

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 5**



FUNCTION DAN DATABASE

Oleh:

Nursyahna Puteri

NIM. 2210817120007

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN WEB II
MODUL 5

Laporan Praktikum Pemrograman Web II Modul 5: Function dan Database ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Web II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Nursyahna Puteri
NIM : 2210817120007

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

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

DAFTAR ISI

LEMBAR PENGESAHAN.....	2
DAFTAR ISI	3
DAFTAR GAMBAR	5
DAFTAR TABEL	6
SOAL PRAKTIKUM.....	7
A. Source Code Koneksi.php.....	8
B. Pembahasan Koneksi.php	9
C. Source Code Model.php	11
D. Pembahasan Model.php	21
E. Source Code Index.php.....	44
F. Output Program Index.php.....	48
G. Pembahasan Index.php	48
H. Source Code Member.php	58
I. Output Program Member.php	63
J. Pembahasan Member.php	63
K. Source Code FormMember.php.....	78
L. Output Program FormMember.php	83
M. Pembahasan FormMember.php	83
N. Source Code Buku.php	97
O. Output Program Buku.php.....	102
P. Pembahasan Buku.php	102
Q. Source Code FormBuku.php.....	114
R. Output Program FormBuku.php	119
S. Pembahasan FormBuku.php.....	119
T. Source Code Peminjaman.php.....	131
U. Output Program Peminjaman.php.....	137
V. Pembahasan Peminjaman.php	137
W. Source Code FormPeminjaman.php	150
X. Output Program FormPeminjaman.php	158

Y.	Pembahasan FormPeminjaman.php	158
Z.	Tautan Git	172

DAFTAR GAMBAR

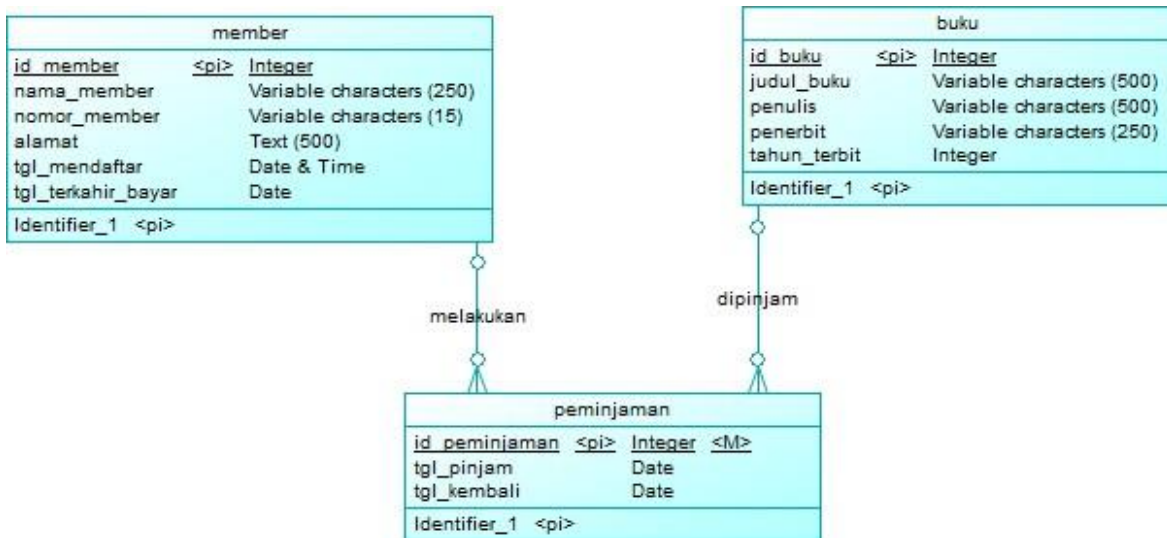
Gambar 1. Screenshot Hasil Jawaban Modul 5 Index.php.....	48
Gambar 2. Screenshot Hasil Jawaban Modul 5 Member.php	63
Gambar 3. Screenshot Hasil Jawaban Modul 5 FormMember.php.....	83
Gambar 4. Screenshot Hasil Jawaban Modul 5 Buku.php	102
Gambar 5. Screenshot Hasil Jawaban Modul 5 FormBuku.php	119
Gambar 6. Screenshot Hasil Jawaban Modul 5 Peminjaman.php.....	137
Gambar 7. Screenshot Hasil Jawaban Modul 5 FormPeminjaman.php	158

DAFTAR TABEL

Table 1. Source Code Modul 5 Koneksi.php.....	8
Table 2. Source Code Modul 5 Model.php	11
Table 3. Source Code Modul 5 Index.php.....	44
Table 4. Source Code Modul 5 Member.php	58
Table 5. Source Code Modul 5 FormMember.php.....	78
Table 6. Source Code Modul 5 Buku.php	97
Table 7. Source Code Modul 5 FormBuku.php.....	114
Table 8. Source Code Modul 5 Peminjaman.php	131
Table 9. Source Code Modul 5 FormPeminjaman.php	150

SOAL PRAKTIKUM

Diberikan desain konseptual sebuah basis data seperti berikut.



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

Adapun ketentuan pembuatannya sebagai berikut:

Adapun ketentuan pembuatannya sebagai berikut:

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

Nama File	Keterangan
Koneksi.php	Berisi fungsi-fungsi untuk melakukan koneksi ke basis data
Model.php	Berisi fungsi-fungsi untuk melakukan operasi data seperti insert, update, delete, get data untuk semua tabel

Member.php	Menampilkan semua data member dalam bentuk tabel dan terdapat tombol untuk melakukan tambah, edit, dan hapus data member.
FormMember.php	Menampilkan formulir isian untuk menambah atau mengubah data member.
Buku.php	Menampilkan semua data buku dalam bentuk tabel dan terdapat tombol untuk melakukan tambah, edit dan hapus data buku.
FormBuku.php	Menampilkan formulir isian untuk menambah atau mengubah data buku.
Peminjaman.php	Menampilkan semua data peminjaman dalam bentuk tabel dan terdapat tombol untuk melakukan tambah, edit, dan hapus data peminjaman.
FormPeminjaman.php	Menampilkan formulir isian untuk menambah atau mengubah data peminjaman.

Simpan dengan nama projek di Github: PRAK501

A. Source Code Koneksi.php

Table 1. Source Code Modul 5 Koneksi.php

1	<?php
2	\$DBHOST = "localhost";
3	\$DBNAME = "praktikum501";
4	\$USERNAME = "root";
5	\$PASSWORD = "";
6	try {
7	\$conn = new PDO(
8	"mysql:host=\$DBHOST;dbname=\$DBNAME",
9	\$USERNAME,
10	\$PASSWORD
11);


```
12 } catch (\Throwable $e) {  
13     echo "Koneksi Gagal, " . $e->getMessage();  
14 }  
15 ?>
```

B. Pembahasan Koneksi.php

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

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

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

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

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

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

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

```
);
```

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

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

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

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

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

C. Source Code Model.php

Table 2. Source Code Modul 5 Model.php

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-
	width, initial-scale=1.0">
8	<link
	href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/di
	st/css/bootstrap.min.css" rel="stylesheet"
9	integrity="sha384-
	1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
	yl2QvZ6jIW3" crossorigin="anonymous">
10	<title>Model</title>
11	</head>
12	
13	<body>
14	</body>
15	
16	</html>
17	<?php
18	include_once("Koneksi.php");
19	
20	function readData(\$nama_tabel)
21	{
22	require "Koneksi.php";
23	

	\$stmt = \$conn->prepare("SELECT * FROM
24	\$nama_tabel");
25	\$stmt->execute();
26	\$result = \$stmt->fetchAll();
27	
28	if (!empty(\$result)) {
29	if (\$nama_tabel == "member") {
30	foreach (\$result as \$hasil) {
31	echo "<tr>";
	echo "<td class='text-center'>" .
32	\$hasil['id_member'] . "</td>";
	echo "<td>" . \$hasil['nama_member'] .
33	"</td>";
	echo "<td>" . \$hasil['nomor_member'] .
	"</td>";
34	echo "<td>" . \$hasil['alamat'] .
	"</td>";
35	echo "<td>" . \$hasil["tgl_mendaftar"]
	. "</td>";
36	echo "<td>" .
	\$hasil["tgl_terakhir_bayar"] . "</td>";
37	echo "<td>";
38	echo "<a class='btn btn-primary'
	href='FormMember.php?id_member="
	\$hasil['id_member'] . "'>Edit";
39	echo " ";
40	echo "<a class='btn btn-danger'
	href='Member.php?id_member=" . \$hasil['id_member'] .
	"' onclick=\"return confirm('Yakin Ingin
	Dihapus?')\">Hapus";
41	echo "</td>";

42	echo "</tr>";
43	}
44	} elseif (\$nama_tabel == "buku") {
45	foreach (\$result as \$hasil) {
46	echo "<tr>";
47	echo "<td class='text-center>" .
	\$hasil['id_buku'] . "</td>";
48	echo "<td>" . \$hasil['judul_buku'] .
	"</td>";
49	echo "<td>" . \$hasil['penulis'] .
	"</td>";
50	echo "<td>" . \$hasil['penerbit'] .
	"</td>";
51	echo "<td>" . \$hasil["tahun_terbit"] .
	"</td>";
52	echo "<td>";
53	echo "<a class='btn btn-primary'
	href='FormBuku.php?id_buku=" . \$hasil['id_buku'] .
	">Edit";
54	echo " ";
55	echo "<a class='btn btn-danger'
	href='Buku.php?id_buku=" . \$hasil['id_buku'] . "
	onclick=\"return confirm('Yakin Ingin
	Dihapus?')\">Hapus";
56	echo "</td>";
57	echo "</tr>";
58	}
59	} elseif (\$nama_tabel == "peminjaman") {
60	require "Koneksi.php";
61	if (isset(\$_GET['id_peminjaman'])) {

62	deletePeminjaman(\$_GET['id_peminjaman']);
63	}
64	foreach (\$result as \$hasil) {
65	echo "<tr>";
66	echo "<td>" . \$hasil["id_peminjaman"]
	. "</td>";
67	echo "<td>" . \$hasil["tgl_pinjam"] .
	"</td>";
68	echo "<td>" . \$hasil["tgl_kembali"] .
	"</td>";
69	foreach (\$dataBuku as \$temp) {
70	if (\$baris['id_buku'] ==
	\$temp['id_buku']) {
71	echo "<td>" .
	\$temp['judul_buku'] . "</td>";
72	}
73	}
74	foreach (\$dataMember as \$temp) {
75	if (\$baris['id_member'] ==
	\$temp['id_member']) {
76	echo "<td>" .
	\$temp['nama_member'] . "</td>";
77	}
78	}
79	echo "<td>";
80	echo "<a class='btn btn-primary'
	href='FormPeminjaman.php?id_peminjaman="
	\$hasil['id_peminjaman'] . ">Edit";
81	echo " ";
82	

	<pre> echo "Hapus"; 84 echo "</td>"; 85 echo "</tr>"; 86 } 87 } 88 } 89 } 90 function insertDataMember(\$nama_member, \$nomor_member, \$alamat, \$tgl_mendaftar, 91 \$tgl_terakhir_bayar) 92 { \$sql = "INSERT INTO `member` (`nama_member`, `nomor_member`, `alamat`, `tgl_mendaftar`, `tgl_terakhir_bayar`) VALUES (:nama_member,:nomor_member,:alamat,:tgl_mendaftar,:t 93 gl_terakhir_bayar)"; 94 require "Koneksi.php"; 95 \$stmt = \$conn->prepare(\$sql); \$result = \$stmt->execute(array(':nama_member' => \$nama_member, ':nomor_member' => \$nomor_member, ':alamat' => \$alamat, ':tgl_mendaftar' => \$tgl_mendaftar, ':tgl_terakhir_bayar' => 96 \$tgl_terakhir_bayar)); 97 if (!empty(\$result)) { 98 header('location:Member.php'); 99 } 100 } </pre>
--	---

101	function insertDataBuku(\$id_buku, \$judul_buku,
102	\$penulis, \$penerbit, \$tahun_terbit)
103	{
	\$sql = "INSERT INTO `buku` (`judul_buku`,
	`penulis`, `penerbit`, `tahun_terbit`) VALUES
104	(:judul_buku,:penulis,:penerbit,:tahun_terbit)";
105	require "Koneksi.php";
106	\$stmt = \$conn->prepare(\$sql);
	\$result = \$stmt->execute(array(':judul_buku' =>
	\$judul_buku, ':penulis' => \$penulis, ':penerbit' =>
107	\$penerbit, ':tahun_terbit' => \$tahun_terbit));
108	if (!empty(\$result)) {
109	header('location:Buku.php');
110	}
111	}
112	
	function insertDataPeminjaman(\$tgl_pinjam,
113	\$tgl_kembali, \$id_buku, \$id_member)
114	{
	\$sql = "INSERT INTO `peminjaman` (`tgl_pinjam`,
	`tgl_kembali`, `id_buku`, `id_member`) VALUES
115	(:tgl_pinjam,:tgl_kembali,:id_buku,:id_member)";
116	require "Koneksi.php";
117	\$stmt = \$conn->prepare(\$sql);
	\$result = \$stmt->execute(array(':tgl_pinjam' =>
	\$tgl_pinjam, ':tgl_kembali' => \$tgl_kembali,
118	':id_buku' => \$id_buku, ':id_member' => \$id_member));
119	if (!empty(\$result)) {
120	header('location:Peminjaman.php');
121	}


```

122 }
123
124 //Edit
125 function editMember()
126 {
127     require "Koneksi.php";
128     $stmt = $conn->prepare("SELECT * FROM member where
129 id_member=" . $_GET["id_member"]);
130     $stmt->execute();
131     $GLOBALS['result'] = $stmt->fetchAll();
132 }
133
134 function editBuku()
135 {
136     require "Koneksi.php";
137     $stmt = $conn->prepare("SELECT * FROM buku where
138 id_buku=" . $_GET["id_buku"]);
139     $stmt->execute();
140     $GLOBALS['result'] = $stmt->fetchAll();
141 }
142
143 function editPeminjaman()
144 {
145     require "Koneksi.php";
146     $stmt = $conn->prepare("SELECT * FROM peminjaman
147 WHERE id_peminjaman =" . $_GET['id_peminjaman']);
148     $stmt->execute();
149     $GLOBALS['result'] = $stmt->fetchAll();
150 }
151
152 //Update

```

	function updateMember(\$id_member, \$nama_member,
	\$nomor_member, \$alamat, \$tgl_mendaftar,
150	\$tgl_terakhir_bayar)
151	{
152	require "Koneksi.php";
153	\$pdo_statement = \$conn->prepare(
	"UPDATE member SET nama_member='" .
	\$nama_member . "', nomor_member='" . \$nomor_member .
	"', alamat='" . \$alamat . "', tgl_mendaftar='" .
	\$tgl_mendaftar . "', tgl_terakhir_bayar='" .
	\$tgl_terakhir_bayar . "' where id_member='" .
154	\$id_member
155);
156	\$result = \$pdo_statement->execute();
157	if (\$result) {
158	header('location:Member.php');
159	}
160	}
161	
	function updateBuku(\$id_buku, \$judul_buku, \$penulis,
162	\$penerbit, \$tahun_terbit)
163	{
164	require "Koneksi.php";
165	\$pdo_statement = \$conn->prepare(
	"UPDATE buku SET judul_buku='" . \$judul_buku .
	"', penulis='" . \$penulis . "', penerbit='" . \$penerbit
	. "', tahun_terbit='" . \$tahun_terbit . "' where
166	id_buku=" . \$id_buku
167);
168	\$result = \$pdo_statement->execute();
169	if (\$result) {

```

170         header('location:Buku.php');
171     }
172 }
173
174 function updatePeminjaman($id_peminjaman,
175 $tgl_pinjam, $tgl_kembali, $id_buku, $id_member)
176 {
177     require "Koneksi.php";
178     $pdo_statement = $conn->prepare(
179         "UPDATE peminjaman SET tgl_pinjam='" .
180 $tgl_pinjam . "', tgl_kembali='" . $tgl_kembali . "',
181 id_buku='" . $id_buku . "', id_member='" . $id_member
182 . "' WHERE id_peminjaman = " . $id_peminjaman
183 );
184 $result = $pdo_statement->execute();
185 if ($result) {
186     header('location:Peminjaman.php');
187 }
188 }
189
190 // Hapus
191 function deleteMember($id_member)
192 {
193     require "Koneksi.php";
194     $stmt = $conn->prepare("DELETE FROM member where
195 id_member=" . $id_member);
196 $result = $stmt->execute();
197 if ($result) {
198     header('location:Member.php');
199 }
200 }

```

```

196 }
197 function deleteBuku($id_buku)
198 {
199     require "Koneksi.php";
200     $stmt = $conn->prepare("DELETE FROM buku where
id_buku=" . $id_buku);
201     $result = $stmt->execute();
202     if ($result) {
203         header('location:Buku.php');
204     }
205 }
206 function deletePeminjaman($id_peminjaman)
207 {
208     require "Koneksi.php";
209     $stmt = $conn->prepare("DELETE FROM peminjaman
WHERE id_peminjaman=" . $id_peminjaman);
210     $result = $stmt->execute();
211     if ($result) {
212         header('location:Peminjaman.php');
213     }
214 }
215
216 function getMember($conn)
217 {
218     $query = $conn->prepare("SELECT * from member");
219     $query->execute();
220     $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
221     return $hasil;
222 }
223
224 function getBuku($conn)

```

```

225 {
226     $query = $conn->prepare("SELECT * from buku");
227     $query->execute();
228     $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
229     return $hasil;
230 }
231
232 function getPeminjaman($conn)
233 {
234     $query = $conn->prepare("SELECT * from
235     peminjaman");
236     $query->execute();
237     $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
238     return $hasil;
239 }

```

D. Pembahasan Model.php

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

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

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

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

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

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

Pada baris [8] [9], syntax

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

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j
IW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS. Terdapat penggunaan atribut penting dalam tag ini yaitu, href=<https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css> digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag `<link>` adalah rel, yang dengan nilai

rel="stylesheet" menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut integrity seperti dalam integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"

berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka akan dilakukan penghitungan hash dari file yang diunduh dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut crossorigin dengan nilai crossorigin="anonymous" digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan. Penggunaan, anonymous berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

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

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

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

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

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

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

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

Pada baris [20] [21], syntax

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

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

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

```
require "Koneksi.php";

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

$stmt->execute();

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

penggunaan baris tersebut adalah sebagai perintah `require "Koneksi.php";` yang digunakan untuk menyertakan file `Koneksi.php` ke dalam skrip saat ini. File `Koneksi.php` berisi definisi koneksi ke database, seperti nama host, nama database, username, password, dan konfigurasi lainnya yang diperlukan untuk mengakses database. Dengan menyertakan file ini, maka dapat menggunakan koneksi database yang telah ditetapkan di berbagai bagian skrip tanpa perlu menuliskan ulang kode koneksi setiap kali dibutuhkan. Selanjutnya, pada baris `$stmt = $conn->prepare("SELECT * FROM $nama_tabel");`, merupakan sebuah pernyataan SQL

disiapkan untuk dieksekusi. Pernyataan SQL yang dipersiapkan ini adalah untuk melakukan seleksi atau SELECT dari tabel yang dinamis, di mana nama tabelnya ditentukan oleh nilai yang diberikan pada parameter \$nama_tabel. Proses persiapan pernyataan ini menggunakan objek PDO (\$conn) yang telah dibuat sebelumnya di file Koneksi.php. Selanjutnya, digunakannya baris \$stmt->execute();, pernyataan SQL yang telah dipersiapkan sebelumnya dijalankan. Lalu penggunaan baris \$result = \$stmt->fetchAll(); digunakan untuk mengambil semua baris hasil dari eksekusi pernyataan SQL sebelumnya dan menyimpannya dalam variabel \$result. Fungsi fetchAll() digunakan untuk mengambil semua baris hasil sebagai array asosiatif yang kemudian dapat digunakan untuk pemrosesan lebih lanjut atau untuk menampilkan data sesuai kebutuhan aplikasi.

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

```
if (!empty($result)) {  
    if ($nama_tabel == "member") {  
        foreach ($result as $hasil) {  
            echo "<tr>";  
            echo "    <td                class='text-center'>" .  
$hasil['id_member'] . "</td>";  
            echo "<td>" . $hasil['nama_member'] . "</td>";  
            echo "    <td>" . $hasil['nomor_member'] .  
"</td>";  
            echo "<td>" . $hasil['alamat'] . "</td>";  
            echo "    <td>" . $hasil["tgl_mendaftar"] .  
"</td>";  
            echo "<td>" . $hasil["tgl_terakhir_bayar"] .  
"</td>";  
            echo "<td>";
```

```

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

        echo " ";

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

        echo "</td>";

        echo "</tr>";

    }

```

penggunaan baris-baris tersebut bertanggung jawab untuk memproses dan menampilkan data yang diambil dari tabel database, khususnya tabel "member". Penggunaan blok kode `if (!empty($result)) {` digunakan untuk mengecek apakah hasil dari query tidak kosong. Fungsi `fetchAll()` mengembalikan array, dan kondisi `!empty($result)` memastikan bahwa array tersebut tidak kosong. Kemudian, `if ($nama_tabel == "member") {` digunakan untuk mengecek apakah nama tabel yang diminta adalah "member". Selanjutnya, `foreach ($result as $hasil) {` melakukan iterasi melalui setiap baris hasil query yang disimpan dalam array `$result`. Setiap elemen dari array tersebut adalah satu baris dari tabel. Penggunaan `echo "<tr>";` membuka baris baru dalam tabel HTML. Baris-baris selanjutnya menggunakan `echo` untuk menambahkan sel tabel yang berisi data anggota: `echo "<td class='text-center'>" . $hasil['id_member'] . "</td>";` menambahkan sel tabel yang berisi ID anggota (`id_member`) dengan teks yang diatur agar berada di tengah (`text-center`), `echo "<td>" . $hasil['nama_member'] . "</td>";` menambahkan sel tabel yang berisi nama anggota (`nama_member`), `echo "<td>" . $hasil['nomor_member'] . "</td>";` menambahkan sel tabel yang berisi nomor anggota (`nomor_member`), `echo "<td>" . $hasil['alamat'] . "</td>";` menambahkan sel tabel yang berisi alamat anggota (`alamat`), `echo "<td>" . $hasil['tgl_mendaftar'] . "</td>";` menambahkan sel tabel yang berisi tanggal pendaftaran anggota (`tgl_mendaftar`), dan `echo "<td>" . $hasil['tgl_terakhir_bayar'] . "</td>";` menambahkan sel tabel yang berisi tanggal terakhir pembayaran anggota.

(`tgl_terakhir_bayar`). Selanjutnya penggunaan `echo "<td>";` membuka sel tabel baru untuk menampung tombol-tombol aksi, yaitu Edit dan Hapus. Tombol Edit ditambahkan dengan kode `echo "Edit";`, yang mengarah ke `FormMember.php` dengan parameter `id_member` yang sesuai dengan ID anggota saat ini. Tombol ini menggunakan kelas Bootstrap `btn btn-primary` untuk styling. Spasi antara tombol Edit dan tombol Hapus ditambahkan dengan `echo " ";`. Tombol Hapus ditambahkan dengan kode `echo "Hapus";`, yang mengarah ke `Member.php` dengan parameter `id_member` yang sesuai dengan ID anggota saat ini. Tombol ini menggunakan kelas Bootstrap `btn btn-danger` untuk styling dan memiliki atribut `onclick` yang menampilkan pesan konfirmasi saat tombol ditekan untuk memastikan pengguna ingin menghapus data tersebut. Selanjutnya, `echo "</td>";` menutup sel tabel yang berisi tombol-tombol aksi, dan `echo "</tr>";` menutup baris tabel HTML.

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

```

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

    foreach ($result as $hasil) {

        echo "<tr>";

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

        echo "<td>" . $hasil['judul_buku'] . "</td>";
        echo "<td>" . $hasil['penulis'] . "</td>";
        echo "<td>" . $hasil['penerbit'] . "</td>";
        echo "<td>" . $hasil["tahun_terbit"] .
"</td>";

        echo "<td>";

```

```

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

        echo " ";

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

        echo "</td>";

        echo "</tr>";

    }

```

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

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

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

```

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

    require "Koneksi.php";

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

        deletePeminjaman($_GET['id_peminjaman']);

    }

    foreach ($result as $hasil) {

        echo "<tr>";

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

"</td>";

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

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

        foreach ($dataBuku as $temp) {

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

{

```

```

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

    }

}

foreach ($dataMember as $temp) {

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

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

    }

}

echo "<td>";

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

echo " ";

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

echo "</td>";

echo "</tr>";

}

}

}

```

```
}
```

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

\$hasil['id_peminjaman'] . ">Edit";, yang mengarah ke FormPeminjaman.php dengan parameter id_peminjaman yang sesuai dengan ID peminjaman saat ini. Tombol ini menggunakan kelas Bootstrap btn btn-primary untuk styling. Spasi antara tombol Edit dan tombol Hapus ditambahkan dengan echo " ";. Tombol Hapus ditambahkan dengan echo "Hapus";, yang mengarah ke Peminjaman.php dengan parameter id_peminjaman yang sesuai dengan ID peminjaman saat ini. Tombol ini menggunakan kelas Bootstrap btn btn-danger untuk styling dan memiliki atribut onclick yang menampilkan pesan konfirmasi saat tombol ditekan untuk memastikan pengguna ingin menghapus data tersebut. Selanjutnya, echo "</td>"; menutup sel tabel yang berisi tombol-tombol aksi, dan echo "</tr>"; menutup baris tabel HTML.

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

```
function      insertDataMember($nama_member,      $nomor_member,
$alamat, $tgl_mendaftar, $tgl_terakhir_bayar)

{

    $sql      =      "INSERT      INTO      `member`      (`nama_member`,
`nomor_member`,      `alamat`,      `tgl_mendaftar`,
`tgl_terakhir_bayar`)      VALUES
(:nama_member,:nomor_member,:alamat,:tgl_mendaftar,:tgl_terak
hir_bayar)";

    require "Koneksi.php";

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

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

    if (!empty($result)) {

        header('location:Member.php');
```



```

    }
}

```

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

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

```

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

    require "Koneksi.php";

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

```

```

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

        if (!empty($result)) {

                header('location:Buku.php');

        }

}

```

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

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

```

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

{

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

        require "Koneksi.php";

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

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

        if (!empty($result)) {

```

```

        header('location:Peminjaman.php');
    }
}

```

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

pada baris [124] [125] [126] [127] [128] [129] [130], syntax

```

function editMember()
{
    require "Koneksi.php";

    $stmt = $conn->prepare("SELECT * FROM member where
id_member=" . $_GET["id_member"]);

    $stmt->execute();

    $GLOBALS['result'] = $stmt->fetchAll();
}

```

penggunaan baris-baris tersebut yaitu editMember digunakan untuk mengambil data anggota (member) berdasarkan ID anggota yang diberikan dalam URL (\$_GET["id_member"]). Dengan menggunakan pernyataan require untuk menyertakan file Koneksi.php, yang diharapkan berisi kode untuk menghubungkan ke database. Ini memastikan bahwa koneksi ke database tersedia untuk eksekusi pernyataan SQL selanjutnya. Lalu, dilakukan persiapan pernyataan SQL untuk memilih semua kolom dari tabel member di mana id_member sama dengan nilai yang diberikan di URL. Langkah tersebut dilakukan dengan tujuan untuk mengambil data anggota yang sesuai dengan ID yang diberikan, sehingga data dapat

ditampilkan atau diedit. Pernyataan SQL dieksekusi menggunakan metode `execute()`. Ini mengirimkan pernyataan SQL ke server database untuk dieksekusi dan menghasilkan hasil yang relevan. Hasil dari eksekusi pernyataan SQL tersebut diambil dengan menggunakan metode `fetchAll()` dan disimpan dalam variabel global `$GLOBALS['result']`. Dengan ini, data anggota yang sesuai dengan ID yang diberikan dapat diakses dan disimpan dalam variabel global ini.

Pada baris [132] [133] [134] [135] [136] [137] [138], syntax function `editBuku()`

```
{  
  
    require "Koneksi.php";  
  
    $stmt = $conn->prepare("SELECT * FROM buku where id_buku=" .  
    $_GET["id_buku"]);  
  
    $stmt->execute();  
  
    $GLOBALS['result'] = $stmt->fetchAll();  
  
}
```

penggunaan baris-baris tersebut sama dengan fungsi `edit member`. Fungsi `editBuku` digunakan untuk mengambil data buku berdasarkan ID buku yang diberikan dalam URL. Penggunaannya dilakukan dengan melakukan query SQL untuk memilih buku yang sesuai dengan ID yang diberikan, kemudian menyimpan hasilnya ke dalam variabel global `$GLOBALS['result']`. Hasil tersebut kemudian dapat digunakan untuk menampilkan data buku pada halaman edit buku.

Pada baris [140] [141] [142] [143] [144] [145] [146], syntax

```
function editPeminjaman()  
  
{  
  
    require "Koneksi.php";  
  
    $stmt = $conn->prepare("SELECT * FROM peminjaman WHERE  
id_peminjaman = " . $_GET['id_peminjaman']);
```

```

$stmt->execute();

$GLOBALS['result'] = $stmt->fetchAll();

}

```

penggunaan baris-baris tersebut sama dengan fungsi edit member. Fungsi editPeminjaman digunakan untuk mengambil data peminjaman berdasarkan ID peminjaman yang diberikan dalam URL. Penggunaannya dilakukan dengan menyusun query SQL untuk memilih data peminjaman yang sesuai dengan ID yang diberikan, kemudian hasilnya disimpan dalam variabel global \$GLOBALS['result']. Hasil tersebut kemudian dapat digunakan untuk menampilkan data pada halaman edit peminjaman.

Pada baris [149] [150] [151] [152] [153] [154] [155] [156] [157] [158] [159], syntax

```

function updateMember($id_member, $nama_member, $nomor_member,
$alamat, $tgl_mendaftar, $tgl_terakhir_bayar)
{
    require "Koneksi.php";

    $pdo_statement = $conn->prepare(

        "UPDATE member SET nama_member='" . $nama_member . "',
        nomor_member='" . $nomor_member . "', alamat='" . $alamat .
        "',      tgl_mendaftar='" . $tgl_mendaftar . "',
        tgl_terakhir_bayar='" . $tgl_terakhir_bayar . "' where
        id_member=" . $id_member

    );

    $result = $pdo_statement->execute();

    if ($result) {

        header('location:Member.php');

    }

}

```

penggunaan baris-baris tersebut merupakan Fungsi updateMember yang bertujuan untuk memperbarui data anggota (member) dalam database dengan nilai-nilai yang baru. Digunakan pernyataan require untuk menyertakan file Koneksi.php, yang diharapkan berisi kode untuk menghubungkan ke database. Hal ini diperlukan untuk memastikan koneksi ke database tersedia sehingga operasi perbaruan data dapat dilakukan. Selanjutnya, pernyataan SQL UPDATE dipersiapkan untuk mengubah data anggota dengan menggunakan nilai baru yang diberikan. Pada tahap ini, nilai baru untuk setiap kolom yang ingin diperbarui telah ditentukan. Kemudian, pernyataan SQL dieksekusi menggunakan objek PDO yang telah dipersiapkan sebelumnya. Proses eksekusi ini akan mengirimkan pernyataan SQL ke server database untuk dieksekusi. Apabila pernyataan SQL berhasil dieksekusi dan mengembalikan nilai true, ini menandakan bahwa proses perbaruan data anggota telah berhasil. Sebagai langkah lanjutannya, pengguna akan diarahkan kembali ke halaman Member.php.

Pada baris [161] [162] [163] [164] [165] [166] [167] [168] [169] [170] [171], syntax function updateBuku(\$id_buku, \$judul_buku, \$penulis, \$penerbit, \$tahun_terbit)

```
{  
  
    require "Koneksi.php";  
  
    $pdo_statement = $conn->prepare(  
        "UPDATE buku SET judul_buku='" . $judul_buku . "',  
        penulis='" . $penulis . "', penerbit='" . $penerbit . "',  
        tahun_terbit='" . $tahun_terbit . "' where id_buku=" . $id_buku  
    );  
  
    $result = $pdo_statement->execute();  
  
    if ($result) {  
        header('location:Buku.php');  
    }  
}
```

penggunaan baris-baris tersebut sama dengan penggunaan fungsi update member. Fungsi updateBuku mengubah data buku dalam database dengan nilai baru yang diberikan. Hal ini dilakukan dengan menyusun pernyataan SQL UPDATE untuk mengubah entri buku berdasarkan ID buku yang diberikan. Setelah pernyataan SQL dieksekusi, pengguna diarahkan kembali ke halaman Buku.php jika pembaruan berhasil.

Pada baris [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183], syntax

```
function      updatePeminjaman($id_peminjaman,      $tgl_pinjam,
$ttl_kembali, $id_buku, $id_member)

{

    require "Koneksi.php";

    $pdo_statement = $conn->prepare(

        "UPDATE peminjaman SET tgl_pinjam='" . $ttl_pinjam .
        "', tgl_kembali='" . $ttl_kembali . "', id_buku='" . $id_buku
        . "', id_member='" . $id_member . "' WHERE id_peminjaman = " .
        $id_peminjaman

    );

    $result = $pdo_statement->execute();

    if ($result) {

        header('location:Peminjaman.php');

    }

}
```

penggunaan dari baris-baris tersebut sama dengan penggunaan fungsi update member. Fungsi updatePeminjaman digunakan untuk mengubah data peminjaman dalam database sesuai dengan nilai-nilai yang baru diberikan. Hal ini dilakukan dengan menyusun pernyataan SQL UPDATE untuk mengubah entri peminjaman berdasarkan ID peminjaman

yang diberikan. Setelah pernyataan SQL dieksekusi, pengguna diarahkan kembali ke halaman Peminjaman.php jika pembaruan berhasil.

Pada baris [186] [187] [188] [189] [190] [191] [192] [193] [194] [195], syntax

```
function deleteMember($id_member)
{
    require "Koneksi.php";

    $stmt = $conn->prepare("DELETE FROM member where
id_member=" . $id_member);

    $result = $stmt->execute();

    if ($result) {
        header('location:Member.php');
    }
}
```

penggunaan baris-baris tersebut merupakan Fungsi deleteMember berperan dalam menghapus data anggota (member) dari database berdasarkan ID member yang diberikan. Fungsi menggunakan pernyataan require untuk menyertakan file Koneksi.php, yang berisi kode untuk menghubungkan ke database. Hal ini diperlukan agar fungsi dapat berinteraksi dengan database untuk menghapus data anggota. Selanjutnya, pernyataan SQL DELETE dipersiapkan untuk menghapus data anggota berdasarkan ID member yang diberikan. Proses ini memastikan bahwa hanya data anggota dengan ID yang sesuai yang akan dihapus dari database. Setelah pernyataan SQL dipersiapkan, langkah selanjutnya adalah mengeksekusi pernyataan tersebut menggunakan objek PDO yang telah dipersiapkan sebelumnya. Proses eksekusi ini akan mengirimkan pernyataan SQL ke server database untuk dieksekusi. Apabila pernyataan SQL berhasil dieksekusi dan mengembalikan nilai true, ini menandakan bahwa proses penghapusan data anggota telah berhasil. Sebagai langkah berikutnya, pengguna akan diarahkan kembali ke halaman Member.php.

pada baris [196] [197] [198] [199] [200] [201] [202] [203] [204], syntax


```

function deleteBuku($id_buku)
{
    require "Koneksi.php";

    $stmt = $conn->prepare("DELETE FROM buku where id_buku=" .
$id_buku);

    $result = $stmt->execute();

    if ($result) {

        header('location:Buku.php');

    }

}

```

penggunaan dari baris-baris tersebut sama dengan penggunaan fungsi delete member. Fungsi deleteBuku bertujuan untuk menghapus data buku dari database berdasarkan ID buku yang diberikan. Langkah-langkahnya termasuk persiapan pernyataan SQL DELETE untuk menghapus entri buku yang sesuai dengan ID buku yang diberikan. Setelah penghapusan berhasil dieksekusi, pengguna akan diarahkan kembali ke halaman Buku.php. Dengan demikian, fungsi ini memungkinkan pengguna untuk menghapus data buku dari database.

Pada baris [205] [206] [207] [208] [209] [210] [211] [212] [213], syntax

```

function deletePeminjaman($id_peminjaman)
{
    require "Koneksi.php";

    $stmt = $conn->prepare("DELETE FROM peminjaman WHERE
id_peminjaman=" . $id_peminjaman);

    $result = $stmt->execute();

    if ($result) {

        header('location:Peminjaman.php');

    }

}

```

```

    }
}

```

Penggunaan baris-baris tersebut sama dengan fungsi delete member. Fungsi deletePeminjaman menghapus data peminjaman dari database berdasarkan ID peminjaman yang diberikan, kemudian mengarahkan pengguna kembali ke halaman Peminjaman.php setelah penghapusan berhasil dieksekusi.

Pada baris [215] [216] [217] [218] [219] [220] [221], syntax

```

function getMember($conn)
{
    $query = $conn->prepare("SELECT * from member");
    $query->execute();
    $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
    return $hasil;
}

```

penggunaan baris-baris tersebut memiliki Fungsi getMember yang memiliki tujuan untuk mengambil semua data anggota dari database. Fungsi mempersiapkan pernyataan SQL untuk memilih semua data dari tabel member. Hal ini dilakukan agar data lengkap dari semua anggota dapat diambil dari database. Setelah pernyataan SQL dipersiapkan, langkah selanjutnya adalah mengeksekusi pernyataan tersebut menggunakan koneksi database yang diberikan sebagai parameter \$conn. Proses eksekusi ini akan mengirimkan pernyataan SQL ke server database untuk dieksekusi. Apabila pernyataan dieksekusi, hasil dari eksekusi pernyataan SQL diambil menggunakan metode fetchAll dari objek statement PDO. Metode ini mengambil semua baris hasil dan mengembalikannya dalam bentuk array. Selanjutnya hasil yang telah diambil dari database dikembalikan oleh fungsi dalam bentuk array asosiatif menggunakan mode PDO::FETCH_ASSOC. Dalam array ini, setiap elemen mewakili satu baris data dari tabel member, di mana kunci array adalah nama kolom dan nilai adalah nilai yang sesuai.

Pada baris [223] [224] [225] [226] [227] [228] [229], syntax

```
function getBuku($conn)
{
    $query = $conn->prepare("SELECT * from buku");
    $query->execute();
    $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
    return $hasil;
}
```

penggunaan baris-baris tersebut memiliki kesamaan dengan fungsi get member. Fungsi getBuku digunakan untuk mengambil semua data buku dari database dan mengembalikan hasil dalam bentuk array asosiatif. Ini memungkinkan pengguna untuk mengakses dan menggunakan data buku dalam kode mereka.

Pada baris [231] [232] [233] [234] [235] [236] [237], syntax

```
function getPeminjaman($conn)
{
    $query = $conn->prepare("SELECT * from peminjaman");
    $query->execute();
    $hasil = $query->fetchAll(PDO::FETCH_ASSOC);
    return $hasil;
}
```

penggunaan baris-baris tersebut memiliki kesamaan dengan fungsi get member. Fungsi getPeminjaman digunakan untuk mengambil semua data peminjaman dari database dan mengembalikan hasil dalam bentuk array asosiatif. Hal ini memungkinkan pengguna untuk mengakses dan menggunakan data peminjaman dalam kode mereka.

E. Source Code Index.php

Table 3. Source Code Modul 5 Index.php

1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
6	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
7	<title>Sistem Informasi Perpustakaan
	Mahasiswa</title>
8	<style>
9	body {
10	background-image: url('library_page.jpg');
11	background-size: cover;
12	background-position: center;
13	color: #000000;
14	font-family: Arial, sans-serif;
15	margin: 0;
16	display: flex;
17	flex-direction: column;
18	align-items: center;
19	justify-content: center;
20	height: 100vh;
21	overflow: hidden;
22	}
23	
24	h2 {
25	text-align: center;
26	animation: fadeInDown 1s ease-in-out;

27	}
28	
29	h1 {
30	font-size: 48px;
31	margin-bottom: 20px;
32	color: #FFF5EE;
33	text-transform: uppercase;
34	letter-spacing: 2px;
35	font-weight: bold;
36	text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
37	animation: scaleIn 1s ease-in-out;
38	}
39	
40	.grid-container {
41	display: grid;
42	grid-template-columns: repeat(3, 1fr);
43	gap: 20px;
44	width: 80%;
45	margin-top: 20px;
46	}
47	
48	.grid-item {
49	background: #FFF8DC;
50	color: #8B4513;
51	border-radius: 21px;
52	padding: 20px;
53	text-align: center;
54	animation: scaleIn 1s ease-in-out;
55	transition: transform 0.3s;
56	}
57	

```
58     .grid-item:hover {
59         transform: scale(1.05);
60     }
61
62     .grid-item img {
63         transition: transform 0.3s;
64         max-width: 100%;
65     }
66
67     .grid-item img:hover {
68         transform: scale(1.1);
69     }
70
71     .grid-item h3 {
72         margin-bottom: 10px;
73         transition: color 0.3s;
74     }
75
76     .grid-item:hover h3 {
77         color: #B22222;
78     }
79
80     @keyframes fadeInDown {
81         0% {
82             opacity: 0;
83             transform: translateY(-20px);
84         }
85         100% {
86             opacity: 1;
87             transform: translateY(0);
88         }
```

```

89         }
90
91     @keyframes scaleIn {
92         0% {
93             opacity: 0;
94             transform: scale(0.8);
95         }
96         100% {
97             opacity: 1;
98             transform: scale(1);
99         }
100    }
101    </style>
102 </head>
103
104 <body>
105     <h1>SISTEM INFORMASI PERPUSTAKAAN MAHASISWA</h1>
106     <div class="grid-container">
107         <div class="grid-item">
108             <h3>Buku</h3>
109             <a href="Buku.php"></a>
111         </div>
112         <div class="grid-item">
113             <h3>Member</h3>
114             <a href="Member.php"></a>
116         </div>
117         <div class="grid-item">
118             <h3>Peminjaman</h3>

```

118	<code></code>
119	<code></div></code>
120	<code></div></code>
121	<code></body></code>
	<code></html></code>

F. Output Program Index.php



Gambar 1. Screenshot Hasil Jawaban Modul 5 *Index.php*

G. Pembahasan Index.php

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

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

Pada baris [3], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa

ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

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

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

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

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

Pada baris [8], syntax `<style>` merupakan tag HTML yang digunakan untuk menentukan aturan gaya (CSS) secara internal dalam dokumen HTML.

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

```
body {
```

```

background-image: url('library_page.jpg');

background-size: cover;

background-position: center;

color: #000000;

font-family: Arial, sans-serif;

margin: 0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

overflow: hidden;

}

```

penggunaan baris-baris tersebut merupakan kode css yang mengatur tampilan elemen `<body>` pada halaman web dengan menggunakan gambar latar belakang "library_page.jpg" dan memastikan gambar mencakup seluruh area konten dengan `background-size: cover;`. `Background-position: center;` menempatkan gambar di tengah horizontal dan vertikal. Warna teks diatur menjadi hitam (`#000000`) dengan font Arial atau sans-serif sebagai cadangan. Margin halaman dihapus dengan `margin: 0;`. Elemen body menjadi flex container dengan `display: flex;` dan konten ditata dalam satu kolom dengan `flex-direction: column;`. `Align-items: center;` dan `justify-content: center;` sehingga memastikan konten terpusat secara horizontal dan vertikal. Tinggi elemen body setara dengan 100% tinggi layar dengan `height: 100vh;` dan `overflow: hidden;` menghilangkan overflow konten. Hal ini memastikan tampilan halaman web yang teratur dengan gambar latar belakang menarik dan konten terpusat.

Pada baris [24] [25] [26] [27], syntax

```
h2 {
```

```

        text-align: center;

        animation: fadeInDown 1s ease-in-out;

    }

```

penggunaan kode CSS tersebut digunakan untuk mengatur gaya tampilan teks dalam elemen <h2>. Teks tersebut akan terpusat secara horizontal dan memiliki efek animasi muncul dengan memudar dan bergeser ke bawah saat ditampilkan dengan durasi 1 detik.

Pada baris [29] [30] [31] [32] [33] [34] [35] [36] [37] [38], syntax

```

h1 {

    font-size: 48px;

    margin-bottom: 20px;

    color: #FFF5EE;

    text-transform: uppercase;

    letter-spacing: 2px;

    font-weight: bold;

    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);

    animation: scaleIn 1s ease-in-out;

}

```

penggunaan kode CSS ini berfungsi untuk mengatur tampilan teks dalam elemen <h1>, dengan setiap properti memiliki perannya masing-masing. Baris ini menggunakan properti font-size menetapkan ukuran font menjadi 48 piksel, menciptakan teks yang besar dan menonjol. Kemudian, margin-bottom menambahkan margin bawah sebesar 20 piksel, memberikan ruang di bagian bawah elemen <h1> untuk memisahkannya dari elemen lain. Selanjutnya, properti color mengubah warna teks menjadi putih kekuningan (#FFF5EE), menyesuaikan tampilan dengan latar belakang atau desain keseluruhan. Teks juga diberi transformasi menjadi huruf kapital dengan properti text-transform, memastikan konsistensi gaya teks. Properti letter-spacing menambahkan jarak antar huruf sebesar 2 piksel, membuat

teks lebih mudah dibaca dan menarik perhatian. Untuk menekankan teks, properti font-weight mengatur ketebalan font menjadi tebal (bold). Efek bayangan teks diberikan oleh properti text-shadow dengan blur 2 piksel dan bayangan berwarna abu-abu kehitaman (rgba(0, 0, 0, 0.5)), menciptakan tampilan teks yang lebih menonjol. Selanjutnya penggunaan properti animation menambahkan animasi bernama scaleIn dengan durasi 1 detik dan efek easing-in-out, memberikan sentuhan dinamis pada teks dengan muncul secara perlahan-lahan dan mulus.

Pada baris [40] [41] [42] [43] [44] [45] [46], syntax

```
.grid-container {  
  
    display: grid;  
  
    grid-template-columns: repeat(3, 1fr);  
  
    gap: 20px;  
  
    width: 80%;  
  
    margin-top: 20px;  
  
}
```

Penggunaan kode CSS ini memiliki tujuan untuk mengatur tata letak dari kontainer grid (.grid-container). Penggunaan properti display: grid;, elemen tersebut diubah menjadi grid container, yang berarti kontennya akan diatur dalam grid. Selanjutnya, properti grid-template-columns: repeat(3, 1fr); menetapkan jumlah kolom grid menjadi 3 dengan lebar yang sama untuk setiap kolom. Nilai 1fr menunjukkan bahwa setiap kolom akan memiliki lebar yang sama dan memenuhi ruang yang tersedia. Kemudian, properti gap: 20px; menetapkan jarak antara setiap item dalam grid sebesar 20 piksel, menciptakan ruang yang konsisten di antara elemen-elemen. Untuk menyesuaikan ukuran grid dengan tata letak halaman, lebar grid container diatur menggunakan properti width: 80%;, sehingga mencakup 80% dari lebar kontainer induknya. Terakhir, margin-top: 20px; ditetapkan untuk memberikan margin atas sebesar 20 piksel, memberikan ruang antara grid container dengan elemen lain di halaman, meningkatkan estetika dan keterbacaan tata letak secara keseluruhan.

Pada baris [48] [49] [50] [51] [52] [53] [54] [55] [56], syntax

```
.grid-item {  
  
    background: #FFF8DC;  
  
    color: #8B4513;  
  
    border-radius: 21px;  
  
    padding: 20px;  
  
    text-align: center;  
  
    animation: scaleIn 1s ease-in-out;  
  
    transition: transform 0.3s;  
  
}
```

penggunaan kode CSS tersebut mengatur tampilan setiap item dengan beberapa aturan gaya yang spesifik. Pertama, `background: #FFF8DC;` menetapkan latar belakang item menjadi warna putih kekuningan. Selanjutnya, `color: #8B4513;` menetapkan warna teks menjadi coklat tua. Efek estetik tambahan diberikan dengan `border-radius: 21px;` yang mengubah sudut elemen menjadi melengkung dengan jari-jari 21 piksel. Untuk memberikan ruang di dalam setiap item, `padding: 20px;` digunakan, menjaga konten dari tepi elemen. Selain itu, `text-align: center;` membuat teks di setiap item rata tengah secara horizontal, menciptakan tampilan yang teratur. Animasi hadir melalui `animation: scaleIn 1s ease-in-out;`, yang membuat setiap item muncul dengan efek perbesaran selama 1 detik dengan transisi yang mulus. Terakhir, `transition: transform 0.3s;` menambahkan efek animasi halus ketika item dihover dengan mengatur durasi transisi untuk perubahan transformasi selama 0.3 detik. Dengan kombinasi aturan-aturan ini, tampilan item-item tersebut dapat disesuaikan dengan presisi dan memberikan pengalaman visual yang menarik.

Pada baris [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78], syntax

```
.grid-item:hover {
```

```

        transform: scale(1.05);
    }

    .grid-item img {
        transition: transform 0.3s;
        max-width: 100%;
    }

    .grid-item img:hover {
        transform: scale(1.1);
    }

    .grid-item h3 {
        margin-bottom: 10px;
        transition: color 0.3s;
    }

    .grid-item:hover h3 {
        color: #B22222;
    }

```

penggunaan kode CSS ini untuk mengatur respons saat pengguna mengarahkan kursor ke atas elemen grid (.grid-item) dan gambar (img) di dalamnya. Ketika kursor diarahkan ke atas item grid, aturan .grid-item:hover { transform: scale(1.05); } membuat item tersebut sedikit membesar sebesar 5% dari ukuran normalnya. Efek transisi diterapkan pada transformasi

gambar di dalam item grid dengan aturan `.grid-item img { transition: transform 0.3s; max-width: 100%; }`, memastikan perubahan ukuran gambar terjadi secara halus dalam waktu 0.3 detik tanpa melebihi lebar maksimum kontainernya. Selain itu, ketika kursor mengarah ke atas gambar di dalam item grid, aturan `.grid-item img:hover { transform: scale(1.1); }` membuat gambar membesar menjadi 110% dari ukuran normalnya. Untuk elemen judul (`h3`) dalam item grid, aturan `.grid-item h3 { margin-bottom: 10px; transition: color 0.3s; }` menetapkan margin bawah sebesar 10 piksel dan memberikan efek transisi pada perubahan warna teks selama 0.3 detik. Lalu, aturan `.grid-item:hover h3 { color: #B22222; }` membuat warna teks judul berubah menjadi merah tua (`#B22222`) dengan transisi warna yang halus saat kursor diarahkan ke atasnya. Pengaturan ini memberikan pengalaman interaktif yang lebih dinamis bagi pengguna.

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

```
@keyframes fadeInDown {  
    0% {  
        opacity: 0;  
        transform: translateY(-20px);  
    }  
    100% {  
        opacity: 1;  
        transform: translateY(0);  
    }  
}  
  
@keyframes scaleIn {  
    0% {
```

```

        opacity: 0;

        transform: scale(0.8);

    }

    100% {

        opacity: 1;

        transform: scale(1);

    }

}

```

penggunaan kode-kode di atas menguraikan dua animasi CSS, yaitu `fadeInDown` dan `scaleIn`, yang didefinisikan melalui aturan `@keyframes`. Animasi `fadeInDown` menghasilkan efek muncul dari atas ke bawah dengan mengubah `opacity` (ketidaktransparanan) dan melakukan translasi (pemindahan posisi vertikal). Pada titik awal (0% dari animasi), elemen memiliki `opacity 0` dan translasi `-20px` ke atas (dimulai dari posisi 20 piksel di atas posisi akhirnya). Pada titik akhir (100% dari animasi), `opacity` elemen mencapai 1 (tidak transparan) dan translasi kembali ke posisi normalnya (0). Animasi `scaleIn`, di sisi lain, menghasilkan efek muncul dari ukuran yang lebih kecil menjadi ukuran normal dengan mengubah `opacity` dan melakukan transformasi skala (perubahan ukuran). Pada titik awal (0% dari animasi), elemen memiliki `opacity 0` (sepenuhnya transparan) dan skala 0.8 (80% dari ukuran normal). Pada titik akhir (100% dari animasi), `opacity` elemen mencapai 1 (tidak transparan) dan skala kembali ke ukuran normal (1).

Pada baris [101], syntax `</style>` merupakan tag penutup di baris 8.

Pad baris [102], syntax `</head>` merupakan tag penutup dari baris 3.

Pada baris [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118][119] [120], syntax

`<body>`

`<h1>SISTEM INFORMASI PERPUSTAKAAN MAHASISWA</h1>`


```

<div class="grid-container">

    <div class="grid-item">

        <h3>Buku</h3>

        <a href="Buku.php"></a>

    </div>

    <div class="grid-item">

        <h3>Member</h3>

        <a href="Member.php"></a>

    </div>

    <div class="grid-item">

        <h3>Peminjaman</h3>

        <a href="Peminjaman.php"></a>

    </div>

</div>

</body>

```

penggunaan baris-baris tersebut digunakan untuk menyusun struktur dasar sebuah halaman web. Elemen <body> berfungsi sebagai wadah untuk semua konten yang akan ditampilkan. Judul utama halaman, <h1>, memuat teks "SISTEM INFORMASI PERPUSTAKAAN MAHASISWA". Selanjutnya, terdapat sebuah div dengan kelas "grid-container" yang digunakan untuk mengatur tata letak elemen-elemen di dalamnya menggunakan grid CSS. Di dalam grid container tersebut, terdapat dua div dengan kelas "grid-item" yang menampilkan informasi tentang "Buku" dan "Member". Setiap elemen <div class="grid-item"> diikuti dengan judul level 3, <h3>, yang berisi teks "Buku", "Member", dan

"Peminjaman" sebagai label untuk setiap item dalam grid. Elemen anchor, ``, mengarahkan pengguna ke halaman "Buku.php" saat diklik. Terakhir, terdapat elemen gambar, ``, yang menampilkan gambar "book.svg" dengan teks alternatif "Buku" dan tinggi 170 piksel.

Pada baris [121], syntax `</html>` baris ini merupakan tag penutup HTML. Tag ini digunakan setelah dua tag penting yaitu tag Head dan tag Body telah digunakan pada baris sebelumnya.

H. Source Code Member.php

Table 4. Source Code Modul 5 *Member.php*

1	<code><?php</code>
2	<code>include_once("Koneksi.php");</code>
3	
4	<code>function deleteMember(\$id_member)</code>
5	<code>{</code>
6	<code> global \$conn;</code>
7	<code> \$stmt = \$conn->prepare("DELETE FROM member WHERE</code>
	<code>id_member = ?");</code>
8	<code> \$stmt->execute([\$id_member]);</code>
9	<code> header("Location: Member.php");</code>
10	<code> exit();</code>
11	<code>}</code>
12	
13	<code>function readData(\$nama_tabel)</code>
14	<code>{</code>
15	<code> global \$conn;</code>
16	<code> \$stmt = \$conn->prepare("SELECT * FROM \$nama_tabel");</code>
17	<code> \$stmt->execute();</code>
18	<code> \$result = \$stmt->fetchAll();</code>
19	
20	<code> if (!empty(\$result)) {</code>
21	<code> if (\$nama_tabel == "member") {</code>
22	<code> foreach (\$result as \$hasil) {</code>
23	<code> echo "<tr>";</code>
24	<code> echo " <td class='text-center'>" .</code>
	<code>\$hasil['id_member'] . "</td>";</code>
25	<code> echo "<td>" . \$hasil['nama_member'] .</code>
	<code>"</td>";</code>

```

26         echo "<td>" . $hasil['nomor_member'] .
"</td>";
27         echo "<td>" . $hasil['alamat'] .
"</td>";
28         echo "<td>" . $hasil["tgl_mendaftar"] .
"</td>";
29         echo "<td>" .
$hasil["tgl_terakhir_bayar"] . "</td>";
30         echo "<td>";
31         echo "<a class='btn btn-primary'
href='FormMember.php?id_member=" . $hasil['id_member']
. "'>Edit</a>";
32         echo " ";
33         echo "<a class='btn btn-danger'
href='Member.php?id_member=" . $hasil['id_member'] . "'
onclick=\"return confirm('Yakin Ingin
Dihapus?')\">Hapus</a>";
34         echo "</td>";
35         echo "</tr>";
36     }
37 }
38 }
39 }
40 ?>
41
42 <!DOCTYPE html>
43 <html lang="en">
44 <head>
45     <meta charset="UTF-8">
46     <meta http-equiv="X-UA-Compatible"
content="IE=edge">
47     <meta name="viewport" content="width=device-width,
initial-scale=1.0">
48     <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dis
t/css/bootstrap.min.css" rel="stylesheet"
49     integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy
l2QvZ6jIW3" crossorigin="anonymous">
50     <title>Data Member</title>
51     <style>

```

```

52     body {
53         background-image: url('library_page.jpg');
54         background-size: cover;
55         background-position: center;
56         color: #ffffff;
57         font-family: Arial, sans-serif;
58         margin: 0;
59         display: flex;
60         flex-direction: column;
61         align-items: center;
62         justify-content: center;
63         overflow: auto;
64     }
65     table,
66     tr,
67     td {
68         border-collapse: collapse;
69         border-radius: 21px;
70         padding: 20px 40px;
71         font-size: 18px;
72     }
73     table {
74         width: max-content;
75         border-radius: 21px;
76         background: rgba(255, 255, 255, 0.8);
77         color: black;
78     }
79     td {
80         width: 100px;
81         height: 10px;
82         text-align: center;
83     }
84     .buttons-container {
85         display: flex;
86         justify-content: flex-end;
87         margin-bottom: 20px;
88     }
89     .buttons-container a:first-child {
90         margin-right: 10px;
91     }
92     button {

```

```

93         border-radius: 10px;
94         border: 1px solid #03045e;
95         position: relative;
96         overflow: hidden;
97         transition: all 0.5s ease-in;
98         z-index: 1;
99     }
100     button::before,
101     button::after {
102         content: '';
103         position: absolute;
104         top: 0;
105         width: 0;
106         height: 100%;
107         transform: skew(15deg);
108         transition: all 0.5s;
109         overflow: hidden;
110         z-index: -1;
111     }
112     button::before {
113         left: -10px;
114         background: #537188;
115     }
116     button::after {
117         right: -10px;
118         background: #537188;
119     }
120     button:hover::before,
121     button:hover::after {
122         width: 70%;
123     }
124     button:hover span {
125         color: #e0aaff;
126         transition: 0.3s;
127     }
128     button span {
129         color: #03045e;
130         font-size: 18px;
131         transition: all 0.3s ease-in;
132     }
133     h1 {

```

```

134         font-size: 48px;
135         margin: 40px auto 20px;
136         color: #FFFAF0;
137         text-transform: uppercase;
138         letter-spacing: 2px;
139         font-weight: bold;
140         text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
141         animation: fadeInDown 1s ease-in-out,
scaleIn 1s ease-in-out;
142         border-bottom: 2px solid #FFFAF0;
143         padding-bottom: 10px;
144         background-color: rgba(0, 0, 0, 0.5);
145         width: 80%;
146         text-align: center;
147     }
148 </style>
149 </head>
150 <body class="p-3" style="background-color: #E1D4BB;">
151     <?php
152         if (isset($_GET['id_member'])) {
153             deleteMember($_GET['id_member']);
154         }
155     ?>
156     <h2>
157         <center><h1>Data Member</h1></center>
158     </h2>
159     <div class="buttons-container">
160         <a href="index.php"><button class="btn btn-dark
mb-4">Kembali</button></a>
161         <a href="FormMember.php"><button class="btn
btn-warning mb-1">Tambah Data Member</button></a>
162     </div>
163     <table class="table">
164         <thead class="table-dark">
165             <tr>
166                 <th class="text-center">ID Member</th>
167                 <th class="text-center">Nama
Member</th>
168                 <th class="text-center">Nomor
Member</th>
169                 <th class="text-center">Alamat</th>

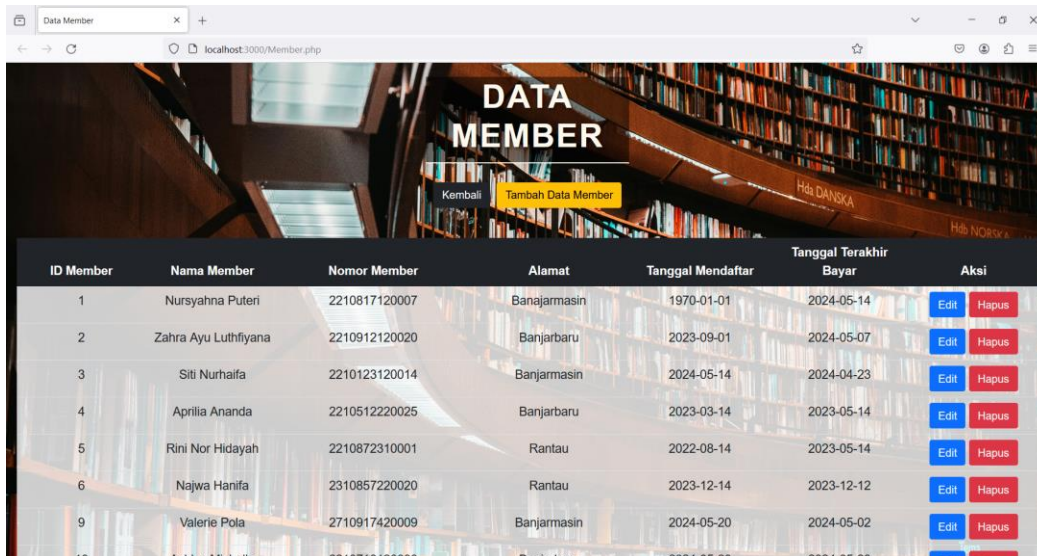
```

```

170         <th class="text-center">Tanggal
Mendaftar</th>
171         <th class="text-center">Tanggal
Terakhir Bayar</th>
172         <th class="text-center">Aksi</th>
173     </tr>
174 </thead>
175 <tbody>
176     <?php readData("member"); ?>
177 </tbody>
178 </table>
179 </body>
180 </html>

```

I. Output Program Member.php



ID Member	Nama Member	Nomor Member	Alamat	Tanggal Mendaftar	Tanggal Terakhir Bayar	Aksi
1	Nursyahna Puteri	2210817120007	Banjarmasin	1970-01-01	2024-05-14	Edit Hapus
2	Zahra Ayu Luthfiyana	2210912120020	Banjarbaru	2023-09-01	2024-05-07	Edit Hapus
3	Siti Nurhalifa	2210123120014	Banjarmasin	2024-05-14	2024-04-23	Edit Hapus
4	Aprilia Ananda	2210512220025	Banjarbaru	2023-03-14	2023-05-14	Edit Hapus
5	Rini Nor Hidayah	2210872310001	Rantau	2022-08-14	2023-05-14	Edit Hapus
6	Najwa Hanifa	2310857220020	Rantau	2023-12-14	2023-12-12	Edit Hapus
9	Valerie Pola	2710917420009	Banjarmasin	2024-05-20	2024-05-02	Edit Hapus
10	Ashliu Michella	2210748120000	Banjarmasin	2024-05-20	2024-05-22	Edit Hapus

Gambar 2. Screenshot Hasil Jawaban Modul 5 *Member.php*

J. Pembahasan Member.php

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

Pada baris [2], syntax `include_once("Koneksi.php");` baris ini adalah konstruksi bahasa dalam PHP yang digunakan untuk menyertakan dan mengevaluasi file yang ditentukan selama eksekusi skrip. `include_once` memastikan bahwa file hanya disertakan satu kali.

Apabila file sudah disertakan sebelumnya, maka `include_once` akan mengabaikannya panggilan yang dilakukan lagi. Pada program ini, file yang akan disertakan adalah `Koneksi.php`, yang berisi kode untuk mengatur koneksi ke database. Dengan menggunakan `include_once`, akan memastikan bahwa kode dalam `Koneksi.php` hanya dieksekusi sekali, bahkan jika `include_once` dipanggil beberapa kali dalam skrip yang sama. Tujuan dari penggunaannya adalah untuk mencegah potensi masalah seperti pengaturan ulang koneksi database atau definisi ulang fungsi dan variabel yang sudah ada.

Pada baris [4] [5] [6] [7] [8] [9] [10] [11], syntax

```
function deleteMember($id_member)

{

    global $conn;

    $stmt = $conn->prepare("DELETE FROM member WHERE id_member
= ?");

    $stmt->execute([$id_member]);

    header("Location: Member.php");

    exit();

}
```

penggunaan kode-kode di atas merupakan sebuah fungsi PHP yang bertujuan untuk menghapus data member dari database. Fungsi ini dideklarasikan dengan nama `deleteMember` yang menerima parameter `$id_member` sebagai identifikasi unik untuk data member yang akan dihapus. Dalam fungsi tersebut, `global $conn;` digunakan untuk mengakses variabel `$conn` yang merupakan koneksi ke database yang sudah didefinisikan di tempat lain dalam program. Pernyataan SQL yang disiapkan (prepared statement) "`$stmt = $conn->prepare('DELETE FROM member WHERE id_member = ?');`" bertujuan untuk menghapus data member dari tabel "member" berdasarkan `$id_member` yang diberikan. Setelah pernyataan SQL tersebut disiapkan, eksekusi dilakukan dengan "`$stmt->execute([$id_member]);`", di mana nilai dari `$id_member` disertakan untuk dihapus. Setelah proses penghapusan selesai, pengguna diarahkan kembali ke halaman "`Member.php`"

menggunakan fungsi `header("Location: Member.php");`. Lalu, `exit();` digunakan untuk menghentikan eksekusi script PHP selanjutnya setelah melakukan pengalihan halaman, sehingga proses di sini berhenti dan tidak ada kode tambahan yang dijalankan setelahnya.

Pada baris [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40], syntax

```
function readData($nama_tabel)
{
    global $conn;

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

    if (!empty($result)) {
        if ($nama_tabel == "member") {
            foreach ($result as $hasil) {
                echo "<tr>";
                echo "    <td        class='text-center'>" .
$hasil['id_member'] . "</td>";
                echo "<td>" . $hasil['nama_member'] . "</td>";
                echo "    <td>" . $hasil['nomor_member'] .
"</td>";
                echo "<td>" . $hasil['alamat'] . "</td>";
                echo "    <td>" . $hasil["tgl_mendaftar"] .
"</td>";
```

```

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

        echo "<td>";

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

        echo " ";

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

        echo "</td>";

        echo "</tr>";

    }

}

}

}

?>

```

penggunaan kode-kode tersebut merupakan Fungsi readData, dimana berguna untuk membaca data dari sebuah tabel yang disebutkan dalam parameter \$nama_tabel. Proses ini dimulai dengan akses global terhadap variabel \$conn, yang merupakan koneksi ke database yang telah didefinisikan sebelumnya dalam program. Selanjutnya, sebuah pernyataan SQL yang telah disiapkan ("SELECT * FROM \$nama_tabel") digunakan untuk memilih semua kolom dari tabel yang ditentukan. Eksekusi dari pernyataan SQL ini dilakukan dengan "\$stmt->execute();". Hasil dari query tersebut kemudian diambil dan disimpan dalam sebuah array menggunakan fungsi "\$result = \$stmt->fetchAll();". Sebelum melakukan iterasi, dilakukan pengujian untuk memastikan bahwa ada hasil yang ditemukan dengan menggunakan if (!empty(\$result)) {...}. Setelah itu, dilakukan iterasi melalui hasil query

yang ditemukan menggunakan loop foreach. Setiap baris hasil dari query diakses dalam loop foreach dengan nama \$hasil. Dalam setiap iterasi, data dari setiap baris hasil ditampilkan dalam bentuk baris tabel HTML. Selain itu, juga ditampilkan link untuk mengedit dan menghapus data, yang mengarahkan pengguna ke halaman FormMember.php untuk proses edit dan ke halaman Member.php untuk proses hapus. Dengan fungsi ini, memungkinkan untuk menampilkan data dari tabel yang spesifik dalam bentuk tabel HTML di halaman web.

Pada baris [42], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisi informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [43], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [44], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa di tambahkan pada tag ini.

Pada baris [45], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [46], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [47], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [48] [49], syntax

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

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j
IW3" crossorigin="anonymous">
```

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

Pada baris [50], syntax <title>Data Member</title> tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [51], syntax <style> merupakan tag HTML yang digunakan untuk menentukan aturan gaya (CSS) secara internal dalam dokumen HTML.

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

```
body {  
  
    background-image: url('library_page.jpg');  
  
    background-size: cover;  
  
    background-position: center;  
  
    color: #ffffff;  
  
    font-family: Arial, sans-serif;  
  
    margin: 0;  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    justify-content: center;  
  
    overflow: auto;  
  
}
```

penggunaan baris-baris tersebut merupakan kode css yang mengatur tampilan elemen <body> pada halaman web dengan menggunakan gambar latar belakang "library_page.jpg" dan memastikan gambar mencakup seluruh area konten dengan background-size: cover;. Background-position: center; menempatkan gambar di tengah horizontal dan vertikal. Warna teks pada halaman web ditetapkan menjadi putih (#ffffff) melalui properti color: #ffffff;. Font

teks ditentukan sebagai Arial atau font tanpa serifikasi dengan properti font-family: Arial, sans-serif;. Selain itu, margin: 0; menghapus margin yang ada secara default di sekitar elemen <body>. Dengan display: flex; digunakan untuk menerapkan model tata letak flexbox, posisi dan tata letak elemen-elemen dalam <body> diatur. flex-direction: column; mengatur arah tata letak flexbox menjadi kolom, sehingga elemen-elemen anak disusun secara vertikal. Properti align-items: center; dan justify-content: center; menetapkan penempatan horizontal dan vertikal elemen-elemen anak di tengah <body>. Lalu, overflow: auto; menentukan perilaku tampilan overflow untuk elemen <body>, yang berarti ketika konten lebih besar dari layar, scrollbar akan muncul untuk memungkinkan pengguna menggulir konten.

Pada baris [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75][76] [77] [78] [79][80] [81] [82] [83], syntax

```
table,

    tr,

    td {

        border-collapse: collapse;

        border-radius: 21px;

        padding: 20px 40px;

        font-size: 18px;

    }

table {

    width: max-content;

    border-radius: 21px;

    background: rgba(255, 255, 255, 0.8);

    color: black;

}
```

```

td {

    width: 100px;

    height: 10px;

    text-align: center;

}

```

penggunaan kode CSS tersebut bertujuan untuk menetapkan gaya elemen-elemen HTML <table>, <tr>, dan <td>. Setiap properti CSS memiliki peran spesifik dalam menentukan penampilan dan perilaku dari elemen-elemen tersebut. Penggunaan border-collapse: collapse; menggabungkan batas sel-sel tabel agar terlihat seperti satu batas tunggal. Properti border-radius: 21px; memberikan sudut yang dibulatkan pada elemen tabel, baris, dan sel, dengan sudut dibulatkan menjadi 21 piksel. Padding: 20px 40px; menambahkan jarak antara isi sel dan batas sel, dengan angka pertama mengatur padding atas dan bawah, dan angka kedua mengatur padding kanan dan kiri. Ukuran font teks di dalam sel tabel ditentukan dengan font-size: 18px;. Properti width: max-content; menyesuaikan lebar tabel dengan ukuran konten maksimum di dalamnya, memungkinkan tabel memperluas sejauh yang diperlukan untuk menampung konten tanpa memotongnya. Warna latar belakang tabel ditentukan dengan background: rgba(255, 255, 255, 0.8); menggunakan format RGBA, dengan nilai transparansi 0.8 untuk memberikan efek transparansi. Warna teks di dalam sel tabel ditetapkan menjadi hitam dengan color: black;. Lebar dan tinggi setiap sel tabel ditetapkan secara khusus dengan width: 100px; dan height: 10px;, masing-masing menjadi 100 piksel dan 10 piksel. Lalu, text-align: center; mengatur teks di dalam sel tabel menjadi rata tengah secara horizontal.

Pada baris [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132], syntax

```

.buttons-container {

    display: flex;

```

```

        justify-content: flex-end;

        margin-bottom: 20px;
    }

    .buttons-container a:first-child {

        margin-right: 10px;
    }

    button {

        border-radius: 10px;

        border: 1px solid #03045e;

        position: relative;

        overflow: hidden;

        transition: all 0.5s ease-in;

        z-index: 1;
    }

    button::before,
    button::after {

        content: '';

        position: absolute;

        top: 0;

        width: 0;

        height: 100%;

        transform: skew(15deg);

        transition: all 0.5s;
    }

```



```

        overflow: hidden;

        z-index: -1;
    }

    button::before {

        left: -10px;

        background: #537188;
    }

    button::after {

        right: -10px;

        background: #537188;
    }

    button:hover::before,
    button:hover::after {

        width: 70%;
    }

    button:hover span {

        color: #e0aaff;

        transition: 0.3s;
    }

    button span {

        color: #03045e;

        font-size: 18px;

        transition: all 0.3s ease-in;
    }

```

```
}
```

penggunaan kode-kode CSS tersebut digunakan untuk menetapkan gaya khusus pada elemen HTML yang memiliki kelas tertentu. Pada, kelas `.buttons-container` bertujuan untuk mengatur gaya sebuah kontainer yang berisi tombol-tombol. Dengan penggunaan properti `display: flex;`, kontainer tersebut diubah menjadi flex container, sehingga penataan fleksibel terhadap tombol-tombol di dalamnya. Properti `justify-content: flex-end;` digunakan untuk menempatkan tombol-tombol di ujung kanan kontainer. Selector `.buttons-container a:first-child` mengarahkan pada link pertama di dalam kontainer tombol. Properti `margin-right: 10px;` memberikan jarak 10 piksel di sebelah kanan link pertama. Gaya dari tombol itu sendiri ditentukan oleh properti `button`, termasuk `border-radius: 10px;` yang memberikan sudut yang dibulatkan pada tombol, serta `border: 1px solid #03045e;` yang menetapkan tepi tombol menjadi garis solid dengan warna tertentu. Pseudoelemen `button::before` dan `button::after` menambahkan elemen sebelum dan sesudah isi tombol, dengan properti yang mengatur penampilannya seperti warna dan lebar. Properti-properti pada `button:hover::before` dan `button:hover::after` mengontrol penampilan elemen sebelum dan sesudah tombol saat tombol dihover. Properti `button:hover span` mengatur penampilan teks di dalam tombol saat dihover, sedangkan `button span` mengatur penampilan teks di dalam tombol secara umum, termasuk ukuran font dan efek transisi.

Pada baris [133] [134] [135] [136] [137] [138] [139] [140] [141] [142] [143] [144] [145] [146] [147], syntax

```
h1 {  
  
    font-size: 48px;  
  
    margin: 40px auto 20px;  
  
    color: #FFFAF0;  
  
    text-transform: uppercase;  
  
    letter-spacing: 2px;  
  
    font-weight: bold;  
  
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);  
}
```

```

        animation: fadeInDown 1s ease-in-out, scaleIn 1s
ease-in-out;

        border-bottom: 2px solid #FFFAF0;

        padding-bottom: 10px;

        background-color: rgba(0, 0, 0, 0.5);

        width: 80%;

        text-align: center;

    }

```

penggunaan kode-kode CSS tersebut bertujuan untuk menetapkan gaya khusus pada elemen `<h1>` (judul utama) dalam sebuah halaman web. Tiap properti CSS memiliki peran tertentu dalam menentukan penampilan dan perilaku dari elemen tersebut. Pertama, `font-size: 48px;` mengatur ukuran font teks menjadi 48 piksel. Properti `margin: 40px auto 20px;` menetapkan margin sebesar 40 piksel di sisi atas dan bawah, dengan margin otomatis di sisi kiri dan kanan. Warna teks ditentukan menjadi putih dengan `color: #FFFAF0;`. `Text-transform: uppercase;` mengubah semua huruf teks menjadi huruf kapital, sementara `letter-spacing: 2px;` menambahkan jarak antara setiap huruf sebesar 2 piksel. `Font-weight: bold;` memberikan efek tebal pada teks. Properti `text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);` menambahkan bayangan pada teks dengan pergeseran 2 piksel ke kanan dan 2 piksel ke bawah, radius bayangan 4 piksel, dan warna bayangan yang ditentukan oleh nilai RGBA (hitam dengan opasitas 50%). `Animation: fadeInDown 1s ease-in-out, scaleIn 1s ease-in-out;` mengatur animasi dengan efek `fadeInDown` (perubahan opasitas dari 0 menjadi 1 sambil turun) selama 1 detik dengan gerakan yang halus (`ease-in-out`), dan `scaleIn` (perbesaran elemen) selama 1 detik dengan gerakan yang halus (`ease-in-out`). Properti `border-bottom: 2px solid #FFFAF0;` menambahkan garis bawah dengan ketebalan 2 piksel dan warna putih, sementara `padding-bottom: 10px;` menambahkan ruang tambahan sebesar 10 piksel di bagian bawah elemen. `Background-color: rgba(0, 0, 0, 0.5);` mengatur warna latar belakang menjadi hitam dengan opasitas 50%. Lebar elemen ditentukan menjadi 80% dari lebar kontainer induk dengan `width: 80%;`, dan `text-align: center;` digunakan untuk memusatkan teks secara horizontal di tengah elemen.

Pada baris [148], syntax `</style>` merupakan tag penutup dari baris 51.

Pada baris [149], syntax `</head>` merupakan tag penutup dari baris 44.

Pada baris [150] [151] [152] [153] [154] [155] [156] [157] [158] [159] [160] [161] [162] [163] [164] [165] [166] [167] [168] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178], syntax

```
<body class="p-3" style="background-color: #E1D4BB;">

    <?php

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

            deleteMember($_GET['id_member']);

        }

    ?>

    <h2>

        <center><h1>Data Member</h1></center>

    </h2>

    <div class="buttons-container">

        <a href="index.php"><button class="btn btn-dark mb-4">Kembali</button></a>

        <a href="FormMember.php"><button class="btn btn-warning mb-1">Tambah Data Member</button></a>

    </div>

    <table class="table">

        <thead class="table-dark">

            <tr>

                <th class="text-center">ID Member</th>
```

```

        <th class="text-center">Nama Member</th>

        <th class="text-center">Nomor Member</th>

        <th class="text-center">Alamat</th>

        <th class="text-center">Tanggal
Mendaftar</th>

        <th class="text-center">Tanggal Terakhir
Bayar</th>

        <th class="text-center">Aksi</th>

    </tr>

</thead>

<tbody>

    <?php readData("member"); ?>

</tbody>

</table>

```

penggunaan kode tersebut bertujuan menampilkan data member dalam bentuk table. Penggunaan atribut kelas Bootstrap p-3 diterapkan pada elemen <body> untuk memberikan padding sebesar 3 unit (0.75 rem) di sekitar konten. Selain itu, properti inline style digunakan untuk menetapkan warna latar belakang halaman dengan kode warna #E1D4BB, menciptakan tampilan yang estetik dan terstruktur. Kemudian, dilakukan pengecekan apakah terdapat parameter id_member dalam URL. Jika parameter tersebut ditemukan, fungsi deleteMember() akan dipanggil untuk menghapus data member sesuai dengan id yang disediakan, menunjukkan fungsionalitas untuk penghapusan data yang terintegrasi dengan halaman. Judul "Data Member" ditampilkan di tengah halaman menggunakan elemen <h1> yang diletakkan di dalam elemen <center>, memberikan fokus visual yang jelas pada konten utama halaman. Selanjutnya, sebuah div dengan kelas buttons-container dibuat untuk menampung dua tombol aksi: "Kembali" dan "Tambah Data Member". Tombol-tombol tersebut mengarahkan pengguna ke halaman index.php dan FormMember.php masing-

masingnya, memastikan navigasi yang mudah dan intuitif bagi pengguna. Tabel dibuat dengan menggunakan kelas Bootstrap table untuk menampilkan data member dengan struktur yang teratur. Kolom-kolom seperti ID Member, Nama Member, Nomor Member, Alamat, Tanggal Mendaftar, Tanggal Terakhir Bayar, dan Aksi disertakan, memberikan informasi yang komprehensif kepada pengguna. Terakhir, fungsi `readData()` dipanggil untuk membaca data member dari database dan menampilkan data tersebut dalam bentuk baris-baris tabel. Setiap baris data ditampilkan menggunakan fungsi `echo` untuk setiap baris data yang diperoleh dari panggilan SQL, memastikan bahwa data ditampilkan dengan benar sesuai dengan yang ada di dalam database.

Pada baris [179], syntax `</body>` merupakan tag penutup dari baris 150.

Pada baris [180], syntax `</html>` merupakan tag penutup dari baris 43.

K. Source Code FormMember.php

Table 5. Source Code Modul 5 *FormMember.php*

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	<code> <meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>
6	<code> <meta name="viewport" content="width=device-width,</code>
	<code>initial-scale=1.0"></code>
7	<code> <link</code>
	<code>href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dis</code>
	<code>t/css/bootstrap.min.css" rel="stylesheet"</code>
8	<code> integrity="sha384-</code>
	<code>1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy</code>
	<code>l2QvZ6jIW3" crossorigin="anonymous"></code>
9	<code> <title>Form Member</title></code>
10	<code> <style></code>
11	<code> body {</code>
12	<code> background-image: url('library_page.jpg');</code>
13	<code> background-size: cover;</code>
14	<code> background-position: center;</code>
15	<code> color: #ffffff;</code>
16	<code> font-family: Arial, sans-serif;</code>

```

17         margin: 0;
18         display: flex;
19         justify-content: center;
20         align-items: center;
21         height: 100vh;
22         overflow: auto;
23     }
24     .form-container {
25         background-color: rgba(225, 212, 187, 0.5);
26         padding: 20px;
27         border-radius: 10px;
28     }
29     input, textarea {
30         font-weight: 500;
31         font-size: 1rem;
32         color: #fff;
33         background-color: rgb(28, 28, 30);
34         box-shadow: 0 0 0.4rem rgba(0, 0, 0, 0.5),
0 0 0 0.15rem transparent;
35         border-radius: 0.4rem;
36         border: none;
37         outline: none;
38         padding: 0.4rem;
39         transition: 0.4s;
40     }
41     input:hover, textarea:hover {
42         box-shadow: 0 0 0 0.15rem rgba(135, 207,
235, 0.186);
43     }
44     input:focus, textarea:focus {
45         box-shadow: 0 0 0 0.15rem skyblue;
46     }
47     button {
48         display: inline-block;
49         border-radius: 10px;
50         border: 1px solid #03045e;
51         position: relative;
52         overflow: hidden;
53         transition: all 0.5s ease-in;
54         z-index: 1;
55     }

```

```

56         button::before, button::after {
57             content: '';
58             position: absolute;
59             top: 0;
60             width: 0;
61             height: 100%;
62             transform: skew(15deg);
63             transition: all 0.5s;
64             overflow: hidden;
65             z-index: -1;
66         }
67         button::before {
68             left: -10px;
69             background: #537188;
70         }
71         button::after {
72             right: -10px;
73             background: #537188;
74         }
75         button:hover::before, button:hover::after {
76             width: 70%;
77         }
78         button:hover span {
79             color: #e0aaff;
80             transition: 0.3s;
81         }
82         button span {
83             color: #03045e;
84             font-size: 1rem;
85             transition: all 0.3s ease-in;
86         }
87         h1 {
88             text-align: center;
89             margin-top: 20px;
90             color: #fff;
91         }
92     </style>
93 </head>
94
95 <body>
96     <?php

```



```

97     ob_start();
98     include_once('Model.php');
99     if (isset($_GET['id_member'])) {
100         editMember();
101     }
102     ?>
103     <div class="container form-container">
104         <div class="row">
105             <form action="" method="post">
106                 <h1          class="mt-2">Tambah          Data
Member</h1>
107
108                 <label for="nama_member" class="form-
label" style="color: #fff;">Nama Member</label>
109                 <input type="text" name="nama_member"
id="nama_member" class="form-control mb-3"
110                     value="<?php          if
(isset($_GET['id_member']))          echo
$result[0]['nama_member']; ?>" required>
111
112                 <label for="nomor_member" class="form-
label" style="color: #fff;">Nomor Member</label>
113                 <input type="text" name="nomor_member"
id="nomor_member" class="form-control mb-3"
114                     value="<?php          if
(isset($_GET['id_member']))          echo
$result[0]['nomor_member']; ?>" required>
115
116                 <label for="alamat" class="form-label"
style="color: #fff;">Alamat Member</label>
117                 <textarea name="alamat" id="alamat"
class="form-control mb-3" required><?php if
(isset($_GET['id_member'])) echo $result[0]['alamat'];
?></textarea>
118
119                 <label for="tgl_mendaftar" class="form-
label" style="color: #fff;">Tanggal Mendaftar</label>
120                 <input          type="datetime-local"
name="tgl_mendaftar" id="tgl_mendaftar" class="form-
control mb-3"
121

```

```

122         value="<?php                                     if
123         (isset($_GET['id_member']))    echo    date('Y-m-d\TH:i',
124         strtotime($result[0]['tgl_mendaftar'])); ?>" required>
125
126         <label                for="tgl_terakhir_bayar"
127         class="form-label"      style="color:    #fff;">Tanggal
128         Terakhir Bayar</label>
129         <input                type="date"
130         name="tgl_terakhir_bayar"    id="tgl_terakhir_bayar"
131         class="form-control mb-3"
132         value="<?php                                     if
133         (isset($_GET['id_member']))                                     echo
134         $result[0]['tgl_terakhir_bayar']; ?>" required>
135         <?php
136         ob_start();
137         if (isset($_GET['id_member'])) {
138             echo    '<button    type="submit"
139             class="btn                btn-success                mt-3"
140             name="update">Update</button>';
141         } else {
142             echo    '<button    type="submit"
143             class="btn                btn-success                mt-3"
144             name="tambah">Tambah</button>';
145         }
146     ?>
147 </form>
148 </div>
149 </div>
150 <?php
151 ob_start();
152 if (isset($_POST['update'])) {
153     $tgl_mendaftar    =    date('Y-m-d    H:i:s',
154     strtotime($_POST['tgl_mendaftar']));
155     updateMember($_GET['id_member'],
156     $_POST['nama_member'],                $_POST['nomor_member'],
157     $_POST['alamat'],                $tgl_mendaftar,
158     $_POST['tgl_terakhir_bayar']);
159 }
160
161 if (isset($_POST['tambah'])) {

```

146	<pre> \$tgl_mendaftar = date('Y-m-d H:i:s', strtotime(\$_POST['tgl_mendaftar'])); insertDataMember(\$_POST['nama_member'], \$_POST['nomor_member'], \$_POST['alamat'], 147 \$tgl_mendaftar, \$_POST['tgl_terakhir_bayar']); 148 } 149 ?> 150 </body> </html> </pre>
-----	--

L. Output Program FormMember.php

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/FormMember.php'. The page has a background image of a library with bookshelves. Overlaid on this is a semi-transparent form titled 'Tambah Data Member'. The form contains five input fields: 'Nama Member', 'Nomor Member', 'Alamat Member', 'tanggal Mendaftar' (with a date picker icon), and 'tanggal terakhir Bayar' (with a date picker icon). Below these fields is a green button labeled 'Tambah'.

Gambar 3. Screenshot Hasil Jawaban Modul 5 *FormMember.php*

M. Pembahasan FormMember.php

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

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

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

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

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

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

Pada baris [7] [8], syntax

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

integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j
IW3" crossorigin="anonymous">
```

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

Pada baris [9], syntax <title>Form Member</title> tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [10], syntax <style> merupakan tag HTML yang digunakan untuk menentukan aturan gaya (CSS) secara internal dalam dokumen HTML.

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

```
body {  
  
    background-image: url('library_page.jpg');  
  
    background-size: cover;
```

```

background-position: center;

color: #ffffff;

font-family: Arial, sans-serif;

margin: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

overflow: auto;

}

```

penggunaan baris-baris tersebut merupakan kode css yang mengatur tampilan elemen `<body>` pada halaman web dengan menggunakan gambar latar belakang "library_page.jpg" dan memastikan gambar mencakup seluruh area konten dengan `background-size: cover;`. `Background-position: center;` menempatkan gambar di tengah horizontal dan vertikal. Warna teks pada halaman web ditetapkan menjadi putih (`#ffffff`) melalui properti `color: #ffffff;`. Font teks ditentukan sebagai Arial atau font tanpa serifikasi dengan properti `font-family: Arial, sans-serif;`. Selain itu, `margin: 0;` menghapus margin yang ada secara default di sekitar elemen `<body>`. Dengan `display: flex;` digunakan untuk mengatur elemen body menggunakan model tata letak flexbox, memudahkan pengaturan penempatan elemen di dalamnya. `justify-content: center;` dan `align-items: center;` bertujuan untuk membuat konten di tengah secara horizontal dan vertikal. Tinggi elemen body ditetapkan sebagai 100% dari tinggi viewport dengan `height: 100vh;`, memastikan halaman selalu memiliki tinggi yang sama dengan tinggi layar pengguna. Terakhir, `overflow: auto;` digunakan untuk mengatur perilaku overflow jika konten melebihi ukuran viewport, sehingga scroll vertikal akan muncul jika konten tidak muat dalam satu layar.

Pada baris [24] [25][26] [27], syntax

```
.form-container {
```

```
background-color: rgba(225, 212, 187, 0.5);

padding: 20px;

border-radius: 10px;
```

penggunaan kode-kode CSS tersebut bertujuan untuk menentukan tampilan dari kontainer formulir pada halaman web. Setiap properti CSS yang digunakan memiliki peran penting dalam menentukan penampilan visual dari kontainer formulir tersebut. Penggunaan, background-color ditetapkan dengan nilai RGBA (Red, Green, Blue, Alpha) yang menghasilkan warna latar belakang untuk kontainer formulir. Dalam kasus ini, warna yang ditentukan adalah kombinasi merah (225), hijau (212), dan biru (187), dengan tingkat transparansi sebesar 0.5. Penggunaan transparansi memungkinkan latar belakang halaman untuk sedikit terlihat melalui kontainer formulir, memberikan tampilan yang lebih menarik secara visual. Selanjutnya, padding ditetapkan sebesar 20 piksel, yang menambahkan ruang pola di sekitar konten dalam kontainer formulir. Hal ini membantu menjaga jarak antara konten formulir dan tepi kontainer, memberikan tampilan yang lebih teratur dan mudah dibaca bagi pengguna. Selanjutnya, border-radius digunakan untuk mengatur sudut lengkung kontainer formulir, sehingga sudut-sudutnya menjadi lebih bulat. Dalam contoh ini, sudut-sudut kontainer formulir diberi lengkungan sebesar 10 piksel.

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

```
input, textarea {

    font-weight: 500;

    font-size: 1rem;

    color: #fff;

    background-color: rgb(28, 28, 30);

    box-shadow: 0 0 0.4rem rgba(0, 0, 0, 0.5), 0 0 0
0.15rem transparent;

    border-radius: 0.4rem;
```

```

        border: none;

        outline: none;

        padding: 0.4rem;

        transition: 0.4s;
    }

    input:hover, textarea:hover {

        box-shadow: 0 0 0 0.15rem rgba(135, 207, 235,
0.186);
    }

    input:focus, textarea:focus {

        box-shadow: 0 0 0 0.15rem skyblue;
    }

```

Penggunaan kode-kode CSS tersebut untuk mengatur tampilan dan perilaku elemen input dan textarea dalam halaman web. Pertama-tama, selektor CSS input dan textarea digunakan untuk menargetkan semua elemen input dan textarea di halaman HTML. Selanjutnya, properti-properti seperti font-weight, font-size, color, dan background-color digunakan untuk menyesuaikan tampilan teks dan latar belakang elemen. Efek bayangan diterapkan menggunakan properti box-shadow, yang memberikan tampilan tiga dimensi pada elemen dengan bayangan yang terdefinisi dengan baik. Properti-properti seperti border-radius, border, outline, dan padding digunakan untuk menyesuaikan tampilan dan ukuran elemen, sementara properti transition menambahkan efek transisi yang halus ketika ada perubahan dalam tampilan elemen. Pada saat kursor berada di atas elemen, properti hover ditetapkan untuk memberikan efek visual tambahan, seperti bayangan highlight. Ketika elemen menerima fokus, properti focus ditetapkan untuk menyoroti elemen dengan bayangan yang berbeda, menunjukkan bahwa elemen sedang aktif untuk menerima input dari pengguna.

Pada baris [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62][63]
[64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82]
[83] [84] [85] [86], syntax

```
button {  
  
    display: inline-block;  
  
    border-radius: 10px;  
  
    border: 1px solid #03045e;  
  
    position: relative;  
  
    overflow: hidden;  
  
    transition: all 0.5s ease-in;  
  
    z-index: 1;  
  
}  
  
button::before, button::after {  
  
    content: '';  
  
    position: absolute;  
  
    top: 0;  
  
    width: 0;  
  
    height: 100%;  
  
    transform: skew(15deg);  
  
    transition: all 0.5s;  
  
    overflow: hidden;  
  
    z-index: -1;  
  
}
```

```

button::before {
    left: -10px;
    background: #537188;
}

button::after {
    right: -10px;
    background: #537188;
}

button:hover::before, button:hover::after {
    width: 70%;
}

button:hover span {
    color: #e0aaff;
    transition: 0.3s;
}

button span {
    color: #03045e;
    font-size: 1rem;
    transition: all 0.3s ease-in;
}

```

penggunaan kode-kode CSS di atas adalah definisi gaya untuk tombol (button) pada halaman web. Pertama-tama, properti display diterapkan dengan nilai inline-block, memungkinkan tombol memiliki lebar dan tinggi yang didefinisikan serta dapat disusun secara horizontal. Selanjutnya, border-radius digunakan untuk memberikan sudut bulatan pada tombol dengan

efek melengkung sebesar 10 piksel, menciptakan tampilan yang lebih estetik. Border solid dengan ketebalan 1 piksel dan warna biru tua diterapkan pada tombol, memberikan garis tepi yang terlihat jelas. Properti position diatur sebagai relative untuk memungkinkan penyesuaian posisi relatif terhadap elemen lain di halaman web. Overflow tombol diatur menjadi hidden, yang berarti konten di luar batas tombol tidak akan ditampilkan. Ini berguna untuk menjaga tampilan tombol tetap rapi dan terkendali. Selanjutnya, efek transisi diterapkan pada semua perubahan properti dengan durasi 0.5 detik dan fungsi timing ease-in. Z-index ditetapkan sebagai 1 untuk mengontrol urutan tumpukan dalam lapisan tata letak, memastikan tombol berada di atas elemen lain jika diperlukan. Pseudo-elements ::before dan ::after dibuat untuk tombol dengan konten kosong dan posisi absolut. Lebar pseudo-elements diatur sebagai 0, sehingga tidak terlihat, sedangkan tinggi diatur sebagai 100% agar memiliki tinggi yang sama dengan tombol. Transformasi skew sebesar 15 derajat diterapkan pada pseudo-elements, memberikan efek miring yang menarik pada tombol. Pseudo-elements ::before dan ::after ditempatkan masing-masing di sebelah kiri dan kanan tombol. Saat tombol dihover, lebar pseudo-elements diperbesar menjadi 70% untuk memberikan efek miring yang lebih terlihat, dan warna teks tombol diubah menjadi ungu muda. Selain itu, efek transisi diterapkan pada perubahan properti dengan durasi 0.3 detik saat tombol dihover, memberikan interaksi yang responsif dan halus. Lalu, properti untuk konten teks tombol ditetapkan dengan ukuran font 1 rem dan warna teks biru tua, menjaga konsistensi dengan gaya keseluruhan halaman web.

Pada baris [87] [88] [89] [90] [91], syntax

```
h1 {  
  
    text-align: center;  
  
    margin-top: 20px;  
  
    color: #fff;  
  
}
```

penggunaan kode-kode CSS tersebut bertujuan untuk menyesuaikan gaya elemen <h1> dalam halaman web. Penggunaan, properti text-align diterapkan dengan nilai center, yang mengatur teks dalam elemen <h1> agar terpusat secara horizontal. Hal ini menciptakan

tampilan yang seragam dan estetis untuk judul halaman. Selanjutnya, margin-top diterapkan dengan nilai 20 piksel, memberikan jarak atau ruang kosong sebesar 20 piksel di bagian atas elemen <h1>. Penggunaan margin ini memungkinkan penyesuaian posisi elemen terhadap elemen lain di sekitarnya, memberikan fleksibilitas dalam tata letak halaman. Lalu, properti color diatur dengan nilai #fff, yang mewakili warna putih.

Pada baris [92], syntax </style> merupakan tag penutup dari baris 10.

Pada baris [93], syntax </head> merupakan tag penutup dari baris 3.

Pada baris [95], syntax <body> tag Body merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang ingin ditampilkan di dalam browser harus dituliskan.

Pada baris [96] [97] [98] [99] [100] [101] [102], syntax

```
<?php
    ob_start();

    include_once('Model.php');

    if (isset($_GET['id_member'])) {
        editMember();
    }

?>
```

penggunaan kode tersebut memiliki beberapa fungsi penting dalam pengelolaan halaman web. Penggunaan, fungsi ob_start() digunakan untuk memulai output buffering, sebuah teknik yang berguna dalam mengelola output pada PHP. Dengan memulai output buffering, semua output yang dihasilkan oleh skrip PHP akan dialirkan ke dalam buffer sementara sebelum dikirimkan ke klien, memungkinkan manipulasi atau pengaturan pesan kesalahan sebelum ditampilkan kepada pengguna. Selanjutnya, perintah include_once('Model.php') berfungsi untuk memasukkan file Model.php ke dalam skrip saat ini. File Model.php berisi definisi-definisi yang diperlukan seperti fungsi-fungsi untuk operasi data, koneksi database, atau definisi kelas yang dibutuhkan. Dengan memasukkan file Model.php, maka bisa

mengakses kode-kode tersebut dan menggunakan fungsionalitas yang disediakan. Lalu, terdapat sebuah kondisional if yang memeriksa apakah parameter GET dengan nama id_member telah diset. Apabila parameter ini ada, maka fungsi editMember() akan dipanggil. Hal ini digunakan ketika halaman perlu memuat data anggota yang akan diedit sebelum menampilkan formulir atau halaman untuk mengeditnya. Dengan memanggil fungsi editMember(), data anggota bisa dimuat dan halaman bisa disiapkan dengan data yang sesuai untuk diedit.

Pada baris [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132] [133] [134] [135] [136], syntax

```
<div class="container form-container">

    <div class="row">

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

            <h1 class="mt-2">Tambah Data Member</h1>

            <label for="nama_member" class="form-label"
style="color: #fff;">Nama Member</label>

            <input type="text" name="nama_member"
id="nama_member" class="form-control mb-3"

                value="<?php                                if
(isset($_GET['id_member']))    echo  $result[0]['nama_member'];
?>" required>

            <label for="nomor_member" class="form-label"
style="color: #fff;">Nomor Member</label>

            <input type="text" name="nomor_member"
id="nomor_member" class="form-control mb-3"
```

```

                                value="<?php                                if
(isset($_GET['id_member']))    echo  $result[0]['nomor_member'];
?>" required>

```

```

                                <label        for="alamat"        class="form-label"
style="color: #fff;">Alamat Member</label>

```

```

                                <textarea        name="alamat"        id="alamat"
class="form-control        mb-3"        required><?php        if
(isset($_GET['id_member']))        echo        $result[0]['alamat'];
?></textarea>

```

```

                                <label for="tgl_mendaftar" class="form-label"
style="color: #fff;">Tanggal Mendaftar</label>

```

```

                                <input                                type="datetime-local"
name="tgl_mendaftar"    id="tgl_mendaftar"    class="form-control
mb-3"

```

```

                                value="<?php                                if
(isset($_GET['id_member']))        echo        date('Y-m-d\TH:i',
strtotime($result[0]['tgl_mendaftar'])); ?>" required>

```

```

                                <label for="tgl_terakhir_bayar" class="form-
label" style="color: #fff;">Tanggal Terakhir Bayar</label>

```

```

                                <input type="date" name="tgl_terakhir_bayar"
id="tgl_terakhir_bayar" class="form-control mb-3"

```

```

                                value="<?php                                if
(isset($_GET['id_member']))                                echo
$result[0]['tgl_terakhir_bayar']; ?>" required>

```

```

        <?php
            ob_start();

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

                echo '<button type="submit" class="btn
btn-success mt-3" name="update">Update</button>';

            } else {

                echo '<button type="submit" class="btn
btn-success mt-3" name="tambah">Tambah</button>';

            }

        ?>

    </form>

</div>

</div>

```

penggunaan kode-kode di atas adalah formulir HTML yang berfungsi untuk menambah atau mengedit data member. Secara rinci, elemen `<div class="container form-container">` menggunakan kelas CSS Bootstrap untuk membuat kontainer formulir dengan latar belakang semi-transparan. Selanjutnya, `<form action="" method="post">` digunakan untuk membuat formulir dengan metode POST, dengan aksi formulir yang ditetapkan kosong (""), yang berarti data formulir akan dikirimkan ke halaman yang sama. Judul formulir ditampilkan dengan menggunakan elemen `<h1 class="mt-2">Tambah Data Member</h1>`. Setiap elemen `<input>` dan `<textarea>` memiliki atribut `name` dan `id` untuk mengidentifikasi elemen tersebut dalam pengiriman formulir dan manipulasi JavaScript. Nilai-nilai atribut `value` diatur untuk menetapkan nilai default ke dalam input saat ini sesuai dengan data yang diedit, jika dalam mode edit. Kondisional PHP `if (isset($_GET['id_member']))` digunakan untuk menentukan apakah formulir dalam mode "tambah" atau "edit". Jika `id_member` tersedia di URL (dalam mode edit), maka tombol akan menampilkan "Update", jika tidak, maka tombol akan menampilkan "Tambah".

Pada baris [137] [138] [139] [140] [141] [142] [143] [144] [145] [146] [147] [148], syntax

```
<?php

    ob_start();

    if (isset($_POST['update'])) {

        $tgl_mendaftar      =      date('Y-m-d      H:i:s',
strtotime($_POST['tgl_mendaftar']));

        updateMember($_GET['id_member'],
$_POST['nama_member'],      $_POST['nomor_member'],
$_POST['alamat'],      $tgl_mendaftar,
$_POST['tgl_terakhir_bayar']);

    }

    if (isset($_POST['tambah'])) {

        $tgl_mendaftar      =      date('Y-m-d      H:i:s',
strtotime($_POST['tgl_mendaftar']));

        insertDataMember($_POST['nama_member'],
$_POST['nomor_member'],    $_POST['alamat'],    $tgl_mendaftar,
$_POST['tgl_terakhir_bayar']);

    }

?>
```

penggunaan kode PHP ini bertanggung jawab untuk mengelola aksi formulir yang dikirimkan melalui metode POST. Pertama, <?php menandakan awal dari blok kode PHP. Kemudian, ob_start(); digunakan untuk memulai output buffering, yang memungkinkan output PHP disimpan ke dalam buffer sementara sebelum dikirimkan ke browser. Selanjutnya, if (isset(\$_POST['update'])) { digunakan untuk memeriksa apakah tombol "Update" telah diklik di formulir. Jika ya, maka kode dalam blok if akan dieksekusi. Di dalamnya, tanggal mendaftar dikonversi ke format yang dapat disimpan di database, dan

fungsi `updateMember()` dipanggil untuk memperbarui data member yang ada. Kemudian, if (`isset($_POST['tambah'])`) { memeriksa apakah tombol "Tambah" telah diklik. Jika ya, maka tanggal mendaftar dikonversi lagi dan fungsi `insertDataMember()` dipanggil untuk menambahkan data member baru ke database. Lalu, `?>` menandakan akhir dari blok kode PHP.

pada baris [149], syntax `</body>` merupakan tag penutup dari baris 95.

Pada baris [150], syntax `</html>` merupakan tag penutup dari baris 2.

N. Source Code Buku.php

Table 6. Source Code Modul 5 Buku.php

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	<code> <meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>
6	<code> <meta name="viewport" content="width=device-width,</code>
	<code>initial-scale=1.0"></code>
7	<code> <link</code>
	<code>href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dis</code>
	<code>t/css/bootstrap.min.css" rel="stylesheet"</code>
	<code>integrity="sha384-</code>
	<code>1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy</code>
	<code>l2QvZ6jIW3" crossorigin="anonymous"></code>
8	<code> <title>Data Buku</title></code>
9	<code> <style></code>
10	<code> body {</code>
11	<code> background-image: url('library_page.jpg');</code>
12	<code> background-size: cover;</code>
13	<code> background-position: center;</code>
14	<code> color: #ffffff;</code>
15	<code> font-family: Arial, sans-serif;</code>
16	<code> margin: 0;</code>
17	<code> display: flex;</code>
18	<code> flex-direction: column;</code>
19	<code> align-items: center;</code>
20	<code> justify-content: center;</code>

```

21         overflow: auto;
22     }
23     table, tr, td {
24         border-collapse: collapse;
25         border-radius: 21px;
26         padding: 20px 40px;
27         font-size: 18px;
28     }
29     table {
30         width: max-content;
31         border-radius: 21px;
32         background: rgba(255, 255, 255, 0.8);
33         color: black;
34     }
35     td {
36         width: 100px;
37         height: 10px;
38         text-align: center;
39     }
40     .buttons-container {
41         display: flex;
42         justify-content: flex-end;
43         margin-bottom: 20px;
44     }
45     .buttons-container a:first-child {
46         margin-right: 10px;
47     }
48     button {
49         border-radius: 10px;
50         border: 1px solid #03045e;
51         position: relative;
52         overflow: hidden;
53         transition: all 0.5s ease-in;
54         z-index: 1;
55     }
56     button::before, button::after {
57         content: '';
58         position: absolute;
59         top: 0;
60         width: 0;
61         height: 100%;

```

```

62         transform: skew(15deg);
63         transition: all 0.5s;
64         overflow: hidden;
65         z-index: -1;
66     }
67     button::before {
68         left: -10px;
69         background: #537188;
70     }
71     button::after {
72         right: -10px;
73         background: #537188;
74     }
75     button:hover::before, button:hover::after {
76         width: 70%;
77     }
78     button:hover span {
79         color: #e0aaff;
80         transition: 0.3s;
81     }
82     button span {
83         color: #03045e;
84         font-size: 18px;
85         transition: all 0.3s ease-in;
86     }
87     h1 {
88         font-size: 48px;
89         margin: 40px auto 20px;
90         color: #FFFAF0;
91         text-transform: uppercase;
92         letter-spacing: 2px;
93         font-weight: bold;
94         text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
95         animation: fadeInDown 1s ease-in-out,
scaleIn 1s ease-in-out;
96         border-bottom: 2px solid #FFFAF0;
97         padding-bottom: 10px;
98         background-color: rgba(0, 0, 0, 0.5);
99         width: 80%;
100        text-align: center;
101    }

```

```

102     </style>
103 </head>
104
105 <body class="p-3" style="background-color: #E1D4BB;">
106 <?php
107 include_once("Koneksi.php");
108
109 function deleteBuku($id_buku)
110 {
111     global $conn;
112     $stmt = $conn->prepare("DELETE FROM peminjaman WHERE
113 id_buku = ?");
114     $stmt->execute([$id_buku]);
115     $stmt = $conn->prepare("DELETE FROM buku WHERE
116 id_buku = ?");
117     $stmt->execute([$id_buku]);
118
119     header("Location: Buku.php");
120     exit();
121 }
122
123 function readData($nama_tabel)
124 {
125     global $conn;
126     $stmt = $conn->prepare("SELECT * FROM $nama_tabel");
127     $stmt->execute();
128     $result = $stmt->fetchAll();
129
130     if (!empty($result)) {
131         if ($nama_tabel == "buku") {
132             foreach ($result as $hasil) {
133                 echo "<tr>";
134                 echo "    <td class='text-center'>" .
135 $hasil['id_buku'] . "</td>";
136                 echo "    <td>" . $hasil['judul_buku'] .
137 "</td>";
138                 echo "    <td>" . $hasil['penulis'] .
139 "</td>";
140                 echo "    <td>" . $hasil['penerbit'] .
141 "</td>";
142             }
143         }
144     }
145 }

```

```

137         echo "<td>" . $hasil["tahun_terbit"] .
138     "</td>";
139         echo "<td>";
140         echo "<a class='btn btn-primary'
href='FormBuku.php?id_buku=" . $hasil['id_buku'] .
139     "'>Edit</a>";
140         echo " ";
141         echo "<a class='btn btn-danger'
href='Buku.php?id_buku=" . $hasil['id_buku'] . "
onclick=\"return confirm('Yakin Ingin
141     Dihapus?')\">Hapus</a>";
142         echo "</td>";
143         echo "</tr>";
144     }
145 }
146 }
147 }
148
149 if (isset($_GET['id_buku'])) {
150     deleteBuku($_GET['id_buku']);
151 }
152 ?>
153
154 <h2>
155     <center><h1>Data Buku</h1></center>
156 </h2>
157 <div class="buttons-container">
158     <a href="index.php"><button class="btn btn-dark
mb-4">Kembali</button></a>
159     <a href="FormBuku.php"><button class="btn btn-
warning mb-1">Tambah Data Buku</button></a>
160 </div>
161 <table class="table">
162     <thead class="table-dark">
163         <tr>
164             <th class="text-center">ID Buku</th>
165             <th class="text-center">Judul Buku</th>
166             <th class="text-center">Penulis</th>
167             <th class="text-center">Penerbit</th>
168             <th class="text-center">Tahun
Terbit</th>

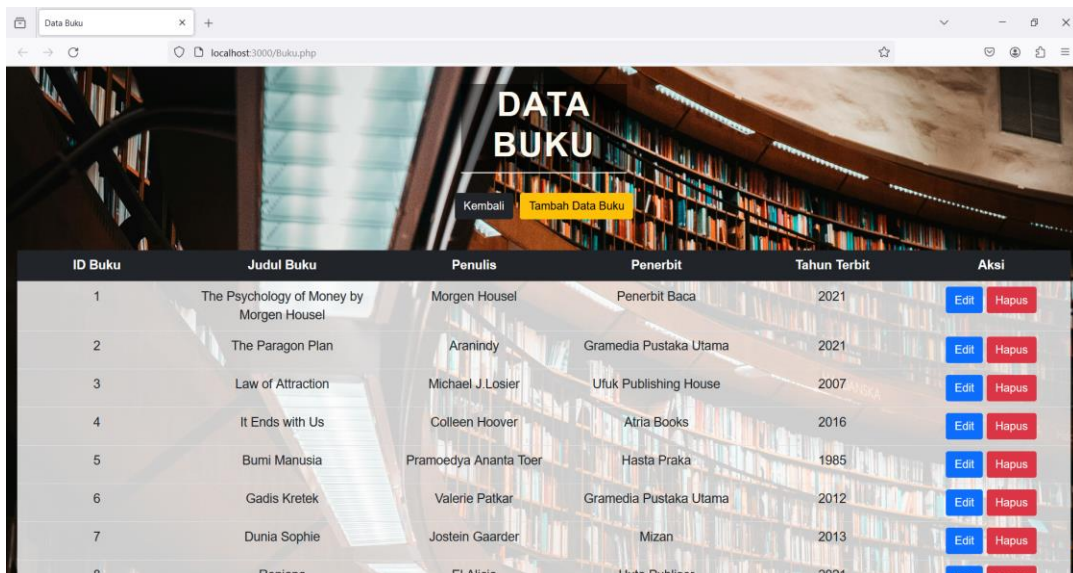
```

```

169         <th class="text-center">Aksi</th>
170     </tr>
171 </thead>
172 <tbody>
173     <?php readData("buku"); ?>
174 </tbody>
175 </table>
176 </body>
</html>

```

O. Output Program Buku.php



ID Buku	Judul Buku	Penulis	Penerbit	Tahun Terbit	Aksi
1	The Psychology of Money by Morgen Housel	Morgen Housel	Penerbit Baca	2021	Edit Hapus
2	The Paragon Plan	Aranindy	Gramedia Pustaka Utama	2021	Edit Hapus
3	Law of Attraction	Michael J. Losier	Ufuk Publishing House	2007	Edit Hapus
4	It Ends with Us	Colleen Hoover	Alria Books	2016	Edit Hapus
5	Bumi Manusia	Pramoedya Ananta Toer	Hasta Praka	1985	Edit Hapus
6	Gadis Kretek	Valerie Patkar	Gramedia Pustaka Utama	2012	Edit Hapus
7	Dunia Sophie	Jostein Gaarder	Mizan	2013	Edit Hapus
8	Renjana	El Alinia	Hita Publisher	2021	Edit Hapus

Gambar 4. Screenshot Hasil Jawaban Modul 5 Buku.php

P. Pembahasan Buku.php

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

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

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

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

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

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

Pada baris [7], syntax

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

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j
IW3" crossorigin="anonymous">
```

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

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

Pada baris [9] – [102], syntax

```
<style>
```

```
body {  
  
    background-image: url('library_page.jpg');  
  
    background-size: cover;  
  
    background-position: center;
```



```

    color: #ffffff;

    font-family: Arial, sans-serif;

    margin: 0;

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    overflow: auto;
}

table, tr, td {

    border-collapse: collapse;

    border-radius: 21px;

    padding: 20px 40px;

    font-size: 18px;
}

table {

    width: max-content;

    border-radius: 21px;

    background: rgba(255, 255, 255, 0.8);

    color: black;
}

td {

    width: 100px;

```

```

        height: 10px;

        text-align: center;
    }

.buttons-container {

    display: flex;

    justify-content: flex-end;

    margin-bottom: 20px;
}

.buttons-container a:first-child {

    margin-right: 10px;
}

button {

    border-radius: 10px;

    border: 1px solid #03045e;

    position: relative;

    overflow: hidden;

    transition: all 0.5s ease-in;

    z-index: 1;
}

button::before, button::after {

    content: '';

    position: absolute;

    top: 0;

```

```

width: 0;

height: 100%;

transform: skew(15deg);

        transition: all 0.5s;

overflow: hidden;

z-index: -1;

}

button::before {

    left: -10px;

    background: #537188;

}

button::after {

    right: -10px;

    background: #537188;

}

button:hover::before, button:hover::after {

    width: 70%;

}

button:hover span {

    color: #e0aaff;

    transition: 0.3s;

}

button span {

```

```

        color: #03045e;

        font-size: 18px;

        transition: all 0.3s ease-in;
    }

    h1 {

        font-size: 48px;

        margin: 40px auto 20px;

        color: #FFFAF0;

        text-transform: uppercase;

        letter-spacing: 2px;

        font-weight: bold;

        text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);

        animation: fadeInDown 1s ease-in-out, scaleIn 1s
ease-in-out;

        border-bottom: 2px solid #FFFAF0;

        padding-bottom: 10px;

        background-color: rgba(0, 0, 0, 0.5);

        width: 80%;

        text-align: center;

    }

</style>

```

penggunaan kode-kode css terseut untuk mengatur beberapa gaya pada web. Pengaturan css pada `body { ... }`: digunakan untuk mendefinisikan gaya untuk elemen `<body>`, termasuk pengaturan gambar latar belakang, warna teks, dan tata letak. Latar belakang menggunakan

gambar dari 'library_page.jpg', dengan ukuran yang menyesuaikan ukuran halaman dan posisi yang diatur ke tengah. Warna teks diatur menjadi putih, menggunakan font Arial atau sans-serif, dan tata letak halaman ditata agar berada di tengah dengan menggunakan flexbox. Lalu table, tr, td { ... }: Mendefinisikan gaya untuk elemen tabel dan selnya, termasuk pengaturan tata letak, padding, dan ukuran teks. Padding diatur menjadi 20px di atas dan bawah serta 40px di kiri dan kanan untuk setiap sel. Selanjutnya, gaya untuk elemen-elemen tabel seperti <table>, <tr>, dan <td> memastikan tampilan tabel yang rapi dengan padding yang merata di setiap selnya. Tambahan gaya untuk elemen tabel mengatur lebar maksimum, warna latar belakang, dan penataan teks. Selain itu, wadah untuk tombol (buttons-container) didefinisikan dengan penataan flex-end untuk meletakkan tombol di bagian kanan wadah dengan margin bawah yang sesuai. Efek visual tambahan pada tombol diperkenalkan menggunakan pseudo-element (::before dan ::after), dengan latar belakang berwarna tertentu yang mengikuti interaksi mouse saat tombol dihover. Gaya untuk elemen <h1> menetapkan ukuran font besar, warna, efek bayangan, dan efek animasi agar judul halaman menonjol dan menarik perhatian pengguna.

Pada baris [103], syntax <head> untuk menutup tag pada baris 3.

Pada baris [105] – [151], syntax <body class="p-3" style="background-color: #E1D4BB;">

```
<?php
```

```
include_once("Koneksi.php");
```

```
function deleteBuku($id_buku)
```

```
{
```

```
    global $conn;
```

```
    $stmt = $conn->prepare("DELETE FROM peminjaman WHERE  
id_buku = ?");
```

```
    $stmt->execute([$id_buku]);
```

```

        $stmt = $conn->prepare("DELETE FROM buku WHERE id_buku =
?");

        $stmt->execute([$id_buku]);

        header("Location: Buku.php");

        exit();

    }

```

```

function readData($nama_tabel)
{

    global $conn;

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

    $stmt->execute();

    $result = $stmt->fetchAll();

    if (!empty($result)) {

        if ($nama_tabel == "buku") {

            foreach ($result as $hasil) {

                echo "<tr>";

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

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

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

```

```

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

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

        echo "<td>";

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

        echo " ";

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

        echo "</td>";

        echo "</tr>";

    }

}

}

}

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

    deleteBuku($_GET['id_buku']);

}

?>

```

penggunaan dari `<body class="p-3" style="background-color: #E1D4BB;">`: Membuat elemen `<body>` dengan kelas `p-3` yang berfungsi untuk memberikan padding sebesar 3 unit ke semua sisi elemen, dan properti gaya inline untuk mengatur warna latar belakang menjadi

#E1D4BB. Selanjutnya `<?php include_once("Koneksi.php"); ?>`: menggunakan pernyataan PHP `include_once` untuk menyisipkan file `Koneksi.php` ke dalam halaman. Ini memungkinkan kode dalam file `Koneksi.php` dijalankan di halaman saat ini. Untuk penggunaan function `deleteBuku($id_buku) { ... }`: sebagai pendefinisian sebuah fungsi PHP bernama `deleteBuku` yang menerima parameter `$id_buku`. Fungsi ini bertujuan untuk menghapus data buku dari database berdasarkan `$id_buku` yang diberikan. Penggunaan fungsi `function readData($nama_tabel) { ... }`: mendefinisikan sebuah fungsi PHP bernama `readData` yang menerima parameter `$nama_tabel`. Fungsi ini bertujuan untuk membaca data dari tabel yang disebutkan dalam parameter dan menampilkannya dalam bentuk tabel HTML. `global $conn;`: digunakan untuk mendeklarasikan variabel global `$conn` yang merepresentasikan koneksi ke database. Variabel ini akan digunakan di dalam fungsi-fungsi yang ada di dalam file ini. Penggunaan `$stmt = $conn->prepare("DELETE FROM peminjaman WHERE id_buku = ?");`: untuk mempersiapkan pernyataan SQL untuk menghapus data buku dari tabel peminjaman berdasarkan `$id_buku`. `$stmt->execute([$id_buku]);`: menjalankan pernyataan SQL untuk menghapus data buku berdasarkan `$id_buku` yang diberikan. Lalu `header("Location: Buku.php");`: mengarahkan pengguna ke halaman `Buku.php` setelah proses penghapusan data buku selesai. `exit();`: bertujuan untuk menghentikan eksekusi skrip PHP secara paksa setelah mengarahkan pengguna ke halaman `Buku.php`. Dan `if (isset($_GET['id_buku'])) { ... }`: memeriksa apakah parameter `id_buku` telah diset dalam URL. Apabila ya, maka akan memanggil fungsi `deleteBuku` untuk menghapus buku berdasarkan id yang diberikan.

Pada baris [153] – [174], syntax

```
<h2>

    <center><h1>Data Buku</h1></center>

</h2>

<div class="buttons-container">

    <a href="index.php"><button class="btn btn-dark mb-4">Kembali</button></a>
```



```

        <a href="FormBuku.php"><button class="btn btn-warning
mb-1">Tambah Data Buku</button></a>

</div>

<table class="table">

    <thead class="table-dark">

        <tr>

            <th class="text-center">ID Buku</th>

            <th class="text-center">Judul Buku</th>

            <th class="text-center">Penulis</th>

            <th class="text-center">Penerbit</th>

            <th class="text-center">Tahun Terbit</th>

            <th class="text-center">Aksi</th>

        </tr>

    </thead>

    <tbody>

        <?php readData("buku"); ?>

    </tbody>

</table>

```

penggunaan baris-baris tersebut merupakan kombinasi dari kode HTML dan PHP. Penggunaan, tag <h2> digunakan untuk membuat heading level 2. Selanjutnya, div dengan kelas CSS buttons-container berfungsi sebagai wadah untuk tombol-tombol yang akan ditempatkan di atas atau bawah halaman web. Kemudian, tautan menggunakan tag <a> mengarahkan pengguna ke halaman tertentu, dengan masing-masing tombol yang terkait memiliki gaya yang ditentukan oleh kelas CSS dari framework Bootstrap. Tag <table> digunakan untuk membuat tabel, dengan kelas CSS table untuk menambahkan gaya default

dari Bootstrap. Bagian header tabel ditandai dengan tag <thead> yang diberi kelas table-dark untuk memberikan latar belakang yang gelap. Setiap sel heading dalam tabel ditandai dengan tag <th>, di mana kelas CSS text-center digunakan untuk menengahkan teks di dalam sel. Lalu, pengekseskusion fungsi PHP readData("buku") bertujuan untuk membaca dan menampilkan data dari tabel buku dalam bentuk baris-baris pada tabel HTML.

Pada baris [175], syntax </body> merupakan tag penutup dari batis 105.

Pada baris [176], syntax </html> merupakan tag penutup dari baris 2.

Q. Source Code FormBuku.php

Table 7. Source Code Modul 5 FormBuku.php

1	<?php
2	ob_start();
3	include 'Koneksi.php';
4	include 'Model.php';
5	?>
6	
7	<!DOCTYPE html>
8	<html lang="en">
9	<head>
10	<meta charset="UTF-8">
	<meta http-equiv="X-UA-Compatible"
11	content="IE=edge">
	<meta name="viewport" content="width=device-width,
12	initial-scale=1.0">
	<link
13	href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dis
	t/css/bootstrap.min.css" rel="stylesheet"
	integrity="sha384-
14	1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy
	12QvZ6jIW3" crossorigin="anonymous">
	<title>Formulir Buku</title>
15	<style>
16	body {
17	background-image: url('library_page.jpg');
18	background-size: cover;
19	background-position: center;
20	color: #ffffff;

```

21         font-family: Arial, sans-serif;
22         margin: 0;
23         display: flex;
24         justify-content: center;
25         align-items: center;
26         height: 100vh;
27         overflow: auto;
28     }
29     .form-container {
30         background-color: rgba(225, 212, 187, 0.5);
31         padding: 20px;
32         border-radius: 10px;
33     }
34     input {
35         font-weight: 500;
36         font-size: .8vw;
37         color: #fff;
38         background-color: rgb(28, 28, 30);
39         box-shadow: 0 0 .4vw rgba(0, 0, 0, 0.5), 0
40 0 0 .15vw transparent;
41         border-radius: 0.4vw;
42         border: none;
43         outline: none;
44         padding: 0.4vw;
45         transition: .4s;
46     }
47     input:hover {
48         box-shadow: 0 0 0 .15vw rgba(135, 207, 235,
49 0.186);
50     }
51     input:focus {
52         box-shadow: 0 0 0 .15vw skyblue;
53     }
54     button {
55         display: inline-block;
56         border-radius: 10px;
57         border: 1px solid #03045e;
58         position: relative;
59         overflow: hidden;
60         transition: all 0.5s ease-in;
61         z-index: 1;

```

```

60     }
61     button::before,
62     button::after {
63         content: '';
64         position: absolute;
65         top: 0;
66         width: 0;
67         height: 100%;
68         transform: skew(15deg);
69         transition: all 0.5s;
70         overflow: hidden;
71         z-index: -1;
72     }
73     button::before {
74         left: -10px;
75         background: #537188;
76     }
77     button::after {
78         right: -10px;
79         background: #537188;
80     }
81     button:hover::before,
82     button:hover::after {
83         width: 70%;
84     }
85     button:hover span {
86         color: #e0aaff;
87         transition: 0.3s;
88     }
89     button span {
90         color: #03045e;
91         font-size: 18px;
92         transition: all 0.3s ease-in;
93     }
94     </style>
95 </head>
96
97 <body class="p-3">
98     <?php
99     ob_start();
100    include_once('Model.php');

```

```

101     if (isset($_GET['id_buku'])) {
102         editBuku();
103     }
104     ?>
105
106     <div class="container form-container">
107         <div class="row">
108             <form action="" method="post">
109                 <h1 style="text-align: center;"
110 class="mt-2" style="color: #fff;">Tambah Data Buku</h1>
111
112                 <label for="judul_buku" class="form-
113 label" style="color: #fff;">Judul Buku</label>
114                 <input type="text" name="judul_buku"
115 id="judul_buku" class="form-control mb-3" value="<?php
116 if (isset($_GET['id_buku']))
117     echo $result[0]["judul_buku"] ?>"
118 required>
119
120                 <label for="penulis" class="form-
121 label" style="color: #fff;">Penulis</label>
122                 <input type="text" name="penulis"
123 id="penulis" class="form-control mb-3" value="<?php if
124 (isset($_GET['id_buku']))
125     echo $result[0]["penulis"] ?>"
126 required>
127
128                 <label for="penerbit" class="form-
129 label" style="color: #fff;">Penerbit</label>
130                 <input type="text" name="penerbit"
131 id="penerbit" class="form-control mb-3" value="<?php if
132 (isset($_GET['id_buku']))
133     echo $result[0]["penerbit"] ?>"
134 required>
135
136                 <label for="tahun_terbit"
137 class="form-label" style="color: #fff;">Tahun
138 Terbit</label>
139                 <input type="text"
140 name="tahun_terbit" id="tahun_terbit" class="form-
141 control mb-3" value="<?php if (isset($_GET['id_buku']))

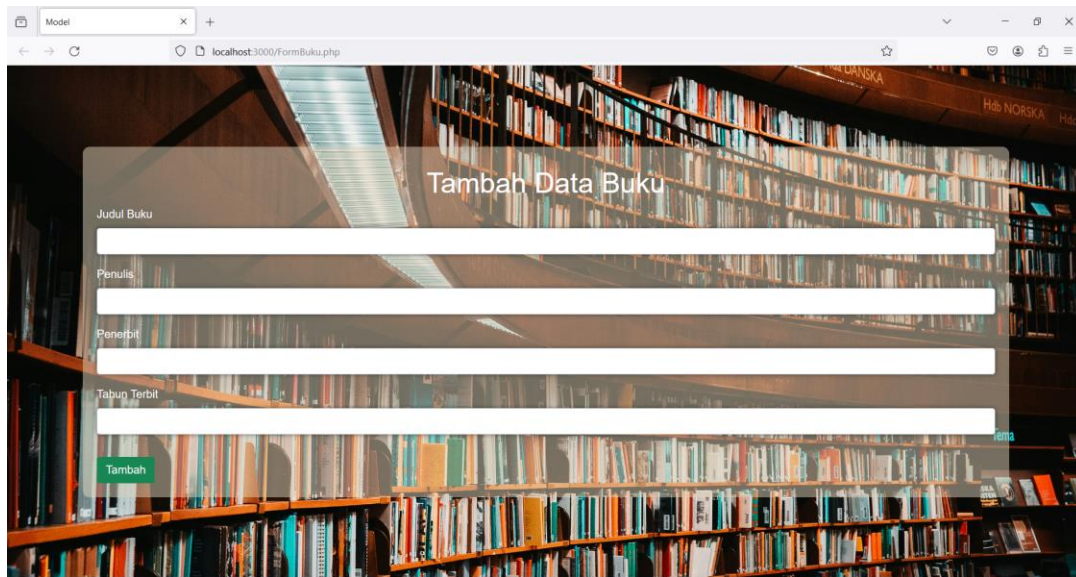
```

```

125         echo $result[0]["tahun_terbit"] ?>"
required>
        <?php
126         ob_start();
127         if (isset($_GET['id_buku'])) {
128             echo "<button      type=\"submit\"
129 class=\"btn btn-success mt-3\" name=\"update\"> Update
</button>";
130         } else {
131             echo "<button      type=\"submit\"
132 class=\"btn btn-success mt-3\" name=\"tambah\"> Tambah
</button>";
133         }
134         ?>
135     </div>
136 </div>
137 </form>
138 <?php
139 ob_start();
140 if (isset($_POST['update'])) {
141     updateBuku($_GET['id_buku'],
$_POST['judul_buku'],          $_POST['penulis'],
$_POST['penerbit'], $_POST['tahun_terbit']);
142 }
143 if (isset($_POST['tambah'])) {
144     insertDataBuku($_POST['id_buku'],
145 $_POST['judul_buku'],          $_POST['penulis'],
$_POST['penerbit'], $_POST['tahun_terbit']);
146 }
147 ?>
148 </body>
149 </html>

```

R. Output Program FormBuku.php



Gambar 5. Screenshot Hasil Jawaban Modul 5 FormBuku.php

S. Pembahasan FormBuku.php

Pada baris [1] [2] [3] [4] [5], syntax

```
<?php
ob_start();

include 'Koneksi.php';

include 'Model.php';

?>
```

Penggunaan pada baris-baris tersebut adalah untuk mengelola output dan menyertakan file eksternal yang diperlukan untuk koneksi dan operasi database. Penggunaan fungsi `ob_start()` digunakan untuk memulai output buffering, yang memungkinkan skrip PHP menahan pengiriman output ke browser hingga `ob_end_flush()` atau `ob_flush()` dipanggil. Selanjutnya, pernyataan `include 'Koneksi.php';` menyertakan file `Koneksi.php`, yang berisi kode untuk menghubungkan aplikasi ke database. Dengan menyertakan file ini, skrip PHP mendapatkan akses ke variabel dan fungsi yang mendefinisikan koneksi database, sehingga memungkinkan operasi database dilakukan di seluruh skrip. Selain itu, pernyataan `include 'Model.php';` menyertakan file `Model.php`, yang berisi definisi fungsi-fungsi untuk

berinteraksi dengan database, seperti fungsi untuk mengambil data, menyimpan data, memperbarui data, dan menghapus data. Dengan menyertakan file ini, skrip PHP dapat menggunakan fungsi-fungsi tersebut untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada data.

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

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

Pada baris [9], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa ditambahkan pada tag ini.

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

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

Pada baris [12], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web

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

Pada baris [13] [14], syntax

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

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j
IW3" crossorigin="anonymous">
```

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

Pada baris [15], syntax `<title>Formulir Buku</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [16] – [95], syntax

`<style>`

```
body {  
  
    background-image: url('library_page.jpg');  
  
    background-size: cover;  
  
    background-position: center;  
  
    color: #ffffff;  
  
    font-family: Arial, sans-serif;  
  
    margin: 0;  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: center;  
  
    height: 100vh;  
  
    overflow: auto;  
  
}  
  
.form-container {  
  
    background-color: rgba(225, 212, 187, 0.5);  
  
    padding: 20px;  
  
    border-radius: 10px;  
  
}
```

```

input {
    font-weight: 500;

    font-size: .8vw;

    color: #fff;

    background-color: rgb(28, 28, 30);

    box-shadow: 0 0 .4vw rgba(0, 0, 0, 0.5), 0 0 0
.15vw transparent;

    border-radius: 0.4vw;

    border: none;

    outline: none;

    padding: 0.4vw;

    transition: .4s;
}

input:hover {
    box-shadow: 0 0 0 .15vw rgba(135, 207, 235, 0.186);
}

input:focus {
    box-shadow: 0 0 0 .15vw skyblue;
}

button {

    display: inline-block;

    border-radius: 10px;

    border: 1px solid #03045e;

```

```

        position: relative;

        overflow: hidden;

        transition: all 0.5s ease-in;

        z-index: 1;
    }

    button::before,
    button::after {

        content: '';

        position: absolute;

        top: 0;

        width: 0;

        height: 100%;

        transform: skew(15deg);

        transition: all 0.5s;

        overflow: hidden;

        z-index: -1;
    }

    button::before {

        left: -10px;

        background: #537188;
    }

    button::after {

        right: -10px;
    }

```

```

        background: #537188;
    }

    button:hover::before,
    button:hover::after {

        width: 70%;

    }

    button:hover span {

        color: #e0aaff;

        transition: 0.3s;

    }

    button span {

        color: #03045e;

        font-size: 18px;

        transition: all 0.3s ease-in;

    }

</style>

```

penggunaan baris-baris tersebut untuk mengatur tampilan halaman web. Selector body menetapkan gaya untuk elemen body, termasuk gambar latar belakang dari file 'library_page.jpg' yang disetel agar mencakup seluruh area tanpa mengubah proporsi dan diposisikan di tengah. Warna teks diatur menjadi putih, menggunakan font Arial atau sans-serif, dengan margin dihapus dan tata letak halaman ditata di tengah menggunakan flexbox. Tinggi elemen body diatur sebesar 100% dari tinggi viewport dan konten yang meluap diizinkan untuk digulir. Selector .form-container menetapkan warna latar belakang dengan transparansi 50%, menambahkan ruang dalam sebesar 20px di sekitar konten, dan membulatkan sudut elemen dengan radius 10px. Selector input menetapkan berbagai gaya

untuk elemen input, termasuk ketebalan font, ukuran font relatif terhadap lebar viewport, warna teks putih, warna latar belakang gelap, bayangan di sekitar elemen, sudut yang membulat, dan berbagai properti lain untuk memastikan tampilan yang konsisten dan responsif. Selain itu, gaya tambahan diterapkan saat elemen input di-hover dan difokuskan, dengan menambahkan bayangan berwarna tertentu. Selector button dan pseudo-elementnya menetapkan berbagai gaya untuk tombol, termasuk membuat elemen menjadi inline-block, membulatkan sudut, menetapkan border, dan menambahkan efek transisi. Pseudo-element button::before dan button::after digunakan untuk menambahkan elemen tambahan sebelum dan sesudah konten tombol, dengan berbagai properti posisi dan transformasi untuk menciptakan efek visual yang menarik saat tombol di-hover. Selector button:hover::before dan button:hover::after mengubah lebar elemen pseudo menjadi 70% saat tombol di-hover, sementara button:hover span mengubah warna teks menjadi #e0aaff dan menetapkan durasi transisi selama 0.3 detik. Selector button span menetapkan warna teks span menjadi #03045e, ukuran font 18px, dan durasi serta jenis transisi untuk memastikan perubahan yang halus dan konsisten.

Pada baris [96], syntax `</head>` merupakan tag penutup dari baris 9.

Pada baris [98], syntax `<body class="p-3">` merupakan framework css.

Pada baris [99] [100] [101] [102] [103] [104] [105], syntax

```
<?php
    ob_start();

    include_once('Model.php');

    if (isset($_GET['id_buku'])) {
        editBuku();
    }

?>
```

Penggunaan baris-baris tersebut digunakan dalam sebuah skrip PHP untuk mengelola data buku dalam suatu aplikasi web. Penggunaan baris pertama yaitu, `<?php ob_start(); ?>`,

memulai output buffering. Ketika output buffering diaktifkan, alih-alih mengirim output (seperti HTML atau teks lainnya) langsung ke browser, PHP akan menahan output tersebut dalam buffer internal. Hal ini memungkinkan untuk melakukan perubahan atau pemeriksaan pada output sebelum dikirim ke browser, dan berguna untuk mengelola header HTTP serta mencegah output yang tidak diinginkan sebelum header dikirim. Selanjutnya, `include_once('Model.php');` menyertakan file `Model.php` satu kali ke dalam skrip. File `Model.php` biasanya berisi fungsi-fungsi yang digunakan untuk berinteraksi dengan database atau model data lainnya. Dengan menggunakan `include_once`, akan memastikan bahwa file `Model.php` hanya dimuat satu kali, menghindari potensi masalah seperti deklarasi ulang fungsi atau variabel yang sama. Kemudian, blok `if (isset($_GET['id_buku'])) { ... }` memeriksa apakah parameter `id_buku` ada dalam query string URL. Query string adalah bagian dari URL yang dimulai dengan tanda `?` dan diikuti oleh pasangan kunci-nilai. Ini memastikan bahwa `id_buku` telah disediakan oleh pengguna atau aplikasi sebelum mencoba melakukan operasi yang bergantung pada parameter ini. Jika `id_buku` tersedia, fungsi `editBuku()` dipanggil. Fungsi ini kemungkinan besar diambil dari `Model.php` dan berisi logika untuk mengedit data buku berdasarkan `id_buku` yang diterima dari query string. Ini memungkinkan pengeditan data buku yang ada di database atau sistem penyimpanan lainnya berdasarkan ID buku yang disediakan.

Pada baris [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132] [133] [134] [135] [136] [137], syntax

```
<div class="container form-container">

    <div class="row">

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

            <h1 style="text-align: center;" class="mt-2"
style="color: #fff;">Tambah Data Buku</h1>

            <label for="judul_buku" class="form-label"
style="color: #fff;">Judul Buku</label>
```

```

                <input      type="text"      name="judul_buku"
id="judul_buku"  class="form-control mb-3"  value="<?php if
(isset($_GET['id_buku']))

                echo      $result[0]["judul_buku"]      ?>"
required>

```

```

                <label  for="penulis"  class="form-label"
style="color: #fff;">Penulis</label>

```

```

                <input      type="text"      name="penulis"
id="penulis"    class="form-control  mb-3"  value="<?php  if
(isset($_GET['id_buku']))

```

```

                echo $result[0]["penulis"] ?>" required>

```

```

                <label  for="penerbit"  class="form-label"
style="color: #fff;">Penerbit</label>

```

```

                <input      type="text"      name="penerbit"
id="penerbit"   class="form-control  mb-3"  value="<?php  if
(isset($_GET['id_buku']))

```

```

                echo $result[0]["penerbit"] ?>" required>

```

```

                <label  for="tahun_terbit"  class="form-
label" style="color: #fff;">Tahun Terbit</label>

```

```

                <input  type="text"  name="tahun_terbit"
id="tahun_terbit"  class="form-control mb-3"  value="<?php  if
(isset($_GET['id_buku']))

```

```

                echo      $result[0]["tahun_terbit"]      ?>"
required>

```



```

        <?php

        ob_start();

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

            echo "<button type=\"submit\" class=\"btn
btn-success mt-3\" name=\"update\"> Update </button>";

        } else {

            echo "<button type=\"submit\" class=\"btn
btn-success mt-3\" name=\"tambah\"> Tambah </button>";

        }

        ?>

    </div>

</div>

</form>

```

penggunaan baris-baris tersebut merupakan bagian dari sebuah formulir HTML yang digunakan untuk menambah atau mengedit data buku dalam suatu aplikasi web. Penggunaan, `<div class="container form-container">` digunakan untuk membuat sebuah div dengan kelas container dan form-container. Div ini bertujuan untuk menempatkan elemen-elemen formulir dalam tata letak yang teratur dan terpusat di tengah halaman, sehingga meningkatkan tampilan dan kegunaan formulir dengan memberikan tata letak yang baik dan mudah diakses. Selanjutnya, `<div class="row">` digunakan untuk membuat sebuah div dengan kelas row. Div ini biasanya digunakan untuk menempatkan elemen-elemen formulir dalam satu baris secara horizontal, memastikan elemen-elemen formulir ditempatkan dalam satu baris sehingga tata letak formulir menjadi lebih terstruktur dan mudah dibaca. Kemudian, `<form action="" method="post">` membuat sebuah form HTML dengan method POST, yang digunakan untuk mengirim data ke server. Ini memungkinkan pengguna untuk memasukkan data buku dan mengirimkannya ke server untuk diproses. Judul "Tambah Data Buku" ditampilkan dengan menggunakan heading level 1 (h1) dan ditengahkan secara horizontal

menggunakan `<h1 style="text-align: center;" class="mt-2" style="color: #fff;">Tambah Data Buku</h1>`, memberikan informasi kepada pengguna bahwa formulir ini digunakan untuk menambah data buku. Setiap field data buku memiliki label dan input field, seperti `<label for="judul_buku" class="form-label" style="color: #fff;">Judul Buku</label>` dan `<input type="text" name="judul_buku" id="judul_buku" class="form-control mb-3" value="<?php if (isset($_GET['id_buku'])) echo $result[0]['judul_buku'] ?>" required>`, yang memungkinkan pengguna untuk memasukkan informasi yang diperlukan dengan cara yang terstruktur dan mudah dipahami. Jika formulir digunakan untuk mengedit data buku, maka nilai field akan diisi dengan data buku yang sudah ada, sehingga memudahkan pengguna dalam melakukan perubahan. Tombol `<button>` digunakan untuk menyimpan perubahan jika pengguna melakukan edit data buku, atau untuk menambahkan data baru jika formulir digunakan untuk menambah data.

Pada baris [138] [139] [140] [141] [142] [143] [144] [145] [146] [147], syntax

```
<?php

    ob_start();

    if (isset($_POST['update'])) {

        updateBuku($_GET['id_buku'],      $_POST['judul_buku'],
$_POST['penulis'],                      $_POST['penerbit'],
$_POST['tahun_terbit']);

    }

    if (isset($_POST['tambah'])) {

        insertDataBuku($_POST['id_buku'],
$_POST['judul_buku'], $_POST['penulis'], $_POST['penerbit'],
$_POST['tahun_terbit']);

    }

?>
```

penggunaan baris-baris tersebut merupakan bagian dari logika pemrosesan formulir PHP. Fungsi `ob_start()` digunakan untuk memulai output buffering, yang memungkinkan penangkapan sementara dari output yang dihasilkan oleh skrip PHP sebelum dikirimkan ke browser. Hal ini sering digunakan untuk menghindari masalah header yang dikirimkan sebelum konten HTML. Selanjutnya, kondisional `if (isset($_POST['update']))` digunakan untuk memeriksa apakah tombol "Update" pada formulir telah ditekan. Jika ya, maka fungsi `updateBuku()` akan dipanggil untuk memperbarui data buku sesuai dengan nilai yang dimasukkan dalam formulir. Ini memastikan bahwa data buku dapat diperbarui dengan mudah melalui formulir yang sesuai. Kemudian, kondisional `if (isset($_POST['tambah']))` digunakan untuk memeriksa apakah tombol "Tambah" pada formulir telah ditekan. Apabila ya, maka fungsi `insertDataBuku()` akan dipanggil untuk menambahkan data buku baru sesuai dengan nilai yang dimasukkan dalam formulir.

Pada baris [148], syntax `</body>` merupakan tag penutup dari baris 98.

Pada baris [149], syntax `</html>` merupakan tag penutup dari baris 8.

T. Source Code Peminjaman.php

Table 8. Source Code Modul 5 Peminjaman.php

1	<code><?php</code>
2	<code>require "Koneksi.php";</code>
3	<code>require 'Model.php';</code>
4	<code>\$result = getPeminjaman(\$conn);</code>
5	<code>\$dataMember = getMember(\$conn);</code>
6	<code>\$dataBuku = getBuku(\$conn);</code>
7	<code>if (isset(\$_GET['id_peminjaman'])) {</code>
8	<code> deletePeminjaman(\$_GET['id_peminjaman']);</code>
9	<code>}</code>
10	<code>?></code>
11	
12	<code><!DOCTYPE html></code>
13	<code><html lang="en"></code>
14	
15	<code><head></code>
16	<code> <meta charset="UTF-8"></code>
17	<code> <meta http-equiv="X-UA-Compatible"</code>
	<code>content="IE=edge"></code>

18	<meta name="viewport" content="width=device-width, initial-scale=1.0">
19	<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy12QvZ6jIW3" crossorigin="anonymous">
20	<title>Peminjaman</title>
21	<style>
22	body {
23	background-image: url('library_page.jpg');
24	background-size: cover;
25	background-position: center;
26	color: #ffffff;
27	font-family: Arial, sans-serif;
28	margin: 0;
29	display: flex;
30	flex-direction: column;
31	align-items: center;
32	justify-content: center;
33	overflow: auto;
34	}
35	
36	.buttons-container {
37	display: flex;
38	justify-content: flex-end;
39	margin-bottom: 20px;
40	}
41	
42	.buttons-container a:first-child {
43	margin-right: 10px;
44	}
45	
46	table,
47	tr,
48	td {
49	border-collapse: collapse;
50	border-radius: 21px;
51	padding: 20px;
52	font-size: 18px;
53	

```

54     }
55
56     table {
57         width: max-content;
58         border-radius: 21px;
59         background: rgba(255, 255, 255, 0.8);
60         color: black;
61     }
62
63     td {
64         width: 100px;
65         text-align: center;
66     }
67
68     button {
69         display: inline-block;
70         border-radius: 10px;
71         border: 1px solid #03045e;
72         position: relative;
73         overflow: hidden;
74         transition: all 0.5s ease-in;
75         z-index: 1;
76     }
77
78     button::before,
79     button::after {
80         content: '';
81         position: absolute;
82         top: 0;
83         width: 0;
84         height: 100%;
85         transform: skew(15deg);
86         transition: all 0.5s;
87         overflow: hidden;
88         z-index: -1;
89     }
90
91     button::before {
92         left: -10px;
93         background: #537188;
94     }

```

```

95
96     button::after {
97         right: -10px;
98         background: #537188;
99     }
100
101     button:hover::before,
102     button:hover::after {
103         width: 70%;
104     }
105
106     button:hover span {
107         color: #e0aaff;
108         transition: 0.3s;
109     }
110
111     button span {
112         color: #03045e;
113         font-size: 18px;
114         transition: all 0.3s ease-in;
115     }
116
117     .header {
118         background-color: rgba(128, 128, 128, 0.7);
119         border-radius: 10px;
120         padding: 20px;
121         margin-bottom: 20px;
122         width: max-content;
123     }
124
125     h1 {
126         font-size: 48px;
127         margin: 40px auto 20px;
128         color: #FFFAF0;
129         text-transform: uppercase;
130         letter-spacing: 2px;
131         font-weight: bold;
132         text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
133         animation: fadeInDown 1s ease-in-out,
scaleIn 1s ease-in-out;
134         border-bottom: 2px solid #FFFAF0;

```

```

135         padding-bottom: 10px;
136         background-color: rgba(0, 0, 0, 0.5);
137         width: 80%;
138         text-align: center;
139     }
140 </style>
141 </head>
142
143 <body class="p-3" style="background-color: #E1D4BB;">
144     <h2>
145         <center><h1>Data Peminjaman</h1></center>
146     </h2>
147     <div class="buttons-container">
148         <a href="index.php"><button class="btn btn-dark
149         mb-4">Kembali</button></a>
150         <a href="FormPeminjaman.php"><button class="btn
151         btn-warning mb-1">Tambah Data Peminjaman</button></a>
152     </div>
153     <table class="table">
154         <thead class="table-dark">
155             <tr>
156                 <th class="text-center">ID
157                 Peminjaman</th>
158                 <th class="text-center">Tanggal
159                 Pinjam</th>
160                 <th class="text-center">Tanggal
161                 Kembali</th>
162                 <th class="text-center">Judul Buku</th>
163                 <th class="text-center">Nama
164                 Member</th>
165                 <th class="text-center">Aksi</th>
166             </tr>
167         </thead>
168         <tbody>
169             <?php
170             foreach ($result as $hasil) {
171                 echo "<tr>";
172                 echo "    <td>" . $hasil["id_peminjaman"] .
173                 "</td>";
174                 echo "    <td>" . $hasil["tgl_pinjam"] .
175                 "</td>";

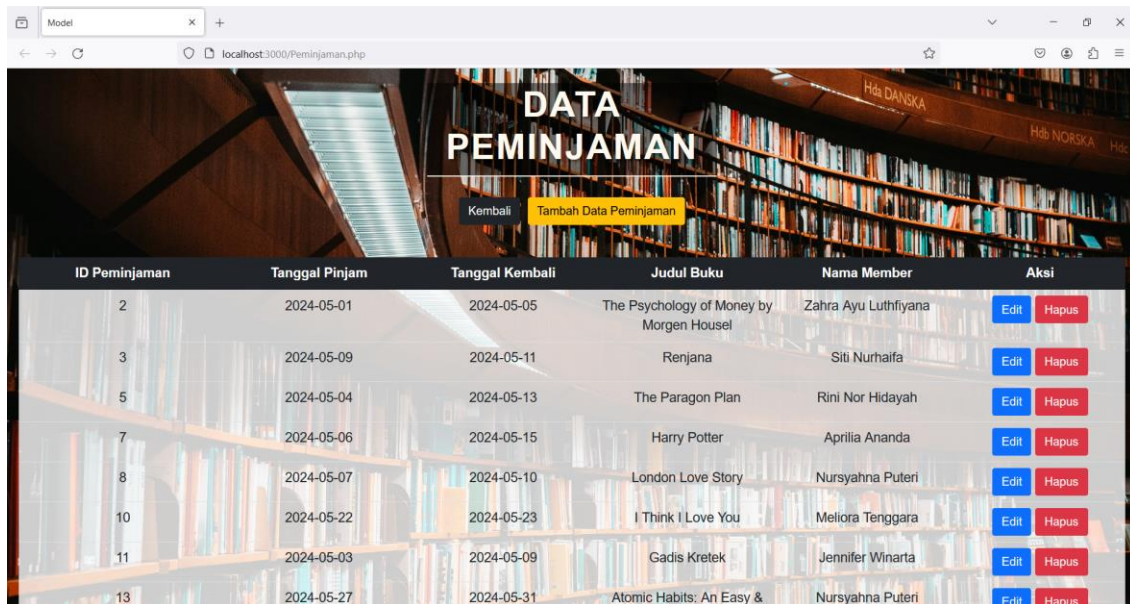
```

```

168         echo "<td>" . $hasil["tgl_kembali"] .
"</td>";
169         $judulBuku = '';
170         foreach ($dataBuku as $temp) {
171             if ($hasil['id_buku'] ==
$temp['id_buku']) {
172                 $judulBuku = $temp['judul_buku'];
173                 break;
174             }
175         }
176         echo "<td>" . $judulBuku . "</td>";
177         $namaMember = '';
178         foreach ($dataMember as $temp) {
179             if ($hasil['id_member'] ==
$temp['id_member']) {
180                 $namaMember = $temp['nama_member'];
181                 break;
182             }
183         }
184         echo "<td>" . $namaMember . "</td>";
185         echo "<td>";
186         echo "<a class='btn btn-primary'
href='FormPeminjaman.php?id_peminjaman="
$hasil['id_peminjaman'] . "'>Edit</a>";
187         echo " ";
188         echo "<a class='btn btn-danger'
href='Peminjaman.php?id_peminjaman="
$hasil['id_peminjaman'] . "' onclick=\"return
confirm('Yakin Ingin Dihapus?')\">Hapus</a>";
189         echo "</td>";
190         echo "</tr>";
191     }
192     ?>
193 </tbody>
194 </table>
195 </body>
196 </html>

```


U. Output Program Peminjaman.php



ID Peminjaman	Tanggal Pinjam	Tanggal Kembali	Judul Buku	Nama Member	Aksi
2	2024-05-01	2024-05-05	The Psychology of Money by Morgen Housel	Zahra Ayu Luthfiyana	Edit Hapus
3	2024-05-09	2024-05-11	Renjana	Siti Nurhalifa	Edit Hapus
5	2024-05-04	2024-05-13	The Paragon Plan	Rini Nor Hidayah	Edit Hapus
7	2024-05-06	2024-05-15	Harry Potter	Aprilia Ananda	Edit Hapus
8	2024-05-07	2024-05-10	London Love Story	Nursyahna Puteri	Edit Hapus
10	2024-05-22	2024-05-23	I Think I Love You	Meliora Tenggara	Edit Hapus
11	2024-05-03	2024-05-09	Gadis Kretek	Jennifer Winarta	Edit Hapus
13	2024-05-27	2024-05-31	Atomic Habits: An Easy &	Nursyahna Puteri	Edit Hapus

Gambar 6. Screenshot Hasil Jawaban Modul 5 Peminjaman.php

V. Pembahasan Peminjaman.php

Pada baris [1] [2] [3] [4] [5] [6] [7] [8] [9] [10], syntax

```
<?php  
  
require "Koneksi.php";  
  
require 'Model.php';  
  
$result = getPeminjaman($conn);  
  
$dataMember = getMember($conn);  
  
$dataBuku = getBuku($conn);  
  
if (isset($_GET['id_peminjaman'])) {  
    deletePeminjaman($_GET['id_peminjaman']);  
}  
  
?>
```

penggunaan baris-baris tersebut bertujuan untuk mempersiapkan dan mengelola data yang akan ditampilkan di halaman web. Dengan menggunakan kode tersebut menghubungkan

dengan database dengan menggunakan require untuk memasukkan file Koneksi.php, yang kemungkinan berisi informasi koneksi ke database. Selanjutnya, kode tersebut memanggil fungsi getPeminjaman, getMember, dan getBuku dari file Model.php. Fungsi-fungsi ini bertugas mengambil data-data terkait peminjaman, anggota, dan buku dari database. Kemudian, kode tersebut memiliki fungsi untuk penghapusan data peminjaman. Hal ini dilakukan dengan memeriksa apakah terdapat parameter id_peminjaman yang diterima melalui URL (\$_GET['id_peminjaman']). Apabila ada, maka panggilan fungsi deletePeminjaman dari Model.php digunakan untuk menghapus data peminjaman yang sesuai dengan ID yang diterima. Selain itu, kode tersebut juga melakukan penyimpanan data hasil panggilan fungsi getPeminjaman, getMember, dan getBuku ke dalam variabel \$result, \$dataMember, dan \$dataBuku masing-masing. Hal ini bertujuan untuk digunakan dalam tampilan halaman web untuk menampilkan data peminjaman, anggota, dan buku.

Pada baris [12], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisikan informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [13], syntax `<html lang="en">` tag html merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [15], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa ditambahkan pada tag ini.

Pada baris [16], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan

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

Pada baris [17], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [18], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada perangkat mobile. Atribut yang termuat di dalamnya, berupa menampilkan lebar dokumen, mengatur tingkat pembesaran.

Pada baris [19] [20], syntax

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

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j
IW3" crossorigin="anonymous">
```

penggunaan baris-baris tersebut adalah untuk menyertakan file eksternal seperti CSS. Terdapat penggunaan atribut penting dalam tag ini yaitu, `href=https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css` digunakan untuk menentukan URL dari stylesheet yang akan disertakan. URL ini mengarah ke file CSS Bootstrap versi 5.1.3 yang dihosting oleh CDN atau Content Delivery Network jsDelivr. Atribut lain yang juga digunakan dalam tag `<link>` adalah `rel`, yang dengan nilai `rel="stylesheet"` menunjukkan bahwa file yang disertakan adalah stylesheet. Selanjutnya, penggunaan atribut `integrity` seperti dalam `integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"` berisikan hash kriptografis yang digunakan untuk memastikan bahwa file yang diambil tidak diubah. Hash ini dihitung menggunakan algoritma SHA-384. Saat browser memuat file, maka

akan dilakukan penghitungan hash dari file yang diunduh dan membandingkannya dengan nilai ini. Apabila hash tidak cocok, maka browser akan membatalkan pemuatan file, sehingga tujuan atribut ini adalah untuk pencegahan kemungkinan serangan dari sumber file yang dimodifikasi. Dan untuk penggunaan atribut `crossorigin` dengan nilai `crossorigin="anonymous"` digunakan sebagai penentuan bagaimana permintaan sumber daya lintas domain harus dijalankan. Penggunaan, `anonymous` berarti permintaan lintas domain akan dilakukan tanpa mengirimkan kredensial pengguna seperti cookie atau otorisasi HTTP.

Pada baris [21], syntax `<title>Peminjaman</title>` tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas untuk memberikan informasi berupa judul dokumen HTML, atau teks pada judul browser pada tab browser saat halaman tersebut diakses. Tag ini berada sejajar dengan tag meta sebelumnya.

Pada baris [23] – [140], syntax

`<style>`

```
body {  
  
    background-image: url('library_page.jpg');  
  
    background-size: cover;  
  
    background-position: center;  
  
    color: #ffffff;  
  
    font-family: Arial, sans-serif;  
  
    margin: 0;  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    justify-content: center;  
  
    overflow: auto;
```

```
}
```

```
.buttons-container {  
    display: flex;  
    justify-content: flex-end;  
    margin-bottom: 20px;  
}
```

```
.buttons-container a:first-child {  
    margin-right: 10px;  
}
```

```
table,  
tr,  
td {  
    border-collapse: collapse;  
    border-radius: 21px;  
    padding: 20px;  
    font-size: 18px;  
}
```

```
table {  
    width: max-content;
```

```
border-radius: 21px;

background: rgba(255, 255, 255, 0.8);

color: black;

}
```

```
td {

    width: 100px;

    text-align: center;

}
```

```
button {

    display: inline-block;

    border-radius: 10px;

    border: 1px solid #03045e;

    position: relative;

    overflow: hidden;

    transition: all 0.5s ease-in;

    z-index: 1;

}
```

```
button::before,

button::after {

    content: '';
```

```
    position: absolute;

    top: 0;

    width: 0;

    height: 100%;

    transform: skew(15deg);

    transition: all 0.5s;

    overflow: hidden;

    z-index: -1;
}
```

```
button::before {

    left: -10px;

    background: #537188;

}
```

```
button::after {

    right: -10px;

    background: #537188;

}
```

```
button:hover::before,

button:hover::after {

    width: 70%;

}
```

```
}
```

```
button:hover span {  
    color: #e0aaff;  
    transition: 0.3s;  
}
```

```
button span {  
    color: #03045e;  
    font-size: 18px;  
    transition: all 0.3s ease-in;  
}
```

```
.header {  
    background-color: rgba(128, 128, 128, 0.7);  
    border-radius: 10px;  
    padding: 20px;  
    margin-bottom: 20px;  
    width: max-content;  
}
```

```
h1 {  
    font-size: 48px;
```



```

margin: 40px auto 20px;

color: #FFFAF0;

text-transform: uppercase;

letter-spacing: 2px;

font-weight: bold;

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);

animation: fadeInDown 1s ease-in-out, scaleIn 1s
ease-in-out;

border-bottom: 2px solid #FFFAF0;

padding-bottom: 10px;

background-color: rgba(0, 0, 0, 0.5);

width: 80%;

text-align: center;

}

</style>

```

penggunaan baris-baris tersebut adalah kumpulan aturan gaya (CSS) yang ditujukan untuk mengatur tampilan dan tata letak elemen-elemen pada halaman web. Bagian "Body Styling" mengatur gambar latar belakang dengan background-image, serta menetapkan tata letak dan penempatan elemen-elemen menggunakan properti seperti display: flex, flex-direction: column, align-items: center, dan justify-content: center. Selain itu, bagian ini juga mengatur warna teks, jenis font, dan margin untuk meningkatkan keterbacaan dan tampilan keseluruhan halaman. Bagian "Buttons Container" fokus pada penataan tombol dan container tombol, dengan tombol-tombol ditempatkan di sebelah kanan bawah halaman untuk memudahkan akses pengguna. Bagian "Table Styling" bertujuan untuk meningkatkan tampilan tabel, baris, dan sel dengan mengatur padding, ukuran teks, border-collapse, serta memberikan efek hover pada seluruh sel tabel. Di sisi lain, aturan gaya dalam "Button

Styling" menambahkan efek transisi dan bayangan pada tombol, serta mengatur warna, ukuran, dan border-radius tombol, sambil memberikan efek hover untuk mengubah warna teks tombol saat disorot oleh kursor. Bagian "Header Styling" memusatkan perhatian pada penampilan header, termasuk warna latar belakang, padding, dan margin, dengan tambahan efek bayangan dan border bawah untuk meningkatkan kerapian dan visualisasi. Terakhir, aturan gaya dalam "Judul H1 Styling" menetapkan tampilan judul H1, termasuk ukuran teks, warna, penempatan, dan efek bayangan, yang membantu menonjolkan judul halaman.

Pada baris [141], syntax `</head>` merupakan tag penutup dari baris 15.

Pada baris [143] [144] [145] [146] [147] [148] [149] [150] [151] [152] [153] [154] [155] [156] [157] [158] [159] [160] [161], syntax

```
<body class="p-3" style="background-color: #E1D4BB;">

    <h2>

        <center><h1>Data Peminjaman</h1></center>

    </h2>

    <div class="buttons-container">

        <a href="index.php"><button class="btn btn-dark mb-4">Kembali</button></a>

        <a href="FormPeminjaman.php"><button class="btn btn-warning mb-1">Tambah Data Peminjaman</button></a>

    </div>

    <table class="table">

        <thead class="table-dark">

            <tr>

                <th class="text-center">ID Peminjaman</th>

                <th class="text-center">Tanggal Pinjam</th>
```

```

        <th class="text-center">Tanggal Kembali</th>

        <th class="text-center">Judul Buku</th>

        <th class="text-center">Nama Member</th>

        <th class="text-center">Aksi</th>

    </tr>

</thead>

```

penggunaan kode-kode tersebut adalah bagian penting dari struktur HTML yang bertujuan untuk mengelola tampilan halaman web yang diperlukan. Setiap elemen dalam kode tersebut memiliki fungsi khusus yang mendukung tujuan keseluruhan halaman. Penggunaan, body class dengan atribut "p-3" memberikan padding sebesar 3 unit pada semua sisi elemen body, memastikan konten terletak dengan nyaman di dalamnya. Judul H1 "<h1>Data Peminjaman</h1>" menonjolkan judul halaman dengan ukuran yang besar, memberikan pengguna informasi yang jelas tentang konten yang ditampilkan. Bagian tombol navigasi, yang terkandung dalam div container "<div class='buttons-container'>", memberikan pengguna akses mudah ke halaman lain. Tombol "Kembali" dengan link "<button class='btn btn-dark mb-4'>Kembali</button>" memungkinkan pengguna untuk kembali ke halaman utama, sementara tombol "Tambah Data Peminjaman" "<button class='btn btn-warning mb-1'>Tambah Data Peminjaman</button>" mengarahkan pengguna ke halaman formulir untuk menambah data peminjaman baru. Selanjutnya, tabel data peminjaman "<table class='table'>" dirancang untuk menampilkan informasi peminjaman dengan menggunakan kelas dari Bootstrap untuk penataan yang lebih baik. Bagian "<thead class='table-dark'>" mengatur tampilan header tabel dengan latar belakang gelap, menciptakan kontras yang jelas dengan konten tabel. Setiap baris dalam tabel, ditandai dengan "<tr>", menampilkan judul kolom-kolom dengan menggunakan "<th class='text-center'>" untuk memposisikan teksnya di tengah.

Pada baris [162] [163] [164] [165] [166] [167] [168] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183] [184] [185] [186] [187] [188] [189] [190] [191] [192] [193], syntax

```

<tbody>

    <?php

        foreach ($result as $hasil) {

            echo "<tr>";

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

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

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

            $judulBuku = '';

            foreach ($dataBuku as $temp) {

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

                    $judulBuku = $temp['judul_buku'];

                    break;

                }

            }

            echo "<td>" . $judulBuku . "</td>";

            $namaMember = '';

            foreach ($dataMember as $temp) {

                if ($hasil['id_member'] == $temp['id_member'])

{

                    $namaMember = $temp['nama_member'];

                    break;

                }

            }

}

```

```

        echo "<td>" . $namaMember . "</td>";

        echo "<td>";

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

        echo " ";

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

        echo "</td>";

        echo "</tr>";

    }

?>

</tbody>

```

Penggunaan baris-baris tersebut adalah proses pembuatan baris-baris dalam table data peminjaman pada halaman web. Penggunaan <tbody> digunakan untuk menandai awal dari bagian tubuh tabel. Di sini, sebuah loop foreach dimulai dengan <?php foreach (\$result as \$hasil) { ?>, yang bertujuan untuk mengulangi setiap item dalam array \$result. Setiap iterasi dari loop tersebut akan menampilkan sebuah baris tabel yang dimulai dengan <tr>. Dalam setiap baris tabel, data peminjaman ditampilkan dalam beberapa kolom. Misalnya, echo "<td>" . \$hasil["id_peminjaman"] . "</td>"; menampilkan nilai dari kunci "id_peminjaman" dari setiap item dalam array \$result sebagai sebuah sel data dalam kolom tabel. Sama halnya dengan kolom "tgl_pinjam" dan "tgl_kembali", yang menampilkan tanggal peminjaman dan tanggal pengembalian. Namun, informasi tambahan seperti judul buku dan nama member juga disertakan dalam tabel. Untuk mendapatkan judul buku dan nama member, dilakukan loop foreach terpisah untuk setiap data buku dan data member. Misalnya, dengan mengakses data buku melalui \$dataBuku, dan jika id buku dalam data peminjaman sama dengan id buku

dalam data buku, judul buku akan ditampilkan dalam kolom yang sesuai. Sama seperti untuk mendapatkan nama member, dilakukan loop foreach terhadap \$dataMember dan memeriksa kesesuaian id member. Setelah semua data ditampilkan dalam kolom yang sesuai, tombol aksi "Edit" dan "Hapus" juga disertakan di setiap baris. Tombol "Edit" akan mengarahkan pengguna ke halaman FormPeminjaman.php untuk mengedit data peminjaman, sedangkan tombol "Hapus" akan mengarahkan ke halaman Peminjaman.php untuk menghapus data peminjaman yang terkait. Sebuah konfirmasi akan muncul sebelum penghapusan untuk memastikan tindakan tersebut diinginkan. Setelah loop foreach selesai, elemen `</tbody>` menandakan akhir dari bagian tubuh tabel

Pada baris [194], syntax `</table>` merupakan tag penutup dari baris 151.

Pada baris [195], syntax `</body>` merupakan tag penutup dari baris 143.

Pada baris [196], syntax `</html>` merupakan tap penutup dari baris 13.

W. Source Code FormPeminjaman.php

Table 9. Source Code Modul 5 FormPeminjaman.php

1	<code><?php</code>
2	<code>require "Koneksi.php";</code>
3	<code>require 'Model.php';</code>
4	
5	<code>if (isset(\$_GET['id_peminjaman'])) {</code>
6	<code> editPeminjaman();</code>
7	<code>}</code>
8	
9	<code>\$dataMember = getMember(\$conn);</code>
10	<code>\$dataBuku = getBuku(\$conn);</code>
11	
12	<code>if (isset(\$_POST['update'])) {</code>
13	<code> updatePeminjaman(\$_GET["id_peminjaman"],</code>
	<code> \$_POST["tgl_pinjam"], \$_POST["tgl_kembali"],</code>
	<code> \$_POST["id_buku"], \$_POST["id_member"]);</code>

14	}
15	if (isset(\$_POST['tambah'])) {
16	insertDataPeminjaman(\$_POST["tgl_pinjam"],
	\$_POST["tgl_kembali"], \$_POST["id_buku"],
	\$_POST["id_member"]);
17	}
18	?>
19	
20	<!DOCTYPE html>
21	<html lang="en">
22	
23	<head>
24	<meta charset="UTF-8">
25	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
26	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
27	<link
	href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dis
	t/css/bootstrap.min.css" rel="stylesheet"
28	integrity="sha384-
	1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqy
	l2QvZ6jIW3" crossorigin="anonymous">
29	<title>Form Peminjaman</title>
30	<style>
31	body {
32	background-image: url('library_page.jpg');
33	background-size: cover;
34	background-position: center;
35	color: #ffffff;
36	font-family: Arial, sans-serif;

37	margin: 0;
38	display: flex;
39	justify-content: center;
40	align-items: center;
41	height: 100vh;
42	overflow: auto;
43	}
44	input {
45	font-weight: 500;
46	font-size: .8vw;
47	color: #fff;
48	background-color: rgb(28, 28, 30);
49	box-shadow: 0 0 .4vw rgba(0, 0, 0, 0.5), 0
	0 0 .15vw transparent;
50	border-radius: 0.4vw;
51	border: none;
52	outline: none;
53	padding: 0.4vw;
54	transition: .4s;
55	}
56	.form-container {
57	background-color: rgba(225, 212, 187, 0.5);
58	padding: 20px;
59	border-radius: 10px;
60	}
61	input:hover {
62	box-shadow: 0 0 0 .15vw rgba(135, 207, 235,
	0.186);
63	}
64	
65	input:focus {


```
66         box-shadow: 0 0 0 .15vw skyblue;
67     }
68
69     button {
70         display: inline-block;
71         border-radius: 10px;
72         border: 1px solid #03045e;
73         position: relative;
74         overflow: hidden;
75         transition: all 0.5s ease-in;
76         z-index: 1;
77     }
78
79     button::before,
80     button::after {
81         content: '';
82         position: absolute;
83         top: 0;
84         width: 0;
85         height: 100%;
86         transform: skew(15deg);
87         transition: all 0.5s;
88         overflow: hidden;
89         z-index: -1;
90     }
91
92     button::before {
93         left: -10px;
94         background: #537188;
95     }
96
```

97	button::after {
98	right: -10px;
99	background: #537188;
100	}
101	
102	button: hover::before,
103	button: hover::after {
104	width: 70%;
105	}
106	
107	button: hover span {
108	color: #e0aaff;
109	transition: 0.3s;
110	}
111	
112	button span {
113	color: #03045e;
114	font-size: 18px;
115	transition: all 0.3s ease-in;
116	}
117	</style>
118	</head>
119	
120	<div class="container form-container">
121	<div class="container">
122	<div class="row">
123	<form action="" method="post">
124	<h1 style="text-align: center;"
	class="mt-2">Tambah Data Peminjaman</h1>
125	<label for="tgl_pinjam" class="form-
	label">Tanggal Peminjaman</label>

126	<code><input type="date" name="tgl_pinjam"</code>
	<code>id="tgl_pinjam" class="form-control mb-3" value="<?php</code>
	<code>if (isset(\$_GET['id_peminjaman']))</code>
127	<code>echo \$result[0]["tgl_pinjam"] ?>"</code>
	<code>required></code>
128	
129	<code><label for="tgl_kembali"</code>
	<code>class="form-label">Tanggal Kembali</label></code>
	<code><input type="date"</code>
130	<code>name="tgl_kembali" id="tgl_kembali" class="form-control</code>
	<code>mb-3" value="<?php if (isset(\$_GET['id_peminjaman']))</code>
	<code>echo \$result[0]["tgl_kembali"] ?>"</code>
131	<code>required></code>
132	<code><label for="id_buku">Judul</code>
133	<code>Buku:</label>
</code>
	<code><select id="id_buku"</code>
134	<code>name="id_buku"></code>
	<code><?php</code>
135	<code>if (!isset(\$_GET['id_peminjaman'])) {</code>
136	<code>?></code>
	<code><option disabled</code>
137	<code>selected></option></code>
	<code><?php</code>
138	<code>}</code>
139	<code>foreach (\$dataBuku as \$barisBuku) {</code>
140	<code>\$selected = '';</code>
141	<code>if (</code>
142	<code>isset(\$_GET['id_peminjaman'])</code>
143	<code>&& \$hasil[0]['id_buku'] ==</code>

```

144         $barisBuku['id_buku']
145     ) {
146
147         $selected = 'selected';
148     }
149     ?>
150     <option          value="<?php          echo
$barisBuku['id_buku']; ?>"    <?php    echo    $selected;
?>><?php echo $barisBuku['judul_buku']; ?></option>
151         <?php } ?>
        </select>
        <br><br>
152
153         <label          for="id_member">Nama
154 Member:</label><br>
155         <select          id="id_member"
156 name="id_member">
        <?php
157         if (!isset($_GET['id_peminjaman']))
{ ?>
158         <option          disabled
159 selected></option>
        <?php
160         }
        foreach          ($dataMember          as
161 $barisMember) {
162         $selected = '';
163         if (
164 isset($_GET['id_peminjaman']) && $hasil[0]['id_member']
165 ==

```

```

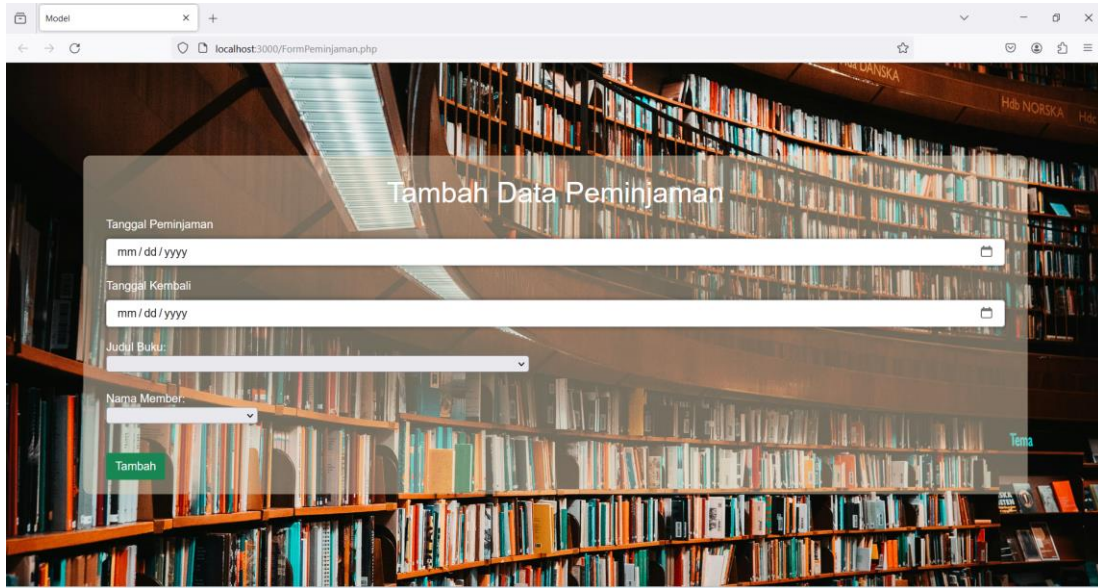
166
167         $barisMember['id_member']
168         ) {
169
170             $selected = 'selected';
171         }
172         ?>
173         <option value="<?php echo
174 $barisMember['id_member']; ?>" <?php echo $selected;
175 ?>><?php echo $barisMember['nama_member']; ?></option>
176         <?php } ?>
177         </select>
178         <br><br>
179
180         <?php
181         if (isset($_GET['id_peminjaman'])) {
182             echo "<button type=\"submit\"
183 class=\"btn btn-success mt-3\" name=\"update\"> Update
184 </button>";
185         } else {
186             echo "<button type=\"submit\"
187 class=\"btn btn-success mt-3\" name=\"tambah\"> Tambah
188 </button>";
189         }
190         ?>
191
192         </div>
193     </div>
194 </form>
195 </body>
196 </html>
197
198

```

189

190

X. Output Program FormPeminjaman.php



Gambar 7. Screenshot Hasil Jawaban Modul 5 FormPeminjaman.php

Y. Pembahasan FormPeminjaman.php

Pada baris [1] [2] [3], syntax

```
<?php
require "Koneksi.php";
require 'Model.php';
```

penggunaannya baris kode tersebut bertujuan untuk mengimpor (require) dua file PHP yang penting untuk menjalankan skrip saat ini. Penggunaan, file "Koneksi.php" berisi konfigurasi koneksi ke database, termasuk informasi seperti host, nama pengguna, kata sandi, dan nama database. Dengan mengimpor file ini, skrip dapat terhubung ke database untuk melakukan berbagai operasi seperti pengambilan, pembaruan, atau penghapusan data. Kemudian, file 'Model.php' kemungkinan berisi definisi dari fungsi-fungsi yang terkait dengan model data dalam aplikasi. Hal ini dapat mencakup fungsi-fungsi yang bertanggung jawab atas

pengambilan, pembaruan, atau penghapusan data dari database, serta fungsi-fungsi lain yang terkait dengan logika bisnis aplikasi. Dengan mengimpor kedua file ini, skrip PHP memperoleh akses ke fungsionalitas yang disediakan di dalamnya. Sehingga memungkinkan skrip untuk menjalankan berbagai operasi yang diperlukan, seperti mengambil data dari database atau memanipulasi model data sesuai dengan kebutuhan aplikasi yang sedang berjalan.

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

```
if (isset($_GET['id_peminjaman'])) {  
    editPeminjaman();  
}
```

Penggunaan kode tersebut bertujuan untuk memeriksa keberadaan parameter `id_peminjaman` dalam URL menggunakan fungsi `isset($_GET['id_peminjaman'])`. Fungsi ini digunakan dalam konteks aplikasi web untuk menentukan apakah pengguna sedang dalam mode pengeditan atau penambahan data. Apabila parameter `id_peminjaman` ada dalam URL, maka dapat disimpulkan bahwa pengguna sedang dalam mode pengeditan data peminjaman dengan ID tertentu. Dalam hal ini, fungsi `editPeminjaman()` akan dipanggil untuk menangani proses pengeditan data peminjaman yang spesifik tersebut. Pemeriksaan keberadaan parameter dalam URL adalah langkah penting dalam pengembangan aplikasi web yang memungkinkan respons yang sesuai dengan tindakan pengguna. Dengan memeriksa keberadaan parameter seperti `id_peminjaman`, aplikasi dapat menyesuaikan perilakunya, seperti menampilkan formulir pengeditan data, menyimpan perubahan yang dibuat, atau mengarahkan pengguna ke halaman yang sesuai. Sehingga memungkinkan aplikasi untuk memberikan pengalaman pengguna yang lebih dinamis dan responsif, sesuai dengan kebutuhan dan interaksi yang terjadi.

Pada baris [9] [10], syntax

```
$dataMember = getMember($conn);  
  
$dataBuku = getBuku($conn);
```

penggunaan kode tersebut bertujuan untuk mendapatkan data member dan data buku dari database menggunakan fungsi `getMember()` dan `getBuku()` yang telah didefinisikan di dalam file `Model.php`. Variabel `$dataMember` akan berisi data dari tabel member, sedangkan `$dataBuku` akan berisi data dari tabel buku. Kedua data ini kemungkinan akan digunakan dalam proses selanjutnya, seperti menampilkan informasi ke layar atau melakukan operasi CRUD (Create, Read, Update, Delete) terhadap data tersebut.

Pada abris [12] [13] [14] [15] [16] [17] [18] [19], syntax

```
if (isset($_POST['update'])) {  
  
    updatePeminjaman($_GET["id_peminjaman"],  
$_POST["tgl_pinjam"],           $_POST["tgl_kembali"],  
$_POST["id_buku"], $_POST["id_member"]);  
  
}  
  
if (isset($_POST['tambah'])) {  
  
    insertDataPeminjaman($_POST["tgl_pinjam"],  
$_POST["tgl_kembali"],           $_POST["id_buku"],  
$_POST["id_member"]);  
  
}  
  
?>
```

penggunaan kode tersebut bertanggung jawab atas pemrosesan data yang diterima melalui metode POST. Ketika variabel POST dengan nama 'update' terdeteksi, langkah selanjutnya adalah memanggil fungsi `updatePeminjaman()`. Fungsi ini bertugas memperbarui data peminjaman di dalam database, dengan memperhitungkan nilai-nilai yang diterima dari formulir, seperti tanggal pinjam, tanggal kembali, id buku, dan id member. Selanjutnya, jika terdapat variabel POST dengan nama 'tambah', fungsi `insertDataPeminjaman()` akan dijalankan. Tugasnya adalah menambahkan data peminjaman baru ke dalam basis data, berdasarkan informasi yang diperoleh dari formulir. Data tambahan ini termasuk tanggal pinjam, tanggal kembali, id buku, dan id member. Baik fungsi `updatePeminjaman()` maupun `insertDataPeminjaman()` memiliki peran krusial dalam manajemen data peminjaman

aplikasi, memastikan bahwa informasi yang dimasukkan oleh pengguna diproses dengan benar dan disimpan secara akurat dalam sistem.

Pada baris [20], syntax `<!DOCTYPE html>` tag ini sebagai mendefinisikan informasi tipe dokumen bahwa suatu dokumen HTML adalah HTML5.

Pada baris [21], syntax `<html lang="en">` tag `html` merupakan tag wajib yang mendefinisikan bahwa dokumen adalah dokumen HTML, tag dasar ini sebagai tanda awal atau tag pertama dalam dokumen HTML, tag tersebut juga yang akan memuat semua tag HTML lainnya. Pada baris ini juga terdapat atribut `lang="en"` yang memiliki pengertian bahwa dokumen HTML5 yang telah dibuat berbahasa Inggris.

Pada baris [23], syntax `<head>` merupakan bagian kepala dari HTML. Tag Head berfungsi sebagai penampung beragam informasi terkait dokumen HTML. Pada tag Head ini bisa ditambahkan tag-tag yang biasanya digunakan untuk memberikan informasi berupa penulis, judul dokumen, kata kunci pada dokumen dan masih banyak lagi informasi yang bisa ditambahkan pada tag ini.

Pada baris [24], syntax `<meta charset="UTF-8">` tag ini memiliki arti bahwa dokumen HTML5 yang telah dibuat menggunakan pengodean karakter UTF-8. Charset merupakan kumpulan kode-kode bit komputer dengan pasangan karakter yang harus ditampilkan. Untuk UTF-8 merupakan kepanjangan dari (Unicode Transformation Format-8). Sehingga tag ini akan memberikan instruksi kepada web browser untuk menerjemahkan karakter-karakter didalam halaman HTML sebagai UTF-8 tersebut. Tag ini berada lebih masuk ke dalam dibandingkan dengan tag sebelumnya.

Pada baris [25], syntax `<meta http-equiv="X-UA-Compatible" content="IE=edge">` tag meta ini digunakan untuk mendefinisikan dokumen HTML agar ditampilkan pada Internet Explorer versi terbaru. Tag meta ini sejajar dengan tag meta lainnya.

Pada baris [26], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` tag meta viewport sering digunakan pada web responsive. Digunakan sebagai pengontrol bagaimana dokumen HTML ditampilkan pada

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

Pada baris [27] [28], syntax

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

    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j
IW3" crossorigin="anonymous">
```

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

Pada baris [29], syntax <title>Form Peminajamn</title> tag Title merupakan salah satu contoh informasi yang terdapat di dalam tag Head. Pada tag ini memiliki tugas

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

Pada baris [30] – [117], syntax

```
<style>

    body {

        background-image: url('library_page.jpg');

        background-size: cover;

        background-position: center;

        color: #ffffff;

        font-family: Arial, sans-serif;

        margin: 0;

        display: flex;

        justify-content: center;

        align-items: center;

        height: 100vh;

        overflow: auto;

    }

    input {

        font-weight: 500;

        font-size: .8vw;

        color: #fff;

        background-color: rgb(28, 28, 30);
```

```

        box-shadow: 0 0 .4vw rgba(0, 0, 0, 0.5), 0 0 0
.15vw transparent;

        border-radius: 0.4vw;

        border: none;

        outline: none;

        padding: 0.4vw;

        transition: .4s;
    }

    .form-container {

        background-color: rgba(225, 212, 187, 0.5);

        padding: 20px;

        border-radius: 10px;
    }

    input:hover {

        box-shadow: 0 0 0 .15vw rgba(135, 207, 235, 0.186);
    }

    input:focus {

        box-shadow: 0 0 0 .15vw skyblue;
    }

    button {

        display: inline-block;

```

```
border-radius: 10px;

border: 1px solid #03045e;

position: relative;

overflow: hidden;

transition: all 0.5s ease-in;

z-index: 1;

}
```

```
button::before,

button::after {

    content: '';

    position: absolute;

    top: 0;

    width: 0;

    height: 100%;

    transform: skew(15deg);

    transition: all 0.5s;

    overflow: hidden;

    z-index: -1;

}
```

```
button::before {

    left: -10px;
```

```
        background: #537188;
    }

```

```
button::after {
    right: -10px;
    background: #537188;
}

```

```
button:hover::before,
button:hover::after {
    width: 70%;
}

```

```
button:hover span {
    color: #e0aaff;
    transition: 0.3s;
}

```

```
button span {
    color: #03045e;
    font-size: 18px;
    transition: all 0.3s ease-in;
}

```

```
</style>
```

penggunaan kode-kode tersebut merupakan kode CSS, penggunaan properti yang diterapkan pada elemen body mengontrol aspek-aspek penting seperti gambar latar belakang, warna teks, jenis font, dan tata letak keseluruhan halaman. Kemudian, untuk elemen input, berbagai properti ditetapkan untuk mengatur tampilan input, termasuk font, warna teks, latar belakang, bayangan, dan transisi saat pengguna mengarahkan kursor ke input tersebut. Selanjutnya, pengaturan pada kelas `.form-container` bertujuan untuk merancang tampilan kontainer formulir. Properti seperti warna latar belakang, padding, dan border-radius diterapkan untuk memberikan tampilan yang konsisten dan menarik pada area formulir. Pada baris tersebut, gaya tombol juga diatur secara detail menggunakan properti CSS. Elemen button mendapat penyesuaian seperti border-radius, border, bayangan, dan transisi saat tombol disorot dengan kursor. Efek tambahan sebelum dan sesudah tombol diimplementasikan melalui pseudo-elements `button::before` dan `button::after` dengan menggunakan efek skewing (memiringkan), yang memberikan kesan visual yang menarik saat tombol dihover. Lalu, untuk menyesuaikan tampilan teks di dalam tombol, elemen button span juga mendapat perhatian dengan pengaturan warna teks dan transisi yang sesuai. Dengan semua pengaturan ini, kombinasi CSS tersebut memastikan bahwa halaman web akan memiliki tampilan yang menarik dan responsif ketika digunakan untuk mengisi formulir.

Pada baris [118], syntax `</head>` merupakan tag penutup dari bari 23.

Pada baris [120] – [188], syntax

```
<div class="container form-container">

  <div class="container">

    <div class="row">

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

        <h1 style="text-align: center;" class="mt-2">Tambah Data Peminjaman</h1>

        <label for="tgl_pinjam" class="form-label">Tanggal Peminjaman</label>
```

```

                <input          type="date"          name="tgl_pinjam"
id="tgl_pinjam"  class="form-control mb-3"  value="<?php if
(isset($_GET['id_peminjaman']))

                echo          $result[0]["tgl_pinjam"]          ?>"
required>

```

```

                <label    for="tgl_kembali"    class="form-
label">Tanggal Kembali</label>

```

```

                <input    type="date"    name="tgl_kembali"
id="tgl_kembali"  class="form-control mb-3"  value="<?php if
(isset($_GET['id_peminjaman']))

                echo    $result[0]["tgl_kembali"]    ?>"
required>

```

```

                <label                                for="id_buku">Judul
Buku:</label><br>

```

```

                <select id="id_buku" name="id_buku">

                <?php

                if (!isset($_GET['id_peminjaman'])) { ?>

                <option disabled selected></option>

                <?php

                }

                foreach ($dataBuku as $barisBuku) {

                $selected = '';

                if (

```



```

                                isset($_GET['id_peminjaman'])      &&
$hasil[0]['id_buku'] ==

                                $barisBuku['id_buku']

                                ) {

                                $selected = 'selected';

                                }

                                ?>

                                <option          value="<?php          echo
$barisBuku['id_buku']; ?>" <?php echo $selected; ?>><?php echo
$barisBuku['judul_buku']; ?></option>

                                <?php } ?>

                                </select>

                                <br><br>

                                <label          for="id_member">Nama
Member:</label><br>

                                <select id="id_member" name="id_member">

                                <?php

                                if (!isset($_GET['id_peminjaman'])) { ?>

                                <option disabled selected></option>

                                <?php

                                }

```

```

        foreach ($dataMember as $barisMember) {

            $selected = '';

            if (

                isset($_GET['id_peminjaman'])    &&
$hasil[0]['id_member'] ==

                $barisMember['id_member']

            ) {

                $selected = 'selected';

            }

            ?>

            <option          value="<?php          echo
$barisMember['id_member']; ?>" <?php echo $selected; ?>><?php
echo $barisMember['nama_member']; ?></option>

            <?php } ?>

        </select>

        <br><br>

        <?php

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

            echo "<button type=\"submit\" class=\"btn
btn-success mt-3\" name=\"update\"> Update </button>";

        } else {

```

```

        echo "<button type=\"submit\" class=\"btn
btn-success mt-3\" name=\"tambah\"> Tambah </button>";

    }

    ?>

</div>

</div>

</form>

```

penggunaan baris-baris tersebut merupakan bagian penting dari formulir yang bertujuan untuk menambah data peminjaman pada halaman web. Setiap baris kode memiliki fungsi spesifik dalam mengatur tampilan dan perilaku formulir tersebut. Pertama, elemen `<div class="container form-container">` digunakan untuk membuat sebuah kontainer yang mengatur tata letak dan tampilan formulir secara keseluruhan. Selanjutnya, `<div class="container">` digunakan sebagai kontainer tambahan untuk mengatur tampilan elemen-elemen dalam formulir, sedangkan `<div class="row">` adalah elemen untuk menyusun elemen-elemen secara horizontal dalam grid sistem Bootstrap. Formulir itu sendiri didefinisikan menggunakan `<form>` dengan metode POST untuk mengirim data ke halaman yang sama. Judul formulir, "Tambah Data Peminjaman", ditampilkan di tengah halaman dengan `<h1>` dan diberi gaya untuk perataan teks. Setiap input tanggal untuk "Tanggal Peminjaman" dan "Tanggal Kembali" disertai dengan label yang jelas menggunakan `<label>`, serta atribut-atribut yang mengatur tampilan dan validasi input. Selain itu, terdapat dropdown untuk memilih judul buku dan nama member dengan menggunakan elemen `<select>`, yang diisi dengan opsi-opsi yang diperoleh dari array `$dataBuku` dan `$dataMember` melalui PHP. Lalu, melalui PHP, ditentukan jenis tombol yang akan ditampilkan berdasarkan keberadaan parameter GET `"id_peminjaman"`. Jika parameter tersebut ada, tombol yang ditampilkan adalah tombol "Update" untuk mengedit data peminjaman yang sudah ada. Jika tidak, tombol yang ditampilkan adalah tombol "Tambah" untuk menambahkan data peminjaman baru.

Pada baris [189], syntax `</body>` merupakan tag penutup .

pada baris [190], syntax `</html>` merupakan tag penutup.

Z. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/syahputerii/Praktikum-Pemrograman-WEB-II/tree/main/PRAK501/Modul5>