?="
117	\$post['penulis'] ?>">
	</div>
	<div class="form-group">
118	<label
	for="penerbit">Penerbit</label>
	<input type="text" class="form-
119	control" id="penerbit" name="penerbit"
120	placeholder="Masukkan Penerbit" value="<?="
121	\$post['penerbit'] ?>">
122	</div>
123	<div class="form-group">
124	<label for="tahun_terbit">Tahun
125	Terbit</label>
126	<input type="number" class="form-
127	control" id="tahun_terbit" name="tahun_terbit"

	<pre> placeholder="Masukkan Tahun Terbit" value="<?=\$post['tahun_terbit'] ?>" </div> <div class="form-group text-right"> <button class="btn btn-success">Update</button> <a href="<?= base_url('home') ?>" class="btn btn-secondary">Kembali </div> <?= form_close(); ?> </div> </div> </div> </div> </div> </div> <?= \$this->endSection() ?> <?= \$this->section('extra-js') ?> <!-- include summernote css/js --> <link href="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dist/summernote-bs4.min.css" rel="stylesheet"> <script src="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dist/summernote-bs4.min.js"></script> <script> \$(document).ready(function() { \$('#post_content').summernote({ tabsize: 2, height: 500 }); }); </script> <?= \$this->endSection() ?> </pre>
--	---

Tabel 37. Source Code vw_edit.php

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-
width, initial-scale=1.0">
7      <title>Login</title>
8      <style>
9          body {
10              display: flex;
11              justify-content: center;
12              align-items: center;
13              flex-direction: column;
14              min-height: 100vh;
15              padding: 0;
16              margin: 0;
17              font-family: Arial, sans-serif;
18              background-color: #121212;
19              color: #ececec;
20              position: relative;
21              overflow: hidden;
22          }
23
24          body::before {
25              content: "";
26              position: absolute;
27              top: 0;
28              left: 0;
29              width: 100%;
30              height: 100%;
31              background-image:
url('https://iispsm.sch.id/new/wp-
content/uploads/2021/06/library.jpg');
32              background-size: cover;
33              background-position: center;
34              opacity: 0.1; /* Set the opacity to 20%
35          */
36              z-index: 1;
37          }
38
39          .welcome-message,
40          .form-signin {
41              z-index: 2;
42              position: relative;

```

43	}
44	
45	.welcome-message {
46	width: 100%;
47	max-width: 600px;
48	padding: 20px;
49	text-align: center;
50	margin-bottom: 20px;
51	background: #1f1f1f;
52	border-radius: 10px;
53	box-shadow: 0 6px 10px rgba(0, 0, 0,
54	0.15);
55	color: #00bfa5;
56	font-size: 24px;
57	}
58	
59	.form-signin {
60	width: 100%;
61	max-width: 330px;
62	padding: 20px;
63	margin: auto;
64	background: #1f1f1f;
65	border-radius: 10px;
66	box-shadow: 0 6px 10px rgba(0, 0, 0,
67	0.15);
68	position: relative;
69	}
70	
71	.form-signin img {
72	margin-bottom: 20px;
73	}
74	
75	.form-signin h1 {
76	margin-bottom: 10px;
77	font-size: 24px;
78	font-weight: normal;
79	color: #00bfa5;
80	}
81	
82	.form-signin p {
83	margin-bottom: 20px;
84	font-size: 14px;
85	color: #b0bec5;
86	}
87	

88	.form-signin input[type="text"],
89	.form-signin input[type="password"] {
90	width: 100%;
91	padding: 10px;
92	margin-bottom: 10px;
93	border: 1px solid #37474f;
94	border-radius: 4px;
95	background: #263238;
96	color: #ececec;
97	box-sizing: border-box;
98	}
99	.form-signin
100	input[type="text"]::placeholder,
101	.form-signin
102	input[type="password"]::placeholder {
103	color: #90a4ae;
104	}
105	
106	.form-signin button {
107	width: 100%;
108	padding: 10px;
109	background-color: #00bfa5;
110	color: white;
111	border: none;
112	border-radius: 4px;
113	cursor: pointer;
114	font-size: 16px;
115	}
116	
117	.form-signin button:hover {
118	background-color: #008e76;
119	}
120	
121	.form-signin .alert {
122	padding: 10px;
123	margin-bottom: 10px;
124	border-radius: 4px;
125	color: #ffab40;
126	background-color: rgba(255, 138, 101,
127	0.1);
128	border: 1px solid #ffab40;
129	display: none;
130	position: relative;
131	}

```

132
133     .form-signin .text-muted {
134         color: #b0bec5;
135         margin-top: 20px;
136     }
137
138     .btn-register {
139         display: block;
140         width: 93%;
141         text-align: center;
142         padding: 10px;
143         margin-top: 10px;
144         background-color: #37474f;
145         color: #00bfa5;
146         border: 1px solid #00bfa5;
147         border-radius: 4px;
148         cursor: pointer;
149         font-size: 14px;
150         text-decoration: none;
151     }
152
153     .btn-register:hover {
154         background-color: #00bfa5;
155         color: #121212;
156     }
157 </style>
158 </head>
159
160 <body>
161
162     <div class="welcome-message">
163         Selamat Datang di Perpustakaan Bersama
164     </div>
165
166     <main class="form-signin">
167         <!-- Alert untuk informasi saat halaman
168         dimuat -->
169         <div class="alert" id="initial-alert"
170         style="display: block;">
171             Login terlebih dahulu!
172         </div>
173
174         <!-- Alert untuk kesalahan login -->

```

173	<div class="alert" style="display: <?php
174	echo (session()->getFlashdata('error')) ? 'block' :
175	'none'; ?>;">
176	<?= session()->getFlashdata('error');
177	?>
178	</div>
179	<form method="post" action="<?= base_url();
180	?>/login/process">
181	<?= csrf_field(); ?>
182	<h1 class="h3 mb-3 fw-
183	normal">Login</h1>
184	<p>Register dulu apabila belum memiliki
185	akun</p>
186	<input type="text" name="username"
187	id="username" placeholder="Username" required
	autofocus>
	<input type="password" name="password"
	id="password" placeholder="Password" required>
	<button type="submit">Login</button>
	</form>
	<a class="btn-register" href="<?=
	base_url('register'); ?>">Register
	</main>
	</body>
	</html>

Tabel 38. Source Code vw_login.php

1	<!DOCTYPE html>
2	<html lang="en" class="h-100">
3	
4	<head>
5	<meta charset="utf-8">
6	<meta name="viewport" content="width=device-width, initial-scale=1.0">
7	<title>Register</title>
8	<style>
9	body,
10	html {
11	margin: 0;
12	padding: 0;
13	font-family: Arial, sans-serif;
14	height: 100%;
15	display: flex;
16	justify-content: center;
17	align-items: center;
18	flex-direction: column;
19	background-color: #121212;
20	color: #ececec;
21	}
22	
23	body::before {
24	content: "";
25	position: absolute;
26	top: 0;
27	left: 0;
28	width: 100%;
29	height: 100%;
30	background-image:
	url('https://iispsm.sch.id/new/wp-
	content/uploads/2021/06/library.jpg');
31	background-size: cover;
32	background-position: center;
33	opacity: 0.1;
34	z-index: 1;
35	}
36	
37	.welcome-message,
38	.form-register {
39	z-index: 2;
40	position: relative;
41	}
42	

43	.welcome-message {
44	width: 100%;
45	max-width: 600px;
46	padding: 20px;
47	text-align: center;
48	margin-bottom: 20px;
49	background: #1f1f1f;
50	border-radius: 10px;
51	box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
52	color: #00bfa5;
53	font-size: 24px;
54	}
55	
56	.form-register {
57	width: 100%;
58	max-width: 330px;
59	padding: 20px;
60	margin: auto;
61	background: #1f1f1f;
62	border-radius: 10px;
63	box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
64	position: relative;
65	}
66	
67	.form-register h1 {
68	margin-bottom: 10px;
69	font-size: 24px;
70	font-weight: normal;
71	color: #00bfa5;
72	}
73	
74	.form-register p {
75	margin-bottom: 20px;
76	font-size: 14px;
77	color: #b0bec5;
78	}
79	
80	.form-register input[type="text"],
81	.form-register input[type="password"],
82	.form-register input[type="email"] {
83	width: 100%;
84	padding: 10px;
85	margin-bottom: 10px;
86	border: 1px solid #37474f;
87	border-radius: 4px;

88	background: #263238;
89	color: #ececec;
90	box-sizing: border-box;
91	}
92	
93	.form-register input[type="text"]::placeholder,
94	.form-register
95	input[type="password"]::placeholder,
96	.form-register input[type="email"]::placeholder {
97	color: #90a4ae;
98	}
99	.form-register button {
100	width: 100%;
101	padding: 10px;
102	background-color: #00bfa5;
103	color: white;
104	border: none;
105	border-radius: 4px;
106	cursor: pointer;
107	font-size: 16px;
108	}
109	.form-register button:hover {
110	background-color: #008e76;
111	}
112	
113	.form-register .alert {
114	padding: 10px;
115	margin-bottom: 10px;
116	border-radius: 4px;
117	color: #ffab40;
118	background-color: rgba(255, 138, 101, 0.1);
119	border: 1px solid #ffab40;
120	display: none;
121	position: relative;
122	}
123	
124	.form-register .text-muted {
125	color: #b0bec5;
126	margin-top: 20px;
127	}
128	</style>
129	</head>
130	

```

131 <body>
132
133     <div class="welcome-message">
134         Form Pendaftaran
135     </div>
136
137     <main class="form-register">
138         <!-- Alert untuk informasi saat halaman dimuat -->
139     >
140         <div class="alert" id="initial-alert"
style="display: block;">
            Isi form untuk mendaftar!
141         </div>
142
143         <!-- Alert untuk kesalahan register -->
144         <div class="alert" style="display: <?php echo
145 (session()->getFlashdata('error')) ? 'block' : 'none';
?>;">
            <?= session()->getFlashdata('error'); ?>
146         </div>
147
148         <form method="post" action="<?= base_url();
149 ?>/register/process">
            <?= csrf_field(); ?>
150             <h1 class="h3 mb-3 fw-normal">Register</h1>
151             <p>Isi data di bawah untuk membuat akun
152 baru</p>
153             <input type="text" name="username"
154 id="username" placeholder="Username" required autofocus>
            <input type="password" name="password"
155 id="password" placeholder="Password" required>
            <input type="password" name="password_conf"
156 id="password_conf" placeholder="Confirm Password"
required>
            <input type="email" name="email" id="email"
157 placeholder="Email" required>
158             <button type="submit">Register</button>
159         </form>
160     </main>
161
162 </body>
163
164 </html>

```

165	
-----	--

Tabel 39. Source Code vw_register.php

```

1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class BukuModel extends Model
8  {
9      protected $table = 'buku';
10     protected $allowedFields = ['judul', 'penulis' ,
    'penerbit', 'tahun_terbit'];
11     protected $createdField = 'created_at';
12     protected $updatedField = 'updated_at';
13     protected $deletedField = 'deleted_at';
14
15     protected $validationRules = ([
16         'judul' => [
17             'rules' => 'required|string',
18             'errors' => [
19                 'required' => 'Judul harus diisi.',
20                 'string' => 'Judul harus berupa string.'
21             ]
22         ],
23         'penulis' => [
24             'rules' => 'required|string',
25             'errors' => [
26                 'required' => 'Penulis harus diisi.',
27                 'string' => 'Penulis harus berupa
    string.'
28             ]
29         ],
30         'penerbit' => [
31             'rules' => 'required|string',
32             'errors' => [
33                 'required' => 'Penerbit harus diisi.',
34                 'string' => 'Penerbit harus berupa
    string.'
35             ]
36         ],
37         'tahun_terbit' => [
38             'rules' => 'required|integer|greater_than[1800]|less_than[2024]',
39             'errors' => [
40                 'required' => 'Tahun terbit harus
    diisi.',

```

41	<code>angka.',</code>	<code>'integer' => 'Tahun terbit harus berupa</code>
42	<code>lebih besar dari 1800.',</code>	<code>'greater_than' => 'Tahun terbit harus</code>
43	<code>kecil dari 2024.'</code>	<code>'less_than' => 'Tahun terbit harus lebih</code>
44	<code>]</code>	
45	<code>]);</code>	
46		
47	<code>protected \$skipValidation = false;</code>	
48		
49	<code>}</code>	
50		

Tabel 40. Source Code BukuModel.php

1	<code><?php</code>
2	
3	<code>namespace App\Models;</code>
4	
5	<code>use CodeIgniter\Model;</code>
6	
7	<code>class UsersModel extends Model</code>
8	<code>{</code>
9	<code>protected \$table = "users";</code>
10	<code>protected \$returnType = "object";</code>
11	<code>protected \$useTimestamps = true;</code>
12	<code>protected \$allowedFields = ['username', 'password',</code>
13	<code>'email'];</code>
	<code>}</code>

Tabel 41. Source Code UserModel.php

```

1  <?php
2
3  namespace App\Database\Migrations;
4
5  use CodeIgniter\Database\Migration;
6
7  class Users extends Migration
8  {
9      public function up()
10     {
11         $this->forge->addField([
12             'id' => [
13                 'type' => 'INT',
14                 'unsigned' => TRUE,
15                 'auto_increment' => TRUE
16             ],
17             'username' => [
18                 'type' => 'VARCHAR',
19                 'constraint' => 255,
20                 'null' => FALSE,
21             ],
22             'email' => [
23                 'type' => 'VARCHAR',
24                 'constraint' => 255,
25                 'null' => FALSE,
26             ],
27             'password' => [
28                 'type' => 'TEXT',
29                 'null' => FALSE,
30             ],
31             'created_at' => [
32                 'type' => 'datetime',
33                 'null' => TRUE
34             ],
35             'updated_at' => [
36                 'type' => 'datetime',
37                 'null' => TRUE
38             ]
39         ]);
40
41         $this->forge->addKey('id', TRUE);
42         $this->forge->createTable('users');
43     }
44
45

```

46	//-----
47	-----
48	public function down()
49	{
50	\$this->forge->dropTable('users');
51	}
	}

Tabel 42. Source Code 2024-06-03-103310_Users.php

```

1  <?php
2
3  namespace App\Database\Migrations;
4
5  use CodeIgniter\Database\Migration;
6
7  class Buku extends Migration
8  {
9      public function up()
10     {
11         $this->forge->addField([
12             'id' => [
13                 'type' => 'BIGINT',
14                 'unsigned' => TRUE,
15                 'auto_increment' => TRUE
16             ],
17             'judul' => [
18                 'type' => 'VARCHAR',
19                 'constraint' => 255,
20                 'null' => FALSE,
21             ],
22             'penulis' => [
23                 'type' => 'VARCHAR',
24                 'constraint' => 255,
25                 'null' => FALSE,
26             ],
27             'penerbit' => [
28                 'type' => 'VARCHAR',
29                 'constraint' => 255,
30                 'null' => FALSE,
31             ],
32             'tahun_terbit' => [
33                 'type' => 'YEAR',
34                 'null' => TRUE
35             ]
36         ]);
37
38         $this->forge->addKey('id', TRUE);
39         $this->forge->createTable('buku');
40     }
41
42     public function down()
43     {
44         $this->forge->dropTable('buku');
45     }

```


46	}
----	---

Tabel 43. Source Code 2024-06-03-122804_Buku.php

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class Buku extends Seeder
8	{
9	public function run()
10	{
11	\$this->db->table('buku')->insert([
12	'judul' => 'Laut Bercerita',
13	'penulis' => 'Leila S. Chudori',
14	'penerbit' => 'Gramedia',
15	'tahun_terbit' => '2017'
16]);
17	
18	\$this->db->table('buku')->insert([
19	'judul' => 'Tentang Kamu',
20	'penulis' => 'Tere Liye',
21	'penerbit' => 'Republika',
22	'tahun_terbit' => '2016'
23]);
24	}
25	}

Tabel 44. Source Code Seeds Buku.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('users')->insert([
12	'username' => 'Farly',
13	'email' => 'farlyhaydy123@gmail.com',
14	'password' => password_hash('123',
15	PASSWORD_DEFAULT),
16]);
17	}
17	}

Tabel 45. Source Code UserSeeder.php

```

1  -- phpMyAdmin SQL Dump
2  -- version 5.2.1
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: 127.0.0.1
6  -- Waktu pembuatan: 11 Jun 2024 pada 19.57
7  -- Versi server: 10.4.32-MariaDB
8  -- Versi PHP: 8.2.12
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET
   @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET
   @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET
   @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Database: `perpustakaan`
22 --
23
24 -- -----
25 ---
26 --
27 -- Struktur dari tabel `buku`
28 --
29
30 CREATE TABLE `buku` (
31   `id` bigint(20) UNSIGNED NOT NULL,
32   `judul` varchar(255) NOT NULL,
33   `penulis` varchar(255) NOT NULL,
34   `penerbit` varchar(255) NOT NULL,
35   `tahun_terbit` year(4) DEFAULT NULL
36 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
  COLLATE=utf8mb4_general_ci;
37
38 --
39 -- Dumping data untuk tabel `buku`
40 --

```

```

41
42 INSERT INTO `buku` (`id`, `judul`, `penulis`, `penerbit`,
43 `tahun_terbit`) VALUES
44 (15, 'Laut Bercerita', 'Leila S. Chudori', 'Gramedia',
45 '2017'),
46 (16, 'Tentang Kamu', 'Tere Liye', 'Republika', '2016');
47
48 ---
49 --
50 -- Struktur dari tabel `migrations`
51 --
52 CREATE TABLE `migrations` (
53   `id` bigint(20) UNSIGNED NOT NULL,
54   `version` varchar(255) NOT NULL,
55   `class` varchar(255) NOT NULL,
56   `group` varchar(255) NOT NULL,
57   `namespace` varchar(255) NOT NULL,
58   `time` int(11) NOT NULL,
59   `batch` int(11) UNSIGNED NOT NULL
60 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
61 COLLATE=utf8mb4_general_ci;
62
63 --
64 -- Dumping data untuk tabel `migrations`
65 --
66 INSERT INTO `migrations` (`id`, `version`, `class`,
67 `group`, `namespace`, `time`, `batch`) VALUES
68 (1, '2024-06-03-103310',
69 'App\\Database\\Migrations\\Users', 'default', 'App',
70 1718081782, 1),
71 (2, '2024-06-03-122804',
72 'App\\Database\\Migrations\\Buku', 'default', 'App',
73 1718081782, 1),
74 (4, '2024-06-11-173406',
75 'App\\Database\\Migrations\\Users', 'default', 'App',
76 1718127395, 3);
77
78 ---
79 --

```

```

74 -- Struktur dari tabel `users`
75 --
76
77 CREATE TABLE `users` (
78   `id` int(10) UNSIGNED NOT NULL,
79   `username` varchar(255) NOT NULL,
80   `email` varchar(255) NOT NULL,
81   `password` text NOT NULL,
82   `created_at` datetime DEFAULT NULL,
83   `updated_at` datetime DEFAULT NULL
84 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
  COLLATE=utf8mb4_general_ci;
85
86 --
87 -- Dumping data untuk tabel `users`
88 --
89
90 INSERT INTO `users` (`id`, `username`, `email`,
91   `password`, `created_at`, `updated_at`) VALUES
92   (4, 'Farly', 'farlyhaydy123@gmail.com',
93     '$2y$10$xYhsftGqi132/1Y8Qrypge2VGyR2bS0TEBFpXTa4sXUKI4Gw
94     nEoh2', NULL, NULL);
95
96 --
97 -- Indexes for dumped tables
98 --
99 -- Indeks untuk tabel `buku`
100 --
101 ALTER TABLE `buku`
102   ADD PRIMARY KEY (`id`);
103 --
104 -- Indeks untuk tabel `migrations`
105 --
106 ALTER TABLE `migrations`
107   ADD PRIMARY KEY (`id`);
108 --
109 -- Indeks untuk tabel `users`
110 --
111 ALTER TABLE `users`
112   ADD PRIMARY KEY (`id`),
113   ADD UNIQUE KEY `email` (`email`);
114

```

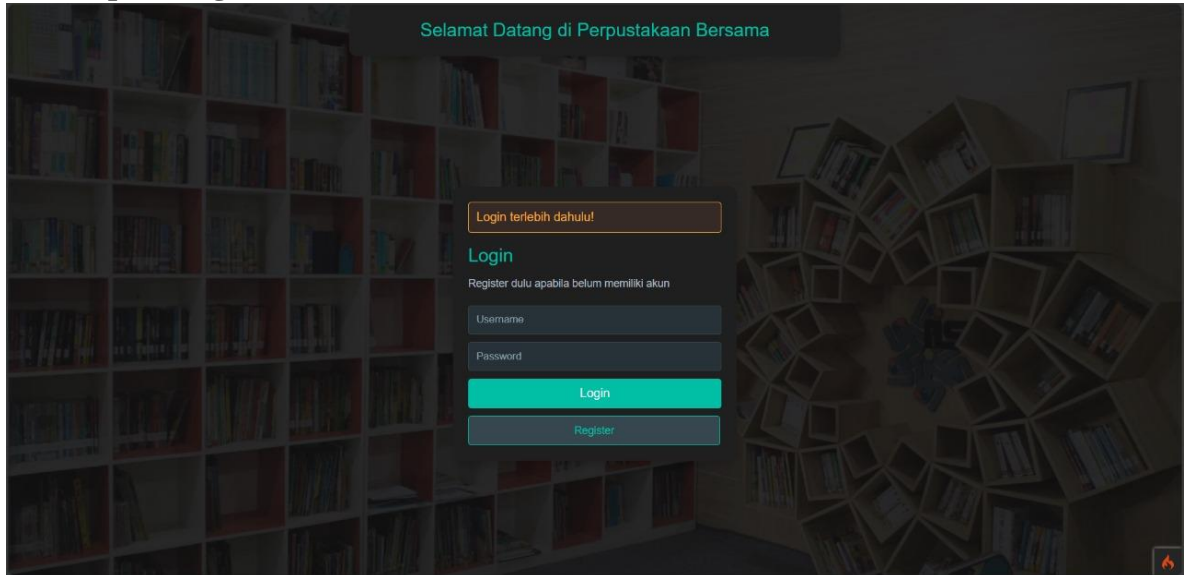
```

115
116 --
117 -- AUTO_INCREMENT untuk tabel yang dibuang
118 --
119
120 --
121 -- AUTO_INCREMENT untuk tabel `buku`
122 --
123 ALTER TABLE `buku`
124     MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=19;
125
126 --
127 -- AUTO_INCREMENT untuk tabel `migrations`
128 --
129 ALTER TABLE `migrations`
130     MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=5;
131
132 --
133 -- AUTO_INCREMENT untuk tabel `users`
134 --
135 ALTER TABLE `users`
136     MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=5;
137 COMMIT;
138
139 /*!40101                                SET
    CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
140 /*!40101                                SET
    CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
141 /*!40101                                SET
    COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

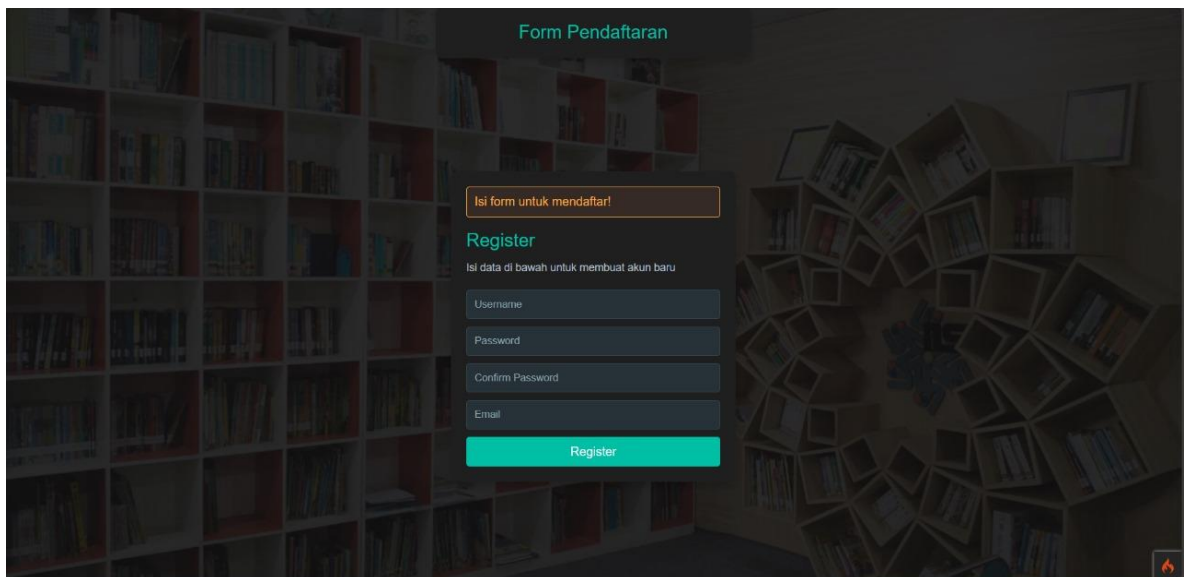
```

Tabel 46. Source Code perpustakaan.sql

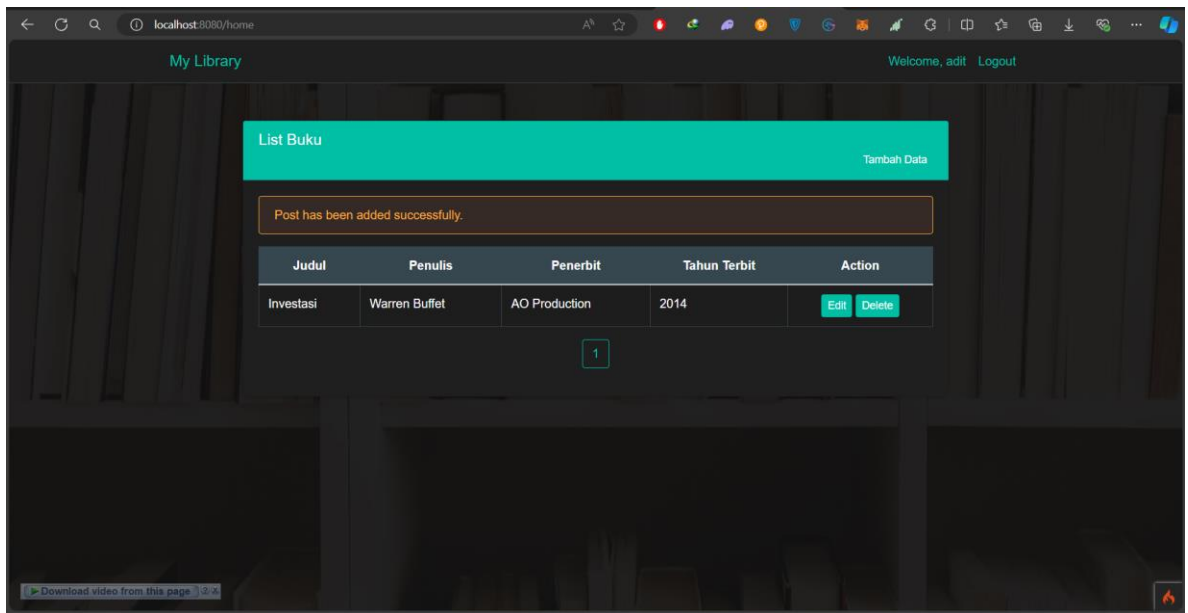
B. Output Program



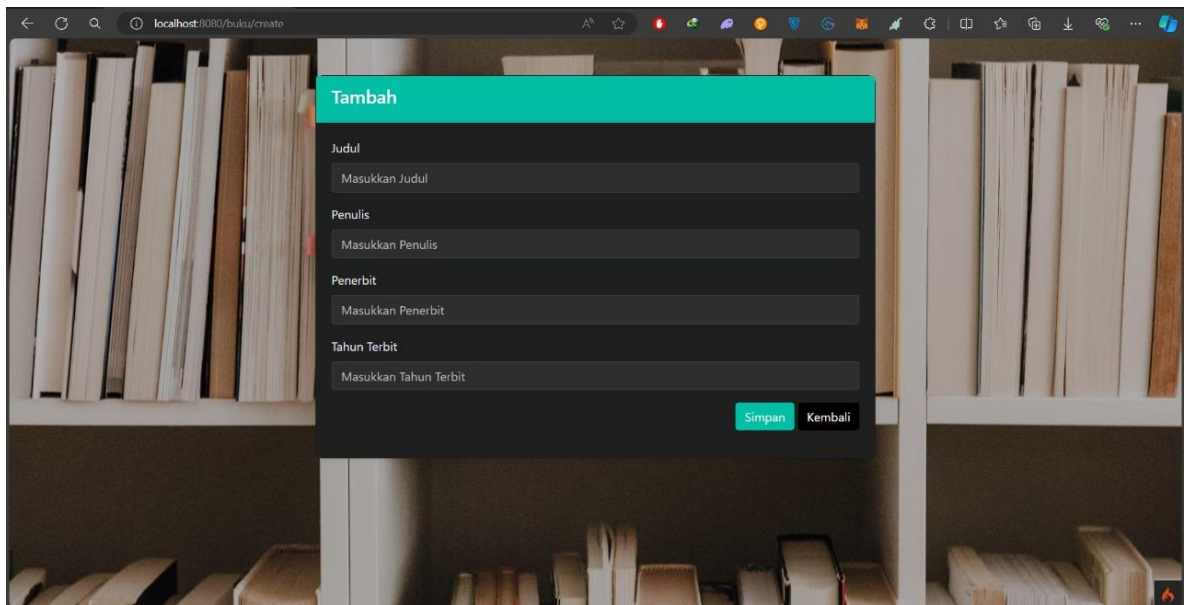
Gambar 27. Halaman Login



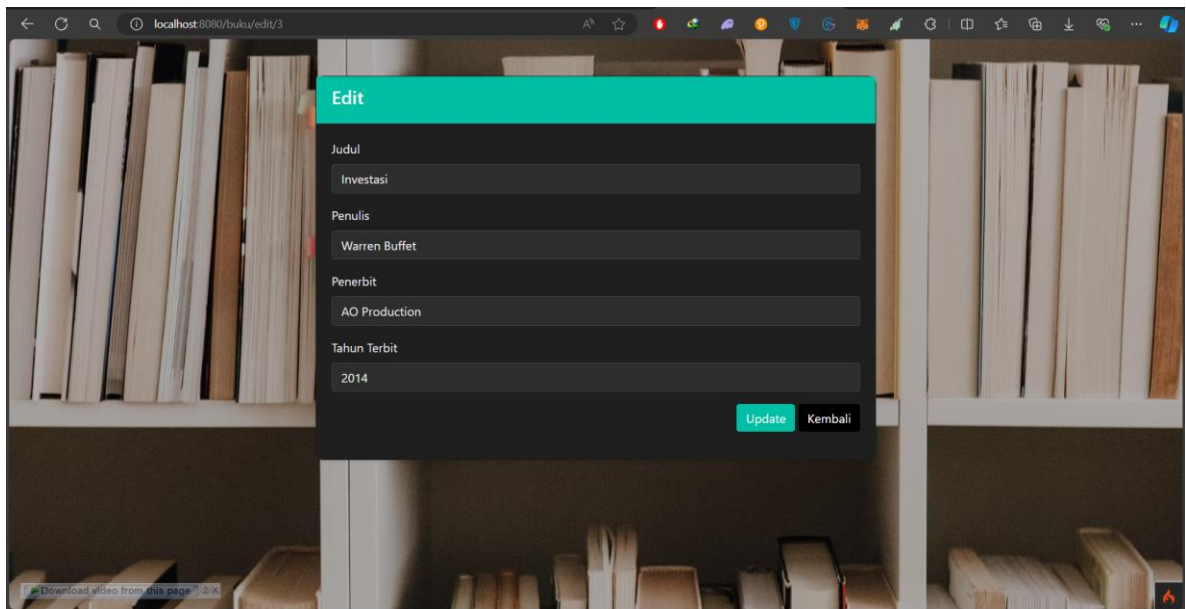
Gambar 28. Halaman Register



Gambar 29. Halaman Home



Gambar 30. Halaman Untuk Menambahkan Data



Gambar 31. Halaman Untuk Edit Data

C. Pembahasan

Routes.php

Kode di atas merupakan konfigurasi routing menggunakan framework CodeIgniter. Pertama, menggunakan namespace CodeIgniter\Router\RouteCollection, di mana \$routes merupakan objek RouteCollection yang digunakan untuk mendefinisikan rute aplikasi. setAutoRoute(true) mengaktifkan pengaturan otomatis untuk mencocokkan rute berdasarkan kontroler dan metode yang disebutkan. Rute-rute yang didefinisikan kemudian adalah sebagai berikut: Rute '/' mengarah ke metode login dari kontroler Buku. Rute 'home' mengarah ke metode index dari kontroler Buku. Rute '/register' mengarah ke metode index dari kontroler Register, dan rute '/register/process' mengarah ke metode process dari kontroler Register, dengan metode HTTP POST. Rute '/login' mengarah ke metode index dari kontroler Login, dan rute '/login/process' mengarah ke metode process dari kontroler Login, dengan metode HTTP POST. Rute '/logout' mengarah ke metode logout dari kontroler Login. Rute '/buku/create' mengarah ke metode create dari kontroler Buku, dan rute '/buku/store' mengarah ke metode store dari kontroler Buku, dengan metode HTTP POST. Terakhir, rute '/buku/update/', '/buku/edit/', dan '/buku/delete/' masing-masing mengarah ke metode update, edit, dan delete dari kontroler Buku, dengan pengecualian filter auth yang menunjukkan bahwa akses ke rute-rute tersebut memerlukan autentikasi pengguna.

Buku.php

Kode kontroler Buku dalam aplikasi CodeIgniter mengelola operasi CRUD untuk entitas buku. Dalam konstruktor, BukuModel diinisialisasi untuk interaksi dengan

database, dan helper form serta url dimuat untuk keperluan pembuatan formulir dan pengelolaan URL. Metode seperti login(), index(), create(), store(), edit(\$id), update(\$id), dan delete(\$id) memfasilitasi tampilan dan manipulasi data buku, dengan memvalidasi input sebelum menyimpan atau memperbarui data, serta memastikan akses terbatas pada pengguna yang sudah login. Pesan sukses atau gagal disertakan dalam setiap operasi untuk memberikan umpan balik kepada pengguna terkait hasil operasi CRUD yang dilakukan.

Home.php

Kode di atas adalah kontroler Home dalam aplikasi CodeIgniter. Kontroler ini merupakan turunan dari kelas BaseController, yang memungkinkannya untuk mengakses fitur dan metode dasar dari framework CodeIgniter. Dalam metode index(), kontroler ini mengembalikan tampilan dengan nama vw_buku, yang kemungkinan besar merupakan halaman utama atau halaman yang menampilkan daftar buku dalam aplikasi tersebut. Dengan mengembalikan tampilan ini, pengguna dapat melihat konten yang sesuai dengan fungsi dari aplikasi buku yang sedang dibangun.

Login.php

Kode kontroler Login dalam aplikasi CodeIgniter mengelola proses autentikasi pengguna. Metode index() memungkinkan pengguna untuk melihat halaman login (vw_login). Metode process() memvalidasi username dan password yang diterima dari formulir login, menggunakan UsersModel untuk mencari pengguna berdasarkan username, dan memverifikasi password dengan fungsi password_verify(). Jika autentikasi berhasil, session pengguna diset dengan informasi username dan status logged_in, dan pengguna diarahkan ke halaman utama (home). Jika gagal, pesan kesalahan ditampilkan dan pengguna diarahkan kembali ke halaman login. Metode logout() menghapus semua session dan mengarahkan pengguna kembali ke halaman login, memastikan proses logout berjalan dengan baik. Ini memastikan bahwa aplikasi hanya memberikan akses ke konten yang sesuai dengan pengguna yang telah terautentikasi dengan benar.

Register.php

Kode kontroler Register dalam aplikasi CodeIgniter bertanggung jawab untuk proses pendaftaran pengguna baru. Metode index() menampilkan halaman registrasi (vw_register), sementara metode process() melakukan validasi terhadap input yang diterima dari formulir registrasi. Validasi mencakup aturan untuk username yang harus unik dan memiliki panjang minimal 4 karakter serta maksimal 20 karakter, password dengan panjang minimal 0 karakter dan maksimal 50 karakter, konfirmasi password

yang harus sesuai dengan password, dan email yang harus valid. Jika validasi gagal, pesan kesalahan ditampilkan dan pengguna diarahkan kembali ke halaman registrasi dengan input sebelumnya dipertahankan. Jika berhasil, password di-hash menggunakan `password_hash()` sebelum disimpan ke database melalui model `UsersModel`, dan pengguna diarahkan ke halaman login. Kontroler ini memastikan bahwa proses registrasi aman dan sesuai dengan aturan validasi yang ditetapkan, serta mengamankan password dengan penggunaan hash sebelum penyimpanan ke database.

AuthFilter.php

Kode filter `AuthFilter` yang terletak di `app/Filters/AuthFilter.php` dalam aplikasi `CodeIgniter` bertujuan untuk memastikan keamanan akses dengan memverifikasi apakah pengguna telah login sebelum mengizinkannya mengakses halaman tertentu. Dalam metode `before()`, filter ini mengecek keberadaan kunci `'user_id'` dalam sesi. Jika tidak ada, pengguna diarahkan kembali ke halaman login menggunakan `redirect()->to('/login')`. Hal ini memastikan bahwa hanya pengguna yang sudah terautentikasi yang dapat mengakses halaman yang dilindungi oleh filter ini, memberikan lapisan keamanan tambahan untuk aplikasi dengan memastikan akses yang sesuai dengan izin pengguna.

vw_add.php

Kode di atas merupakan sebuah template halaman untuk menambah data buku dalam aplikasi `CodeIgniter`, yang diperluas (`extend`) dari layout utama yang disebut `'layouts/main'`. Halaman ini memiliki gaya yang ditentukan secara internal dengan latar belakang gambar dari `Unsplash`, serta pengaturan gaya untuk elemen-elemen seperti kartu (`card`), tombol (`button`), dan input form. Pada bagian konten (`content`), terdapat sebuah form untuk menambahkan buku baru, dengan validasi pesan kesalahan yang ditampilkan jika ada, menggunakan `session()->getFlashdata('error')`. Form ini mengirimkan data ke rute `buku/store` melalui `form_open()` dari `CodeIgniter`. Tombol "Simpan" akan mengirimkan data form, sementara tombol "Kembali" akan mengarahkan pengguna kembali ke halaman utama (`home`). Bagian `JavaScript` ekstra (`extra-js`) memuat library `Summernote` untuk menyediakan editor teks yang memungkinkan pengguna untuk memasukkan konten dengan lebih interaktif. Ini termasuk pengaturan untuk menginisialisasi `Summernote` pada elemen dengan `id post_content`. Dengan ini, halaman ini menyediakan antarmuka yang nyaman dan estetis untuk menambahkan informasi buku dalam aplikasi.

vw_buku.php

Kode di atas adalah sebuah halaman dalam aplikasi web CodeIgniter yang menampilkan daftar buku. Halaman ini diperluas (extend) dari layout utama yang disebut 'layouts/main'. Desain halaman menggunakan CSS untuk menetapkan gaya tata letak dan penampilan elemen-elemen seperti navbar, kartu (card), tombol (button), dan tabel. Latar belakang halaman menggunakan gambar dari Unsplash dengan efek opasitas yang ditambahkan untuk mempertahankan fokus pada konten utama. Navbar di bagian atas halaman menampilkan menu navigasi yang berubah berdasarkan status login pengguna, menampilkan tautan untuk login, registrasi, atau tautan selamat datang dan logout jika pengguna sudah login. Bagian konten halaman berisi daftar buku dalam tabel yang menampilkan judul, penulis, penerbit, dan tahun terbit dari setiap buku. Setiap baris tabel memiliki tombol untuk mengedit dan menghapus buku yang memicu konfirmasi sebelum aksi penghapusan dilakukan. Di bagian bawah tabel, terdapat navigasi halaman (pagination) untuk memudahkan navigasi antara halaman-halaman data. JavaScript tambahan digunakan untuk menambahkan kelas CSS ke elemen-elemen navigasi halaman untuk memperbaiki tampilan paginasi. Dengan ini, halaman menyediakan antarmuka yang responsif dan estetik untuk manajemen data buku dalam aplikasi.

vw_edit.php

Kode di atas merupakan halaman untuk mengedit entri buku dalam aplikasi web menggunakan CodeIgniter. Pertama, halaman ini diperluas (extend) dari layout 'layouts/main' yang menentukan struktur umum dan gaya halaman. Di dalam <style> tag, pengaturan tampilan background, warna teks, dan gaya kartu (card) diatur untuk menciptakan tampilan yang estetik. Dalam bagian konten, sebuah form edit dibuat menggunakan <?= form_open('buku/update/' . \$post['id']); ?> untuk mengirimkan data update ke fungsi update dalam controller. Setiap input form seperti judul, penulis, penerbit, dan tahun terbit sudah diisi dengan nilai yang sudah ada (\$post['judul'], \$post['penulis'], \$post['penerbit'], dan \$post['tahun_terbit']) untuk memastikan informasi yang ada tidak hilang saat pengeditan. Tombol "Update" akan mengirimkan form untuk memperbarui entri buku, sementara tombol "Kembali" akan mengarahkan pengguna kembali ke halaman utama. Di bagian extra-js, digunakan plugin Summernote untuk memperluas kemampuan input teks dengan menambahkan editor WYSIWYG, memungkinkan pengguna untuk mengedit konten dengan lebih mudah. Dengan demikian, halaman ini tidak hanya berfungsi untuk mengedit data buku tetapi juga memperkaya pengalaman pengguna dengan tambahan fitur editor teks yang responsif.

vw_login.php

Kode HTML di atas adalah halaman login untuk aplikasi perpustakaan, dengan pengaturan tampilan dan interaksi yang responsif. Di dalam `<style>` tag, elemen `<body>` ditata agar konten berada di tengah layar dengan latar belakang gambar yang diberikan efek opacity untuk menciptakan nuansa visual yang menarik. Dua div dengan class `.welcome-message` dan `.form-signin` masing-masing menampilkan pesan selamat datang dan form login. Pesan alert awal di `#initial-alert` memberitahukan pengguna untuk login terlebih dahulu dan diatur untuk tampil secara default. Alert kedua menampilkan pesan error jika login gagal dengan menggunakan session Flashdata dari framework CodeIgniter. Form login meminta username dan password dengan desain input yang konsisten dan tombol "Login" yang mengeksekusi proses login. Terdapat juga tombol "Register" yang mengarahkan pengguna ke halaman registrasi untuk membuat akun baru jika belum terdaftar. Halaman ini dirancang agar mudah diakses, informatif, dan responsif terhadap perangkat yang digunakan pengguna.

vw_register.php

Kode HTML di atas adalah halaman registrasi untuk aplikasi perpustakaan, dengan tampilan yang dirancang agar responsif dan menarik. Pada bagian `<style>`, elemen `<body>` dan `<html>` diatur agar konten berada di tengah layar dengan latar belakang gambar yang memiliki efek opacity untuk menambahkan nuansa visual. Dua div dengan class `.welcome-message` dan `.form-register` masing-masing menampilkan pesan selamat datang dan form registrasi. Pesan alert awal di `#initial-alert` memberitahukan pengguna untuk mengisi form pendaftaran, dan diatur untuk tampil secara default. Alert kedua menampilkan pesan error jika proses registrasi gagal dengan menggunakan session Flashdata dari framework CodeIgniter. Form registrasi meminta username, password (dengan konfirmasi), dan alamat email dengan desain input yang konsisten dan tombol "Register" yang mengeksekusi proses registrasi. Halaman ini dirancang agar mudah digunakan, informatif, serta dapat menangani interaksi pengguna dengan baik.

BukuModel.php

Kelas `BukuModel` merupakan model yang digunakan dalam aplikasi berbasis CodeIgniter untuk berinteraksi dengan tabel 'buku' di database. Pada baris pertama, namespace `App\Models` menunjukkan lokasi file kelas ini di dalam struktur direktori aplikasi. Kelas ini meng-extend dari `CodeIgniter\Model`, yang menyediakan berbagai fitur untuk pengelolaan data. Properti `$table` menentukan nama tabel yang akan diakses, yaitu 'buku'. `$allowedFields` mengizinkan kolom mana saja yang dapat diisi dengan data menggunakan model ini, dalam hal ini adalah 'judul', 'penulis', 'penerbit', dan 'tahun_terbit'. Properti `$createdField`, `$updatedField`, dan `$deletedField` menentukan nama kolom untuk penanda waktu pembuatan, pembaruan, dan

penghapusan data. Properti `$validationRules` mendefinisikan aturan validasi untuk setiap kolom, seperti 'judul' harus diisi dan berupa string, 'tahun_terbit' harus berupa angka antara 1800 dan 2024, dan seterusnya. `$skipValidation` diatur false, sehingga validasi data akan diterapkan secara default. Dengan struktur ini, `BukuModel` memfasilitasi operasi CRUD (Create, Read, Update, Delete) dengan pengelolaan validasi data yang ketat sesuai dengan aturan yang telah ditentukan.

UserModel.php

Kelas `UsersModel` merupakan model yang digunakan dalam aplikasi berbasis CodeIgniter untuk berinteraksi dengan tabel 'users' di database. Pada baris pertama, namespace `App\Models` menunjukkan lokasi file kelas ini di dalam struktur direktori aplikasi. Kelas ini meng-extend dari `CodeIgniter\Model`, yang menyediakan berbagai fitur untuk pengelolaan data. Properti `$table` menentukan nama tabel yang akan diakses, yaitu 'users'. Properti `$returnType` diatur sebagai "object", yang menandakan bahwa data hasil query akan dikembalikan dalam bentuk objek. `$useTimestamps` diatur true, sehingga secara otomatis akan menggunakan kolom timestamps untuk menyimpan waktu pembuatan dan pembaruan data. Properti `$allowedFields` mengizinkan kolom mana saja yang dapat diisi dengan data menggunakan model ini, dalam hal ini adalah 'username', 'password', dan 'email'. Dengan struktur ini, `UsersModel` memfasilitasi operasi CRUD (Create, Read, Update, Delete) pada tabel 'users' dengan pengelolaan waktu dan pengisian data yang terkontrol sesuai dengan aturan yang telah ditentukan.

2024-06-03-103310_Users.php

Kelas `Users` adalah migration untuk mengelola struktur tabel 'users' dalam database menggunakan CodeIgniter. Metode `up()` digunakan untuk mendefinisikan struktur tabel dengan menggunakan `$this->forge->addField(...)` untuk menentukan kolom-kolom seperti 'id' sebagai INT dengan auto increment, 'username' sebagai VARCHAR dengan panjang maksimum 255 karakter, 'email' sebagai VARCHAR dengan panjang maksimum 255 karakter, 'password' sebagai TEXT, serta 'created_at' dan 'updated_at' sebagai datetime yang dapat bernilai NULL. Setelah menentukan struktur, `addKey('id', TRUE)` digunakan untuk menambahkan kunci utama pada kolom 'id', dan `createTable('users')` untuk membuat tabel 'users' berdasarkan definisi yang telah disiapkan. Metode `down()` akan menjatuhkan (drop) tabel 'users' jika migration tersebut di-rollback atau dihapus. Migration ini memfasilitasi pengelolaan struktur database secara konsisten dan mudah diubah melalui penggunaan CodeIgniter's Migration class.

2024-06-03-122804_Buku.php

Kelas Buku adalah migration untuk mengelola struktur tabel 'buku' dalam database menggunakan CodeIgniter. Pada metode `up()`, `addField(...)` digunakan untuk menentukan kolom-kolom seperti 'id' dengan tipe BIGINT yang auto increment, 'judul', 'penulis', dan 'penerbit' sebagai VARCHAR dengan panjang maksimum 255 karakter, serta 'tahun_terbit' sebagai YEAR yang dapat bernilai NULL. Setelah menentukan struktur kolom, `addKey('id', TRUE)` digunakan untuk menambahkan kunci utama pada kolom 'id', dan `createTable('buku')` untuk membuat tabel 'buku' berdasarkan definisi tersebut. Metode `down()` akan menjatuhkan (drop) tabel 'buku' jika migration ini di-rollback atau dihapus. Migration ini memudahkan pengelolaan struktur database dalam aplikasi CodeIgniter, memastikan konsistensi dan fleksibilitas dalam pengaturan skema database.

Seeds Buku.php

Kelas Buku adalah sebuah seeder dalam framework CodeIgniter yang digunakan untuk mengisi data awal ke tabel 'buku' dalam database. Pada metode `run()`, digunakan `$this->db->table('buku')->insert(...)` untuk menyisipkan dua baris data ke tabel 'buku'. Baris pertama memasukkan judul 'Laut Bercerita' oleh Leila S. Chudori, diterbitkan oleh Gramedia pada tahun 2017. Baris kedua memasukkan judul 'Tentang Kamu' oleh Tere Liye, diterbitkan oleh Republika pada tahun 2016. Seeder ini membantu dalam menginisialisasi data penting secara otomatis untuk pengujian atau pengembangan aplikasi, mempercepat proses pengisian data pada tahap awal pengembangan.

UserSeeder.php

Kelas UserSeeder merupakan sebuah seeder dalam framework CodeIgniter yang bertanggung jawab untuk menambahkan satu baris data ke tabel 'users' dalam database. Pada metode `run()`, terdapat satu perintah `$this->db->table('users')->insert(...)` yang memasukkan data pengguna dengan username 'Adit', email 'aditya.oktaviari@gmail.com', dan password yang telah di-hash menggunakan fungsi `password_hash()` dengan algoritma default PASSWORD_DEFAULT. Seeder ini digunakan untuk menyediakan data awal pengguna pada aplikasi, memungkinkan pengembang untuk melakukan pengujian dan konfigurasi awal dengan lebih mudah.

perpustakaan.sql

Ini adalah dump SQL untuk database perpustakaan yang mencakup struktur tabel dan data yang terkait dengan buku dan pengguna. Dump dimulai dengan pengaturan SQL_MODE, timezone, dan struktur database yang disesuaikan dengan pengaturan InnoDB dan charset utf8mb4.

Tabel buku memiliki kolom id sebagai primary key dengan tipe data BIGINT UNSIGNED yang diatur untuk auto increment. Tabel ini juga memuat kolom seperti judul, penulis, penerbit, dan tahun_terbit yang digunakan untuk menyimpan informasi terkait buku, seperti judul buku, nama penulis, penerbit, dan tahun terbit.

Tabel migrations digunakan untuk melacak versi migrasi yang telah dijalankan dengan detail seperti id, version, class, group, namespace, time, dan batch.

Tabel users berisi informasi pengguna dengan kolom seperti id sebagai primary key dengan tipe data INT UNSIGNED, username, email, password (yang di-hash untuk keamanan), serta created_at dan updated_at untuk mencatat waktu pembuatan dan pembaruan data pengguna.

Setiap tabel dilengkapi dengan indeks yang sesuai untuk mempercepat pencarian data dan konfigurasi AUTO_INCREMENT untuk mengelola penambahan otomatis pada kolom primary key. Data awal yang dimasukkan ke dalam tabel buku mencakup informasi tentang buku dengan judul "Investasi" yang ditulis oleh Warren Buffet dan diterbitkan oleh AO Production pada tahun 2014. Tabel users juga memiliki data awal dengan informasi pengguna dengan username 'adit', email 'aditya.oktaviari@gmail.com', dan password yang telah di-hash.

Dump ini menyediakan struktur dasar dan data awal yang siap digunakan untuk aplikasi perpustakaan, menggunakan framework atau aplikasi yang mendukung PHP dan database MariaDB/MySQL

TAUTAN GIT

Berikut adalah tautan untuk source code yang telah dibuat.

[aditokta/Pemrograman-Web-2 \(github.com\)](https://github.com/aditokta/Pemrograman-Web-2)