

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 5**



FUNCTION DAN DATABASE

Oleh:

Aisya Anindia Septha NIM. 2210817220007

**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 : Aisya Anindia Septha
NIM : 2210817220007

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.....	4
DAFTAR TABEL	5
SOAL 1.....	6
A. Source Code	7
B. Output Program.....	45
C. Pembahasan	48
Tautan Git	58

DAFTAR GAMBAR

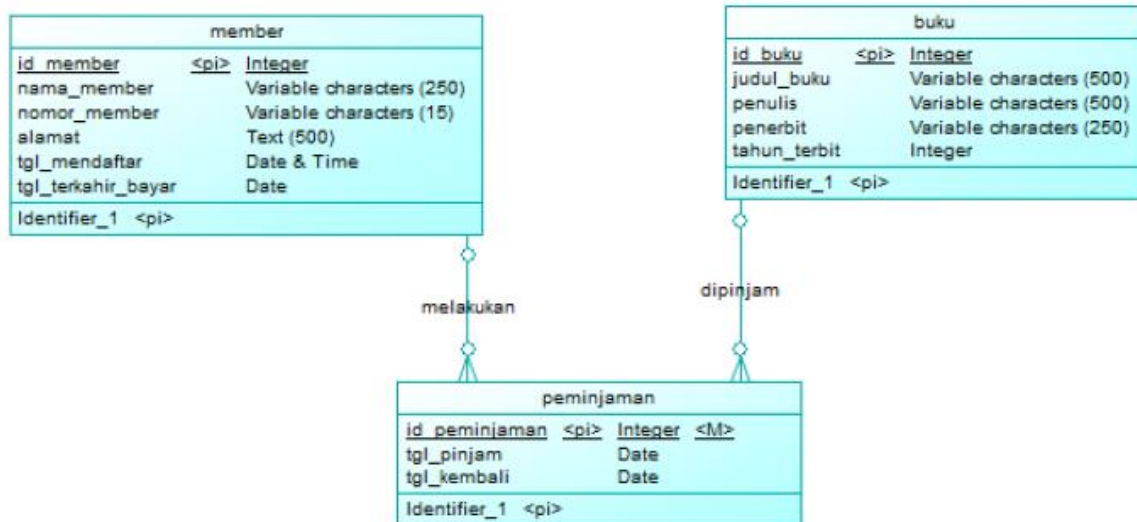
Gambar 1. Screenshot Hasil Jawaban Soal 1 Member	45
Gambar 2. Screenshot Hasil Jawaban Soal 1 FormMember	45
Gambar 3. Screenshot Hasil Jawaban Soal 1 Buku.....	46
Gambar 4. Screenshot Hasil Jawaban Soal 1 Form Buku	46
Gambar 5. Screenshot Hasil Jawaban Soal 1 Peminjaman	47
Gambar 6. Screenshot Hasil Jawaban Soal 1 FormPeminjaman.....	47

DAFTAR TABEL

Table 1. Source Code Soal1	7
----------------------------------	---

SOAL 1

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:

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 masingmasing dan disimpan di dalam satu file khusus yaitu Model.php
3. Minimal terdapat file-file berikut di dalam proyek

Nama File	Keterangan
Koneksi.php	Berisi fungsi-fungsi untuk melakukan koneksi ke basis data
Model.php	Berisi fungsi-fungsi untuk melakukan operasi data seperti insert, update, delete, get data untuk semua tabel
Member.php	Menampilkan semua data member dalam bentuk tabel dan terdapat tombol untuk

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

Simpan dengan nama projek di Github: PRAK501

A. Source Code

Table 1. Source Code Soal1

1	<?php
2	try {
3	\$pdo = new PDO('mysql:host=localhost;dbname=perpustakaan_nindy', 'asynnd', 'lsampai8');
4	\$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
5	} catch (PDOException \$e) {
6	echo 'Connection failed: ' . \$e->getMessage();
7	}
8	?>

```
1 <?php
2 class Model {
3     private $pdo;
4     public function __construct($pdo) {
5         $this->pdo = $pdo;
6     }
7     public function insertMember($data) {
8         try {
9             $columns = implode(', ', array_keys($data));
10            $placeholders = ':' . implode(':', array_keys($data));
11            $stmt = $this->pdo->prepare("INSERT INTO
member ($columns) VALUES ($placeholders)");
12            foreach ($data as $key => $value) {
13                $stmt->bindValue(":$key", $value);
14            }
15            if ($stmt->execute()) {
16                return true;
17            } else {
18                return false;
19            }
20        } catch (PDOException $e) {
21            echo "Error: " . $e->getMessage();
22            return false;
23        }
24    }
25    public function updateMember($data) {
26        try {
27            $set = '';
28            foreach ($data as $key => $value) {
29                $set .= "$key=$key, ";
```



```

30         }
31         $set = rtrim($set, ', ');
32         $stmt = $this->pdo->prepare("UPDATE member
SET $set WHERE id_member = :id_member");
33         foreach ($data as $key => $value) {
34             $stmt->bindValue(":$key", $value);
35         }
36         if ($stmt->execute()) {
37             return true;
38         } else {
39             return false;
40         }
41     } catch (PDOException $e) {
42         echo "Error: " . $e->getMessage();
43         return false;
44     }
45 }
46 public function readMemberById($id) {
47     try {
48         $stmt = $this->pdo->prepare("SELECT * FROM
member WHERE id_member = ?");
49         $stmt->bindValue(1, $id);
50         $stmt->execute();
51         return $stmt->fetch(PDO::FETCH_ASSOC);
52     } catch (PDOException $e) {
53         echo "Error: " . $e->getMessage();
54         return false;
55     }
56 }
57 public function readAllMembers() {
58     try {

```

59	<code>\$stmt = \$this->pdo->query("SELECT * FROM member");</code>
60	<code>return \$stmt->fetchAll(PDO::FETCH_ASSOC);</code>
61	<code>} catch (PDOException \$e) {</code>
62	<code>echo "Error: " . \$e->getMessage();</code>
63	<code>return false;</code>
64	<code>}</code>
65	<code>}</code>
66	<code>public function deleteMember(\$id) {</code>
67	<code>try {</code>
68	<code>\$stmt = \$this->pdo->prepare("DELETE FROM member WHERE id_member = ?");</code>
69	<code>\$stmt->bindValue(1, \$id);</code>
70	<code>return \$stmt->execute();</code>
71	<code>} catch (PDOException \$e) {</code>
72	<code>echo "Error: " . \$e->getMessage();</code>
73	<code>return false;</code>
74	<code>}</code>
75	<code>}</code>
76	<code>public function readAllData(\$table) {</code>
77	<code>try {</code>
78	<code>\$stmt = \$this->pdo->query("SELECT * FROM \$table");</code>
79	<code>return \$stmt->fetchAll(PDO::FETCH_ASSOC);</code>
80	<code>} catch (PDOException \$e) {</code>
81	<code>echo "Error: " . \$e->getMessage();</code>
82	<code>return false;</code>
83	<code>}</code>
84	<code>}</code>
85	<code>public function deleteDataByStruktur(\$table, \$id, \$column) {</code>

```

86         try {
87             $stmt = $this->pdo->prepare("DELETE FROM
$stable WHERE $column = ?");
88             $stmt->bindValue(1, $id);
89             return $stmt->execute();
90         } catch (PDOException $e) {
91             echo "Error: " . $e->getMessage();
92             return false;
93         }
94     }
95     public function insertBuku($data) {
96         try {
97             $columns = implode(', ', array_keys($data));
98             $placeholders = ':' . implode(', ',
array_keys($data));
99             $stmt = $this->pdo->prepare("INSERT INTO
buku ($columns) VALUES ($placeholders)");
100             foreach ($data as $key => $value) {
101                 $stmt->bindValue(":$key", $value);
102             }
103             if ($stmt->execute()) {
104                 return true;
105             } else {
106                 return false;
107             }
108         } catch (PDOException $e) {
109             echo "Error: " . $e->getMessage();
110             return false;
111         }
112     }
113     public function readDataByStruktur($id) {

```

```

114         try {
115             $stmt = $this->pdo->prepare("SELECT * FROM
buku WHERE id_buku = ?");
116             $stmt->bindValue(1, $id);
117             $stmt->execute();
118             return $stmt->fetch(PDO::FETCH_ASSOC);
119         } catch (PDOException $e) {
120             echo "Error: " . $e->getMessage();
121             return false;
122         }
123     }
124     public function updateData($table, $primaryKey, $id,
$data) {
125         try {
126             $set = '';
127             foreach ($data as $key => $value) {
128                 if ($key !== $primaryKey) {
129                     $set .= "$key=:key, ";
130                 }
131             }
132             $set = rtrim($set, ', ');
133             $stmt = $this->pdo->prepare("UPDATE $table
SET $set WHERE $primaryKey = :id");
134             $stmt->bindValue(':id', $id);
135             foreach ($data as $key => $value) {
136                 if ($key !== $primaryKey) {
137                     $stmt->bindValue(":$key", $value);
138                 }
139             }
140             return $stmt->execute();
141         } catch (PDOException $e) {

```

```

142         echo "Error: " . $e->getMessage();
143         return false;
144     }
145 }
146 public function selectDataById($table, $id,
147     $primaryKey) {
148     try {
149         $stmt = $this->pdo->prepare("SELECT * FROM
150     $table WHERE $primaryKey = ?");
151         $stmt->bindValue(1, $id);
152         $stmt->execute();
153         return $stmt->fetch(PDO::FETCH_ASSOC);
154     } catch (PDOException $e) {
155         echo "Error: " . $e->getMessage();
156         return false;
157     }
158 }
159 public function insertPeminjaman($data) {
160     try {
161         $columns = implode(', ', array_keys($data));
162         $placeholders = ':' . implode(', ',
163     array_keys($data));
164         $stmt = $this->pdo->prepare("INSERT INTO
165     peminjaman ($columns) VALUES ($placeholders)");
166         foreach ($data as $key => $value) {
167             $stmt->bindValue(":$key", $value);
168         }
169         if ($stmt->execute()) {
170             return true;
171         } else {
172             return false;
173         }
174     }
175 }

```

169	}
170	} catch (PDOException \$e) {
171	echo "Error: " . \$e->getMessage();
172	return false;
173	}
174	}
175	}
176	?>

1	<?php
2	require "Koneksi.php";
3	require "Model.php";
4	
5	\$model = new Model(\$pdo);
6	if (isset(\$_POST['delete'])) {
7	\$id_member_to_delete = \$_POST['id_member'];
8	\$model->deleteMember(\$id_member_to_delete);
9	}
10	\$members = \$model->readAllMembers();
11	
12	?>
13	<!DOCTYPE html>
14	<html>
15	<head>
16	<title>Data Member</title>
17	<style>
18	body {
19	text-align: center;
20	background-color: #ffadad;
21	font-family: Arial, sans-serif;
22	margin: 0;

```
23         padding: 0;
24     }
25     .container {
26         width: 80%;
27         margin: auto;
28         padding-top: 20px;
29     }
30     table {
31         border-collapse: collapse;
32         width: 100%;
33         margin: 20px auto;
34         background-color: #fff;
35         border-radius: 10px;
36         overflow: hidden;
37         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
38     }
39     th, td {
40         border: 1px solid black;
41         padding: 8px;
42         text-align: left;
43     }
44     th {
45         background-color: #f2f2f2;
46     }
47     button {
48         background-color: #ff0000;
49         color: white;
50         border: none;
51         padding: 5px 10px;
52         cursor: pointer;
53         border-radius: 5px;
```

```
54         }
55         button:hover {
56             background-color: #ff0000;
57         }
58         .add-button {
59             display: block;
60             margin: 20px auto;
61             text-align: center;
62         }
63     </style>
64 </head>
65 <body>
66 <div class="container">
67     <h2>Data Member</h2>
68
69     <table>
70         <tr>
71             <th>ID</th>
72             <th>Nama</th>
73             <th>Nomor</th>
74             <th>Alamat</th>
75             <th>Tanggal Daftar</th>
76             <th>Tanggal Bayar Terakhir</th>
77             <th>Aksi</th>
78         </tr>
79         <?php foreach ($members as $member): ?>
80             <tr>
81                 <td><?php echo $member['id_member'];
?></td>
82                 <td><?php echo $member['nama_member'];
?></td>
```


83	<code><td><?php echo \$member['nomor_member']; ?></td></code>
84	<code><td><?php echo \$member['alamat']; ?></td></code>
85	<code><td><?php echo \$member['tanggal_mendaftar']; ?></td></code>
86	<code><td><?php echo \$member['tanggal_terakhir_bayar']; ?></td></code>
87	<code><td></code>
88	<code><form style="display: inline- block;" method="post" onsubmit="return confirm('Apakah Anda yakin ingin menghapus member ini?')"></code>
89	<code><input type="hidden" name="id_member" value="<?php echo \$member['id_member']; ?>"></code>
90	<code><button type="submit" name="delete">Hapus</button></code>
91	<code></form></code>
92	<code><a href="FormMember.php?mode=edit&id=<?php echo \$member['id_member']; ?>"><button>Edit</button></code>
93	<code></td></code>
94	<code></tr></code>
95	<code><?php endforeach; ?></code>
96	<code></table></code>
97	
98	<code><div class="add-button"></code>
99	<code><button>Tambah Member</button></code>
100	<code></div></code>

101	</div>
102	</body>
103	</html>

1	<?php
2	require "Koneksi.php";
3	require "Model.php";
4	
5	\$model = new Model(\$pdo);
6	\$id_member = isset(\$_GET['id']) ? \$_GET['id'] : '';
7	\$mode = isset(\$_GET['mode']) ? \$_GET['mode'] : '';
8	if (\$mode === 'edit' && \$id_member) {
9	\$member = \$model->readMemberById(\$id_member);
	} else {
10	\$member = array(
11	"id_member" => "",
12	"nama_member" => "",
13	"nomor_member" => "",
14	"alamat" => "",
15	"tanggal_mendaftar" => "",
16	"tanggal_terakhir_bayar" => ""
17);
18	}
19	if (\$_SERVER["REQUEST_METHOD"] == "POST") {
20	\$data = array(
21	"id_member" => \$_POST['id_member'],
22	"nama_member" => \$_POST['nama_member'],
23	"nomor_member" => \$_POST['nomor_member'],
24	"alamat" => \$_POST['alamat'],
25	"tanggal_mendaftar" => date ('Y-m-d H:i:s'),
26	

27	<code>"tanggal_terakhir_bayar"</code>	=>
	<code>\$_POST['tanggal_terakhir_bayar']</code>	
28	<code>);</code>	
29	<code>if (\$data['id_member']) {</code>	
30	<code>if (\$model->updateMember(\$data)) {</code>	
31	<code>header('location: Member.php');</code>	
32	<code>exit();</code>	
33	<code>} else {</code>	
34	<code>echo "Gagal mengupdate member.";</code>	
35	<code>}</code>	
36	<code>} else {</code>	
37	<code>if (\$model->insertMember(\$data)) {</code>	
38	<code>header('location: Member.php');</code>	
39	<code>exit();</code>	
40	<code>} else {</code>	
41	<code>echo "Gagal menambahkan member.";</code>	
42	<code>}</code>	
43	<code>}</code>	
44	<code>}</code>	
45	<code>?></code>	
46	<code><!DOCTYPE html></code>	
47	<code><html></code>	
48	<code><head></code>	
49	<code><title>Form Member</title></code>	
50	<code><style></code>	
51	<code>body {</code>	
52	<code>font-family: Arial, sans-serif;</code>	
53	<code>background-color: #e79cff;</code>	
54	<code>margin: 0;</code>	
55	<code>padding: 0;</code>	
56	<code>display: flex;</code>	

```
57         justify-content: center;
58         align-items: center;
59         height: 100vh;
60     }
61
62     .form-container {
63         width: 400px;
64         padding: 20px;
65         background-color: #fff;
66         border-radius: 10px;
67         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
68     }
69
70     .form-header {
71         text-align: center;
72         margin-bottom: 20px;
73     }
74
75     form div {
76         margin-bottom: 15px;
77     }
78
79     label {
80         display: block;
81         margin-bottom: 5px;
82     }
83
84     input[type="text"],
85     input[type="date"] {
86         width: calc(100% - 12px);
87         padding: 10px;
```

88	border: 1px solid #ccc;
89	border-radius: 5px;
90	box-sizing: border-box;
91	}
92	
93	button[type="submit"] {
94	padding: 10px 20px;
95	border: none;
96	border-radius: 5px;
97	background-color: #bf00ff;
98	color: #fff;
99	font-size: 16px;
100	cursor: pointer;
101	}
102	
103	button[type="submit"]:hover {
104	background-color: #bf00ff;
105	}
106	</style>
107	</head>
108	<body>
109	<div class="form-container">
110	<h2 class="form-header"><?php echo (\$id_member && \$mode === 'edit') ? 'Edit' : 'Tambah'; ?> Member</h2>
111	<form method="post">
112	<input type="hidden" name="id_member" value="<?php echo \$member['id_member']; ?>">
113	<div>
114	<label for="nama_member">Nama Member:</label>

115	<code><input type="text" id="nama_member"</code>
	<code>name="nama_member" value="<?php echo</code>
	<code>\$member['nama_member']; ?>"></code>
116	<code></div></code>
117	<code><div></code>
118	<code><label for="nomor_member">Nomor</code>
	<code>Member:</label></code>
119	<code><input type="text" id="nomor_member"</code>
	<code>name="nomor_member" value="<?php echo</code>
	<code>\$member['nomor_member']; ?>"></code>
120	<code></div></code>
121	<code><div></code>
122	<code><label for="alamat">Alamat:</label></code>
123	<code><input type="text" id="alamat" name="alamat"</code>
	<code>value="<?php echo \$member['alamat']; ?>"></code>
124	<code></div></code>
125	<code><div></code>
126	<code><label for="tanggal_mendaftar">Tanggal</code>
	<code>Mendaftar:</label></code>
127	<code><input type="date" id="tanggal_mendaftar"</code>
	<code>name="tanggal_mendaftar" value="<?php echo</code>
	<code>\$member['tanggal_mendaftar']; ?>"></code>
128	<code></div></code>
129	<code><div></code>
130	<code><label for="tanggal_terakhir_bayar">Tanggal</code>
	<code>Terakhir Bayar:</label></code>
131	<code><input type="date"</code>
	<code>id="tanggal_terakhir_bayar"</code>
	<code>name="tanggal_terakhir_bayar" value="<?php echo</code>
	<code>\$member['tanggal_terakhir_bayar']; ?>"></code>
132	<code></div></code>

133	<code><button</code>	<code>type="submit"</code>
	<code>name="submit">Simpan</button></code>	
134	<code></form></code>	
135	<code></div></code>	
136	<code></body></code>	
137	<code></html></code>	

1	<code><?php</code>	
2	<code>require 'Koneksi.php';</code>	
3	<code>require 'Model.php';</code>	
4		
5	<code>\$model = new Model(\$pdo);</code>	
6	<code>if (isset(\$_GET['action']) && \$_GET['action'] ===</code>	
	<code>'delete' && !empty(\$_GET['id_buku'])) {</code>	
7	<code> \$id_buku = \$_GET['id_buku'];</code>	
8	<code> \$model->deleteDataByStruktur("buku", \$id_buku,</code>	
	<code>"id_buku");</code>	
9	<code> header('Location: Buku.php');</code>	
10	<code> exit();</code>	
11	<code>}</code>	
12	<code>if (isset(\$_POST['submit']) && empty(\$_POST['id_buku']))</code>	
	<code>{</code>	
13	<code> \$data = array(</code>	
14	<code> 'judul_buku' => \$_POST['judul'],</code>	
15	<code> 'penulis' => \$_POST['pengarang'],</code>	
16	<code> 'penerbit' => \$_POST['penerbit'],</code>	
17	<code> 'tahun_terbit' => \$_POST['tahun_terbit']</code>	
18	<code>);</code>	
19		
20	<code> if (\$model->insertBuku(\$data)) {</code>	
21	<code> echo "Data buku berhasil ditambahkan.";</code>	

```
22     } else {
23         echo "Error: Gagal menambahkan data buku.";
24     }
25 }
26 if (isset($_POST['update'])) {
27     $id_buku = $_POST['id_buku'];
28     $data = [
29         'judul_buku' => $_POST['judul_buku'],
30         'penulis' => $_POST['penulis'],
31         'penerbit' => $_POST['penerbit'],
32         'tahun_terbit' => $_POST['tahun_terbit']
33     ];
34     $model->updateData('buku', 'id_buku', $id_buku,
35 $data);
36     header('Location: Buku.php');
37 }
38 $bukuToEdit = null;
39 if (isset($_GET['id_buku'])) {
40     $id_buku = $_GET['id_buku'];
41     $bukuToEdit = $model->readDataByStruktur($id_buku);
42 }
43 ?>
44 <!DOCTYPE html>
45 <html lang="en">
46 <head>
47     <title>Buku Perpus</title>
48     <style>
49         body {
50             font-family: Arial, sans-serif;
51             background-color: #e79cff;
52             margin: 0;
```



```
52         padding: 0;
53     }
54     .container {
55         max-width: 800px;
56         margin: 20px auto;
57         padding: 20px;
58         background-color: #fff;
59         border-radius: 5px;
60         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
61     }
62     h1 {
63         text-align: center;
64     }
65     table {
66         width: 100%;
67         border-collapse: collapse;
68     }
69     table th, table td {
70         border: 1px solid #ddd;
71         padding: 8px;
72         text-align: left;
73     }
74     table th {
75         background-color: #f2f2f2;
76     }
77     .btn {
78         display: inline-block;
79         padding: 8px 16px;
80         background-color: #6b47a2;
81         color: #fff;
82         text-decoration: none;
```

```
83         border: none;
84         border-radius: 4px;
85         cursor: pointer;
86     }
87     .btn-danger {
88         background-color: #f44336;
89     }
90     .btn:hover {
91         background-color: #6b47a2;
92     }
93     .btn-danger:hover {
94         background-color: #d32f2f;
95     }
96     .form-container {
97         max-width: 500px;
98         margin: 0 auto;
99     }
100    .form-container form {
101        background-color: #f2f2f2;
102        padding: 20px;
103        border-radius: 5px;
104    }
105    .form-container label, .form-container
input[type="text"] {
106        display: block;
107        margin-bottom: 10px;
108    }
109    .form-container input[type="text"] {
110        width: 100%;
111        padding: 8px;
112        border: 1px solid #ccc;
```

```
113         border-radius: 4px;
114     }
115     .form-container input[type="submit"] {
116         padding: 10px 16px;
117         background-color: #6b47a2;
118         color: #6b47a2;
119         border: none;
120         border-radius: 4px;
121         cursor: pointer;
122     }
123     .form-container input[type="submit"]:hover {
124         background-color: #6b47a2;
125     }
126 </style>
127 </head>
128 <body>
129 <div class="container">
130     <h1>Buku</h1>
131     <?php if ($bukuToEdit): ?>
132         <div class="form-container">
133             <h2>Edit Buku</h2>
134             <form method="post" action="Buku.php">
135                 <input type="hidden" name="id_buku"
136 value="<?php                                     echo
137 htmlspecialchars($bukuToEdit['id_buku']); ?>">
138                 <label for="judul">Judul Buku:</label>
139                 <input type="text" id="judul"
140 name="judul_buku" value="<?php                                     echo
141 htmlspecialchars($bukuToEdit['judul_buku']); ?>">
```

```

139         <label for="pengarang">Pengarang
Buku:</label>
140         <input type="text" id="pengarang"
name="penulis" value="<?php echo
htmlspecialchars($bukuToEdit['penulis']); ?>">
141
142         <label for="penerbit">Penerbit
Buku:</label>
143         <input type="text" id="penerbit"
name="penerbit" value="<?php echo
htmlspecialchars($bukuToEdit['penerbit']); ?>">
144
145         <label for="tahun_terbit">Tahun Terbit
Buku:</label>
146         <input type="text" id="tahun_terbit"
name="tahun_terbit" value="<?php echo
htmlspecialchars($bukuToEdit['tahun_terbit']); ?>">
147
148         <input type="submit" name="update"
value="Update" class="btn">
149     </form>
150 </div>
151 <?php else: ?>
152     <table>
153         <thead>
154         <tr>
155             <th>ID Buku</th>
156             <th>Judul Buku</th>
157             <th>Penulis</th>
158             <th>Penerbit</th>
159             <th>Tahun Terbit</th>

```

160	<code><th>Aksi</th></code>
161	<code></tr></code>
162	<code></thead></code>
163	<code><tbody></code>
164	<code><?php</code>
165	<code>\$result = \$model->readAllData('buku');</code>
166	<code>if (\$result !== false) {</code>
167	<code> foreach (\$result as \$data) { ?></code>
168	<code> <tr></code>
169	<code> <td><?php echo</code>
	<code>htmlspecialchars(\$data['id_buku']); ?></td></code>
170	<code> <td><?php echo</code>
	<code>htmlspecialchars(\$data['judul_buku']); ?></td></code>
171	<code> <td><?php echo</code>
	<code>htmlspecialchars(\$data['penulis']); ?></td></code>
172	<code><td><?php echo htmlspecialchars(\$data['penerbit']);</code>
	<code>?></td></code>
173	<code><td><?php echo htmlspecialchars(\$data['tahun_terbit']);</code>
	<code>?></td></code>
174	<code><td class="actions"></code>
175	<code> <a href="Buku.php?id_buku=<?php echo</code>
	<code>htmlspecialchars(\$data['id_buku']); ?>" class="btn btn-</code>
	<code>primary">Edit</code>
176	<code> <a href="Buku.php?action=delete&id_buku=<?php echo</code>
	<code>htmlspecialchars(\$data['id_buku']); ?>" onclick="return</code>
	<code>confirm('Hapus data?')" class="btn btn-danger">Hapus</code>
177	<code></td></code>
178	<code></tr></code>
179	<code><?php }</code>
180	<code> } else {</code>

```

181             echo    '<tr><td    colspan="6">No    data
found</td></tr>';
182         }
183     ?>
184     </tbody>
185 </table>
186 <br>
187 <div style="text-align: center;">
188     <a    href="FormBuku.php"    class="btn    btn-
success">Tambah Buku</a>
189 </div>
190 <?php endif; ?>
191 </div>
192 </body>
193 </html>

```

```

1 <?php
2 require 'Koneksi.php';
3 require 'Model.php';
4
5 $model = new Model($pdo);
6 if (isset($_POST['submit'])) {
7     $data = array(
8         'judul_buku' => $_POST['judul'],
9         'penulis' => $_POST['pengarang'],
10        'penerbit' => $_POST['penerbit'],
11        'tahun_terbit' => $_POST['tahun_terbit']
12    );
13    if      (empty($data['judul_buku'])           ||
empty($data['penulis']) || empty($data['penerbit']) ||
empty($data['tahun_terbit'])) {

```

```
14         echo "Error: Semua bidang harus diisi.";
15     } else {
16         if ($model->insertBuku($data)) {
17             echo "Data buku berhasil ditambahkan.";
18             header('Location: Buku.php');
19             exit();
20         } else {
21             echo "Error: Gagal menambahkan data buku.";
22         }
23     }
24 }
25 ?>
26
27 <!DOCTYPE html>
28 <html>
29 <head>
30     <title>Form Buku</title>
31     <style>
32         body {
33             font-family: Arial, sans-serif;
34             background-color: #e79cff;
35         }
36         .form-container {
37             max-width: 400px;
38             margin: auto;
39             padding: 20px;
40             background-color: #fff;
41             border: 1px solid #ccc;
42             border-radius: 10px;
43             box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
44             margin-top: 50px;
```

```
45     }
46     .form-container h2 {
47         text-align: center;
48         color: #333;
49     }
50     .form-container label {
51         display: block;
52         margin-bottom: 8px;
53         color: #333;
54     }
55     .form-container input[type="text"] {
56         width: 100%;
57         padding: 8px;
58         margin-bottom: 10px;
59         border: 1px solid #ccc;
60         border-radius: 5px;
61         box-sizing: border-box;
62     }
63     .form-container input[type="submit"] {
64         width: 100%;
65         padding: 10px;
66         background-color: #8359a3;
67         color: white;
68         border: none;
69         border-radius: 5px;
70         cursor: pointer;
71         font-size: 16px;
72     }
73     .form-container input[type="submit"]:hover {
74         background-color: #6b47a2;
75     }
```


76	</style>
77	</head>
78	<body>
79	<div class="form-container">
80	<h2>Formulir Data Buku</h2>
81	<form method="post" action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]); ?>">
82	<label for="judul">Judul Buku:</label>
83	<input type="text" id="judul" name="judul">
84	
85	<label for="pengarang">Pengarang Buku:</label>
86	<input type="text" id="pengarang" name="pengarang">
87	
88	<label for="penerbit">Penerbit Buku:</label>
89	<input type="text" id="penerbit" name="penerbit">
90	
91	<label for="tahun_terbit">Tahun Terbit Buku:</label>
92	<input type="text" id="tahun_terbit" name="tahun_terbit">
93	
94	<input type="submit" name="submit" value="Tambah">
95	</form>
96	</div>
97	</body>
98	</html>

1	<?php
2	require "Koneksi.php";

```

3  require "Model.php";
4
5  $model = new Model($pdo);
6  if (!empty($_GET['id_peminjaman'])) {
7      $id_peminjaman = $_GET['id_peminjaman'];
8      $model->deleteDataByStruktur("peminjaman",
9      $id_peminjaman, "id_peminjaman");
10     header('location:Peminjaman.php');
11     exit();
12 }
13 if (isset($_POST['submit']) &&
14 empty($_POST['id_peminjaman'])) {
15     $data = array(
16         'tanggal_pinjam' => $_POST['tanggal_pinjam'],
17         'tanggal_kembali' => $_POST['tanggal_kembali'],
18     );
19     if ($model->insertPeminjaman($data)) {
20         echo "Data Peminjaman berhasil ditambahkan.";
21     } else {
22         echo "Error:      Gagal      menambahkan      data
23 peminjaman.";
24     }
25 }
26 ?>
27
28 <!DOCTYPE html>
29 <html>
30 <head>
31     <title>Data Peminjaman</title>
32     <style>
33         body {

```

```
31         font-family: Arial, sans-serif;
32         background-color: #87cefa;
33         margin: 0;
34         padding: 0;
35     }
36     .container {
37         width: 80%;
38         margin: auto;
39         overflow: hidden;
40         background: #fff;
41         padding: 20px;
42         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
43         margin-top: 50px;
44     }
45     h2 {
46         text-align: center;
47         color: #333;
48     }
49     table {
50         width: 100%;
51         border-collapse: collapse;
52         margin: 20px 0;
53     }
54     table, th, td {
55         border: 1px solid #ddd;
56     }
57     th, td {
58         padding: 12px;
59         text-align: center;
60     }
61     th {
```

62	background-color: #f2f2f2;
63	color: #333;
64	}
65	tr:nth-child(even) {
66	background-color: #f9f9f9;
67	}
68	.btn {
69	background-color: #1692c6;
70	color: white;
71	border: none;
72	padding: 10px 20px;
73	cursor: pointer;
74	text-decoration: none;
75	margin: 5px;
76	border-radius: 5px;
77	}
78	.btn:hover {
79	background-color: #1692c6;
80	}
81	.btn-danger {
82	background-color: #f44336;
83	}
84	.btn-danger:hover {
85	background-color: #e53935;
86	}
87	.btn-secondary {
88	background-color: #1692c6;
89	}
90	.btn-secondary:hover {
91	background-color: #1692c6;
92	}

93	.action {
94	display: flex;
95	justify-content: center;
96	}
97	</style>
98	</head>
99	<body>
100	<div class="container">
101	<h2>Peminjaman</h2>
102	<table>
103	<thead>
104	<tr>
105	<th>ID Peminjaman</th>
106	<th>Tanggal Peminjaman</th>
107	<th>Tanggal Kembali</th>
108	<th colspan="2">Aksi</th>
109	</tr>
110	</thead>
111	<tbody>
112	<?php
113	\$result = \$model->readAllData("peminjaman");
114	if (\$result !== false && !empty(\$result)) {
115	foreach (\$result as \$data) { ?>
116	<tr>
117	<td><?php echo \$data['id_peminjaman'] ?></td>
118	<td><?php echo \$data['tanggal_pinjam'] ?></td>
119	<td><?php echo \$data['tanggal_kembali'] ?></td>

120	<code><td class="action"></code>
121	<code><a</code> <code>href="FormPeminjaman.php?id_peminjaman=<?php echo</code> <code>\$data['id_peminjaman']; ?>"><button</code> <code>class="btn">Edit</button></code></code>
122	<code><a</code> <code>href="Peminjaman.php?id_peminjaman=<?php echo</code> <code>\$data['id_peminjaman']; ?>" onclick="return</code> <code>confirm('Hapus data?')"><button class="btn btn-</code> <code>danger">Hapus</button></code></code>
123	<code></td></code>
124	<code></tr></code>
125	<code><?php }</code>
126	<code>} else {</code>
127	<code>echo '<tr><td colspan="6">Tidak ada data</code> <code>peminjaman</td></tr>';</code>
128	<code>}</code>
129	<code>?></code>
130	<code></tbody></code>
131	<code></table></code>
132	<code><div style="text-align: center;"></code>
133	<code><button</code> <code>class="btn">Tambah Data Peminjaman</button></code>
134	<code></div></code>
135	<code></div></code>
136	<code></body></code>
137	<code></html></code>

1	<code><?php</code>
2	<code>require "koneksi.php";</code>
3	<code>require "model.php";</code>

```

4 $model = new Model($pdo);
5
6 if (isset($_POST['submit'])) {
7     $data = array(
8         "id_peminjaman" => $_POST['id_peminjaman'] .
9         "",
10        "tanggal_pinjam"      =>      date('Y-m-d',
11        strtotime($_POST['tanggal_pinjam'])) . "",
12        "tanggal_kembali"     =>      date('Y-m-d',
13        strtotime($_POST['tanggal_kembali'])) . "",
14    );
15    $model->insertPeminjaman($data, 'Peminjaman');
16    header('location:Peminjaman.php');
17 }
18
19 if (isset($_POST['edit'])) {
20     $id      =      isset($_GET['id_peminjaman'])      ?
21     $_GET['id_peminjaman'] : '';
22     $data = array(
23         "id_peminjaman" => $_POST['id_peminjaman'] .
24         "",
25        "tanggal_pinjam"      =>      date('Y-m-d',
26        strtotime($_POST['tanggal_pinjam'])) . "",
27        "tanggal_kembali"     =>      date('Y-m-d',
28        strtotime($_POST['tanggal_kembali'])) . "",
29    );
30     $model->updateData('Peminjaman', 'id_peminjaman',
31     $id, $data);
32     header("location:Peminjaman.php");
33 }
34

```

```
27 if (isset($_POST['kembali'])) {
28     header("location:Peminjaman.php");
29 }
30
31 $id      =      isset($_GET['id_peminjaman'])      ?
32 $_GET['id_peminjaman'] : '';
33
34 $data    =    $model->selectDataById('peminjaman',    $id,
35 "id_peminjaman");
36
37 ?>
38 <!doctype html>
39 <html lang="en">
40 <head>
41     <title>Form Peminjaman</title>
42     <meta charset="utf-8">
43     <meta name="viewport" content="width=device-width,
44 initial-scale=1, shrink-to-fit=no">
45     <link rel="stylesheet" href="style.css">
46     <style>
47         body {
48             font-family: Arial, sans-serif;
49             background-color: #FF6FFF;
50             margin: 0;
51             padding: 0;
52         }
53         .container {
54             width: 50%;
55             margin: auto;
56             overflow: hidden;
57             background: #fff;
58             padding: 20px;
```

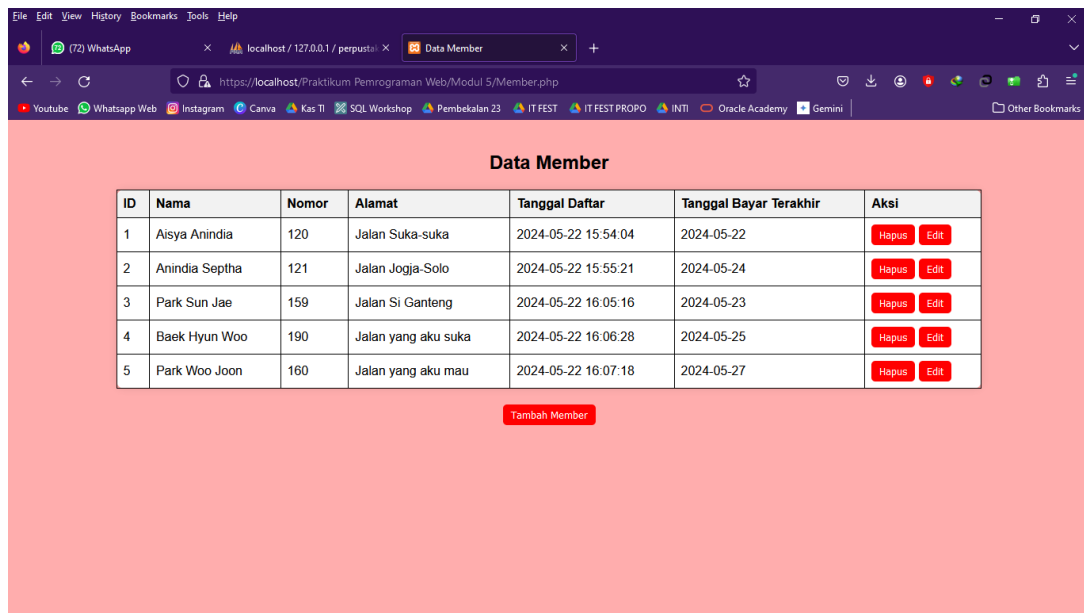

55	box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
56	margin-top: 50px;
57	}
58	h1 {
59	text-align: center;
60	color: #333;
61	}
62	form div {
63	margin-bottom: 15px;
64	}
65	label {
66	display: block;
67	margin-bottom: 5px;
68	color: #333;
69	}
70	input[type="text"],
71	input[type="date"] {
72	width: 100%;
73	padding: 10px;
74	margin: 5px 0 10px 0;
75	display: inline-block;
76	border: 1px solid #ccc;
77	border-radius: 4px;
78	box-sizing: border-box;
79	}
80	.button-container {
81	display: flex;
82	justify-content: center;
83	gap: 10px;
84	}
85	button {

86	width: 300px;
87	background-color: #E0115F;
88	color: white;
89	padding: 14px 20px;
90	margin: 8px 0;
91	border: none;
92	border-radius: 4px;
93	cursor: pointer;
94	text-align: center;
95	}
96	button:hover {
97	opacity: 0.8;
98	}
99	</style>
100	</head>
101	<body>
102	<div class="container">
103	<?php if (isset(\$_GET['id_peminjaman'])) : ?>
104	<h1>Edit Peminjaman</h1>
105	<form method="POST">
106	<div>
107	<label for="id_peminjaman">ID Peminjaman:</label>
108	<input type="text" id="id_peminjaman" name="id_peminjaman" value="<?php echo isset(\$data['id_peminjaman']) ? htmlspecialchars(\$data['id_peminjaman']) : ''; ?>" required>
109	</div>
110	<div>

111	<code><label for="tanggal_pinjam">Tanggal Pinjam:</label></code>
112	<code><input type="date" id="tanggal_pinjam" name="tanggal_pinjam" value="<?php echo isset(\$data['tanggal_pinjam']) ? htmlspecialchars(\$data['tanggal_pinjam']) : ''; ?>" required></code>
113	<code></div></code>
114	<code><div></code>
115	<code><label for="tanggal_kembali">Tanggal Kembali:</label></code>
116	<code><input type="date" id="tanggal_kembali" name="tanggal_kembali" value="<?php echo isset(\$data['tanggal_kembali']) ? htmlspecialchars(\$data['tanggal_kembali']) : ''; ?>" required></code>
117	<code></div></code>
118	<code><div class="button-container"></code>
119	<code><button type="submit" name="edit">Simpan Perubahan</button></code>
120	<code></div></code>
121	<code></form></code>
122	<code><?php else : ?></code>
123	<code><h1>Tambah Peminjaman</h1></code>
124	<code><form method="POST"></code>
125	<code><div></code>
126	<code><label for="id_peminjaman">ID Peminjaman:</label></code>
127	<code><input type="text" id="id_peminjaman" name="id_peminjaman"></code>
128	<code></div></code>

129	<div>
130	<label for="tanggal_pinjam">Tanggal
	Pinjam:</label>
131	<input type="date"
	id="tanggal_pinjam" name="tanggal_pinjam">
132	</div>
133	<div>
134	<label
	for="tanggal_kembali">Tanggal Kembali:</label>
135	<input type="date"
	id="tanggal_kembali" name="tanggal_kembali">
136	</div>
137	<div class="button-container">
138	<button type="submit"
	name="submit">Tambah Data</button>
139	</div>
140	</form>
141	<?php endif; ?>
142	</div>
143	</body>
144	</html>

B. Output Program

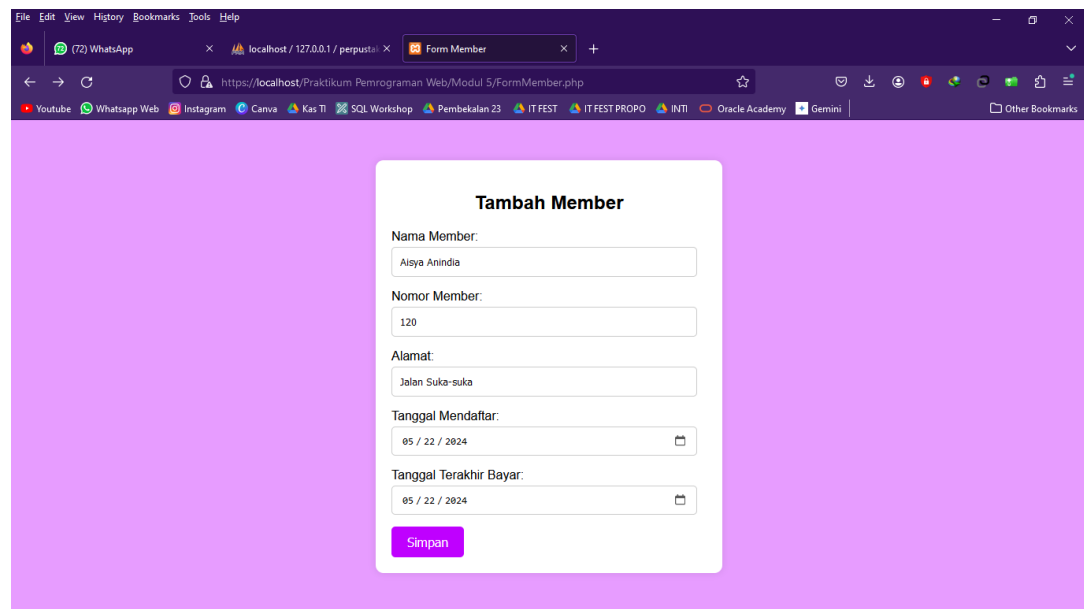


The screenshot shows a web browser window with the address bar displaying `https://localhost/Praktikum Pemrograman Web/Modul 5/Member.php`. The page has a light pink background and is titled "Data Member". It contains a table with 7 columns: ID, Nama, Nomor, Alamat, Tanggal Daftar, Tanggal Bayar Terakhir, and Aksi. There are 5 rows of data. Below the table is a red button labeled "Tambah Member".

ID	Nama	Nomor	Alamat	Tanggal Daftar	Tanggal Bayar Terakhir	Aksi
1	Aisya Anindia	120	Jalan Suka-suka	2024-05-22 15:54:04	2024-05-22	Hapus Edit
2	Anindia Septha	121	Jalan Jogja-Solo	2024-05-22 15:55:21	2024-05-24	Hapus Edit
3	Park Sun Jae	159	Jalan Si Canteng	2024-05-22 16:05:16	2024-05-23	Hapus Edit
4	Baek Hyun Woo	190	Jalan yang aku suka	2024-05-22 16:06:28	2024-05-25	Hapus Edit
5	Park Woo Joon	160	Jalan yang aku mau	2024-05-22 16:07:18	2024-05-27	Hapus Edit

[Tambah Member](#)

Gambar 1. Screenshot Hasil Jawaban Soal 1 Member



The screenshot shows a web browser window with the address bar displaying `https://localhost/Praktikum Pemrograman Web/Modul 5/FormMember.php`. The page has a light purple background and is titled "Tambah Member". It contains a form with the following fields: Nama Member (Aisya Anindia), Nomor Member (120), Alamat (Jalan Suka-suka), Tanggal Mendaftar (05 / 22 / 2024), and Tanggal Terakhir Bayar (05 / 22 / 2024). There is a purple "Simpan" button at the bottom.

Tambah Member

Nama Member:

Nomor Member:

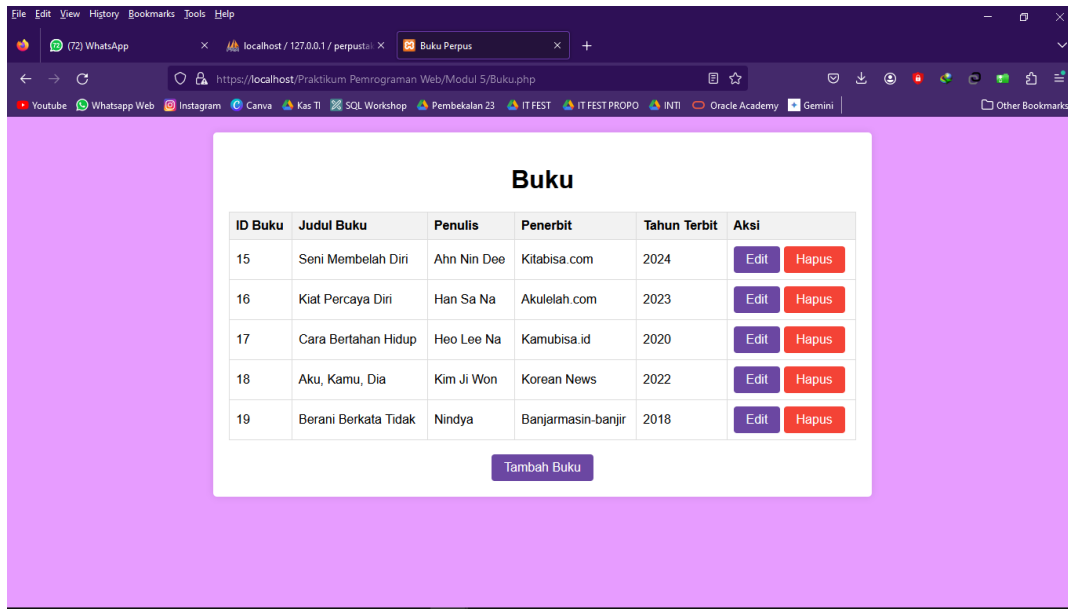
Alamat:

Tanggal Mendaftar:

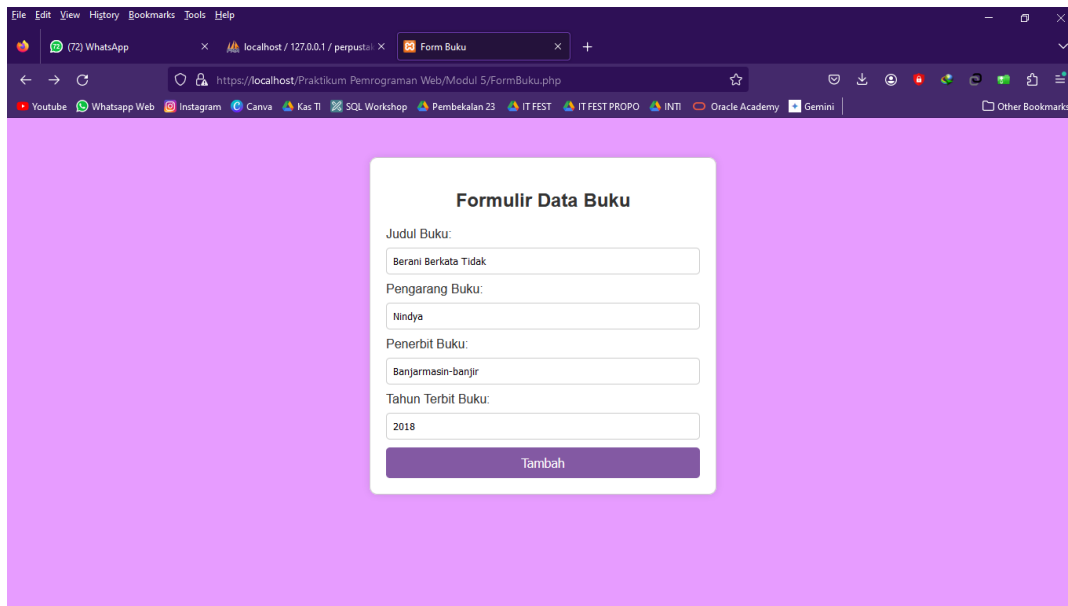
Tanggal Terakhir Bayar:

[Simpan](#)

Gambar 2. Screenshot Hasil Jawaban Soal 1 FormMember



Gambar 3. Screenshot Hasil Jawaban Soal 1 Buku



Gambar 4. Screenshot Hasil Jawaban Soal 1 Form Buku

Tambah Peminjaman

ID Peminjaman:

Tanggal Pinjam:

Tanggal Kembali:

[Tambah Data](#)

Gambar 5. Screenshot Hasil Jawaban Soal 1 Peminjaman

Peminjaman

ID Peminjaman	Tanggal Peminjaman	Tanggal Kembali	Aksi
8	2024-05-23	2024-05-23	Edit Hapus
12	2024-05-23	2024-05-24	Edit Hapus
13	2024-05-02	2024-05-20	Edit Hapus
90	2024-05-06	2024-05-20	Edit Hapus

[Tambah Data Peminjaman](#)

Gambar 6. Screenshot Hasil Jawaban Soal 1 FormPeminjaman

C. Pembahasan

Koneksi.php

1. `<?php` digunakan sebagai ag pembuka untuk memulai blok kode PHP.
2. ``require 'Koneksi.php'` digunakan untuk memanggil file Koneksi.php yang berisi koneksi ke database.
3. ``require 'Model.php';`` digunakan untuk memanggil file Model.php yang berisi definisi kelas Model.
4. ``$model = new Model($pdo);`` digunakan untuk Membuat instance dari kelas Model dengan parameter objek koneksi database ``$pdo``.
5. ``if (isset($_GET['action']) && $_GET['action'] === 'delete' && !empty($_GET['id_buku'])) {`` digunakan untuk Memeriksa apakah terdapat parameter ``action`` dalam query string dan nilainya adalah ``delete``, serta apakah parameter ``id_buku`` tidak kosong.
6. ``$id_buku = $_GET['id_buku'];`` digunakan untuk Mengambil nilai dari parameter ``id_buku`` dari query string.
7. ``$model->deleteDataByStruktur("buku", $id_buku, "id_buku");``: Memanggil metode ``deleteDataByStruktur`` dari objek ``$model`` untuk menghapus data buku dengan id yang sesuai.
8. ``header('Location: Buku.php');`` digunakan untuk Mengalihkan pengguna ke halaman ``Buku.php`` setelah penghapusan data.
9. ``exit();``: Menghentikan eksekusi skrip agar tidak melanjutkan ke bawahnya.
10. ``if (isset($_POST['submit']) && empty($_POST['id_buku'])) {``: Memeriksa apakah tombol submit ditekan dan apakah nilai dari ``id_buku`` dalam ``$_POST`` kosong, menandakan bahwa ini adalah permintaan untuk menambahkan data baru.
11. ``$data = array(...);``: Membuat array ``$data`` yang berisi informasi buku yang akan ditambahkan.
12. ``if ($model->insertBuku($data)) { ... }``: Memanggil metode ``insertBuku`` dari objek ``$model`` untuk menambahkan data buku. Jika berhasil, akan menampilkan pesan sukses, jika gagal, akan menampilkan pesan error.
13. ``if (isset($_POST['update'])) {``: Memeriksa apakah tombol "Update" ditekan.

14. ``$id_buku = $_POST['id_buku'];``: Mengambil nilai ``id_buku`` dari data yang akan diperbarui.
15. ``$data = [...];``: Membuat array ``$data`` yang berisi informasi buku yang akan diperbarui.
16. ``$model->updateData('buku', 'id_buku', $id_buku, $data);``: Memanggil metode ``updateData`` dari objek ``$model`` untuk memperbarui data buku yang sesuai.
17. ``header('Location: Buku.php');``: Mengalihkan pengguna kembali ke halaman ``Buku.php`` setelah proses pembaruan selesai.
18. ``$bukuToEdit = null;``: Inisialisasi variabel ``$bukuToEdit`` dengan nilai ``null``.
19. ``if (isset($_GET['id_buku'])) {``: Memeriksa apakah ada parameter ``id_buku`` dalam query string
20. ``$id_buku = $_GET['id_buku'];``: Mengambil nilai dari parameter ``id_buku`` dari query string.
21. ``$bukuToEdit = $model->readDataByStruktur($id_buku);``: Memanggil metode ``readDataByStruktur`` dari objek ``$model`` untuk membaca data buku berdasarkan id.
22. ``<!DOCTYPE html> ... </html>``: Bagian HTML dari halaman web, yang akan dirender oleh browser.
23. ``<?php if ($bukuToEdit): ?>``: Memulai blok PHP untuk menampilkan form edit buku jika ``$bukuToEdit`` tidak null.
24. ``<?php else: ?>``: Memulai blok PHP untuk menampilkan tabel data buku jika ``$bukuToEdit`` null.
25. ``<?php endforeach; ?>``: Akhir dari loop foreach untuk menampilkan data buku dalam tabel.
26. ``</body>`` dan ``</html>``: Penutup dari tag HTML untuk menutup halaman HTML.

Model.php

1. ``__construct($pdo)``: Konstruktor kelas yang menerima parameter objek PDO yang digunakan untuk koneksi ke basis data. Objek PDO disimpan ke dalam properti ``$pdo`` untuk digunakan di seluruh kelas

2. ``insertMember($data)``: Metode untuk menambahkan data anggota ke dalam tabel 'member'. Metode ini menggunakan prepared statement untuk mencegah serangan SQL Injection.
3. ``updateMember($data)``: Metode untuk memperbarui data anggota dalam tabel 'member' berdasarkan ID anggota. Juga menggunakan prepared statement untuk keamanan.
4. ``readMemberById($id)``: Metode untuk membaca data anggota berdasarkan ID anggota.
5. ``readAllMembers()``: Metode untuk membaca semua data anggota dari tabel 'member'.
6. ``deleteMember($id)``: Metode untuk menghapus data anggota berdasarkan ID anggota.
7. ``readAllData($table)``: Metode untuk membaca semua data dari tabel tertentu.
8. ``deleteDataByStruktur($table, $id, $column)``: Metode untuk menghapus data dari tabel tertentu berdasarkan nilai kolom tertentu.
9. ``insertBuku($data)``: Metode untuk menambahkan data buku ke dalam tabel 'buku'.
10. ``readDataByStruktur($id)``: Metode untuk membaca data buku berdasarkan ID buku.
11. ``updateData($table, $primaryKey, $id, $data)``: Metode untuk memperbarui data dalam tabel tertentu berdasarkan nilai ID tertentu.
12. ``selectDataById($table, $id, $primaryKey)``: Metode untuk memilih data dari tabel tertentu berdasarkan nilai ID tertentu.
13. ``insertPeminjaman($data)``: Metode untuk menambahkan data peminjaman ke dalam tabel 'peminjaman'.

Member.php

1. Baris 1-14: Memuat file ``Koneksi.php`` dan ``Model.php``, serta membuat instance objek dari kelas ``Model`` dengan parameter objek PDO ``$pdo`` yang sudah terkoneksi ke basis data. Juga, terdapat kondisi untuk menghapus anggota jika tombol "Hapus" ditekan dan melakukan pemanggilan fungsi ``readAllMembers()`` untuk mendapatkan seluruh data anggota dari basis data
2. Baris 19-62: Bagian HTML dari halaman web yang akan dirender oleh browser. Terdapat sebuah tabel yang menampilkan data anggota dengan kolom-kolom seperti ID,

nama, nomor, alamat, tanggal mendaftar, dan tanggal bayar terakhir. Setiap baris tabel memiliki tombol "Hapus" yang akan menampilkan konfirmasi sebelum menghapus anggota, serta tombol "Edit" yang akan mengarahkan ke halaman `FormMember.php` untuk mengedit anggota.

3. Dalam loop `foreach`, data anggota yang telah diambil dari basis data ditampilkan di setiap baris tabel.

4. Ketika tombol "Hapus" pada suatu baris tabel ditekan, sebuah form disubmit dengan metode POST. Form ini menyertakan input hidden yang menyimpan ID anggota yang akan dihapus. Sebuah konfirmasi akan ditampilkan sebelum proses penghapusan dilakukan.

5. Tombol "Tambah Member" di bagian bawah halaman akan mengarahkan pengguna ke halaman `FormMember.php` dengan mode tambah data anggota.

6. CSS yang disertakan digunakan untuk mengatur tata letak dan tampilan halaman, seperti pengaturan warna latar belakang, pengaturan lebar, dan pengaturan tampilan tombol.

7. Halaman ini memiliki fungsi utama untuk menampilkan data anggota, memungkinkan pengguna untuk menghapus anggota, dan memberikan akses untuk menambah atau mengedit data anggota melalui halaman `FormMember.php`.

FormMember.php

1. Baris 1-15: Membuka tag PHP dan menyertakan file `Koneksi.php` dan `Model.php`. Selanjutnya, membuat instance objek dari kelas `Model` dengan parameter objek PDO `\$pdo` yang sudah terkoneksi ke basis data.

2. Baris 16-23: Mendapatkan nilai `id` dan `mode` dari query string. Jika mode adalah `edit` dan terdapat nilai `id`, maka ambil data anggota dari basis data berdasarkan ID. Jika tidak, inisialisasi variabel `\$member` dengan array kosong atau data anggota default.

3. Baris 24-39: Memproses data yang dikirimkan melalui metode POST. Jika metode POST terdeteksi, data diproses sesuai dengan mode yang diberikan (edit atau tambah). Setelah pemrosesan, pengguna akan dialihkan kembali ke halaman `Member.php`.

4. Baris 41-65: Bagian HTML dari halaman web, yang akan dirender oleh browser. Ini adalah formulir untuk menambahkan atau mengedit data anggota. Jika mode adalah 'edit',

formulir akan diisi dengan data anggota yang akan diubah. Ketika formulir disubmit, data akan dikirim ke halaman itu sendiri untuk diproses.

5. ``$mode === 'edit'`` digunakan untuk menentukan apakah halaman berada dalam mode edit atau tambah data.

6. Fungsi ``readMemberById($id_member)``: Digunakan untuk membaca data anggota berdasarkan ID anggota dari basis data.

7. Fungsi ``insertMember($data)``: Digunakan untuk menambahkan data anggota baru ke dalam basis data.

8. Fungsi ``updateMember($data)``: Digunakan untuk memperbarui data anggota yang sudah ada dalam basis data.

9. Input hidden dengan nama ``id_member`` digunakan untuk menyimpan ID anggota saat mengedit data. Saat menambah data, input ini akan kosong. Itu memungkinkan untuk membedakan antara mode tambah dan mode edit saat data disubmit.

10. Tombol "Simpan" digunakan untuk mengirimkan formulir ke server untuk diproses. Saat tombol ini ditekan, data yang dimasukkan akan dikirimkan ke halaman itu sendiri melalui metode POST.

Buku.php

1. Pemanggilan Koneksi dan Model: Kode dimulai dengan memanggil file ``Koneksi.php`` dan ``Model.php`` yang berisi konfigurasi koneksi database dan definisi kelas Model. Setelah itu, objek Model dibuat dengan parameter objek PDO ``$pdo`` yang telah terkoneksi ke database.

2. Penanganan Aksi Delete: Terdapat penanganan aksi delete jika parameter ``action`` yang diterima dari URL adalah `'delete'` dan jika terdapat parameter ``id_buku`` yang tidak kosong. Jika kondisi ini terpenuhi, fungsi ``deleteDataByStruktur()`` dari objek Model dipanggil untuk menghapus data buku dengan ID yang sesuai. Setelah itu, pengguna dialihkan kembali ke halaman ``Buku.php`` menggunakan header location.

3. Penanganan Submit Form Tambah Buku: Jika terdeteksi bahwa form telah disubmit (method POST) dan field ``id_buku`` kosong, maka dianggap sebagai tindakan penambahan data buku. Data dari form diambil dan disimpan dalam array ``$data``. Fungsi ``insertBuku()`` dari objek Model dipanggil untuk menyimpan data buku baru ke dalam

database. Jika operasi berhasil, pesan "Data buku berhasil ditambahkan." ditampilkan, jika tidak, pesan error ditampilkan.

4. Penanganan Submit Form Edit Buku: Jika tombol "Update" pada form edit buku ditekan, data dari form diambil dan disimpan dalam array ``$data``. Fungsi ``updateData()`` dari objek Model dipanggil untuk mengupdate data buku yang telah ada dalam database. Setelah itu, pengguna dialihkan kembali ke halaman ``Buku.php`` menggunakan header location.

5. Pengambilan Data untuk Edit: Jika terdapat parameter ``id_buku`` dalam URL, maka halaman dianggap sedang dalam mode edit. Data buku dengan ID yang sesuai diambil dari database menggunakan fungsi ``readDataByStruktur()`` dari objek Model dan disimpan dalam variabel ``$bukuToEdit``.

6. Tampilan HTML: Seluruh tampilan HTML untuk menampilkan tabel buku, form tambah/edit buku, dan tombol navigasi ditampilkan di bagian bawah kode. Jika dalam mode edit, form edit buku ditampilkan dengan data buku yang sesuai. Jika tidak, tabel buku ditampilkan dengan tombol edit dan hapus untuk setiap baris data. Juga, terdapat tombol "Tambah Buku" yang mengarahkan ke halaman ``FormBuku.php`` untuk menambahkan buku baru.

FormBuku.php

1. Pemanggilan Koneksi dan Model: Kode memanggil file ``Koneksi.php`` dan ``Model.php`` yang berisi konfigurasi koneksi database dan definisi kelas Model. Kemudian, objek Model dibuat dengan parameter objek PDO ``$pdo`` yang telah terkoneksi ke database.

2. Penanganan Submit Form: Baris kode pertama setelah pemanggilan koneksi dan model adalah penanganan dari submit form. Jika tombol submit dengan nama ``submit`` ditekan, maka kode di dalam blok ini dieksekusi. Data dari form diambil dan disimpan dalam array ``$data``.

3. Validasi Data: Dilakukan pengecekan apakah semua bidang formulir telah diisi. Jika ada bidang yang kosong, maka pesan error "Error: Semua bidang harus diisi." akan ditampilkan.

4. Penambahan Data Buku: Jika semua bidang formulir telah diisi, data buku akan disimpan ke dalam database menggunakan fungsi `insertBuku()` dari objek Model. Jika operasi penyimpanan berhasil, pesan "Data buku berhasil ditambahkan." akan ditampilkan dan pengguna akan diarahkan kembali ke halaman `Buku.php` menggunakan fungsi `header('Location: Buku.php')`. Exit digunakan untuk memastikan tidak ada kode ekstra yang dieksekusi setelah redirect.

5. Tampilan Formulir HTML: Bagian terakhir dari kode adalah tampilan HTML untuk menampilkan formulir tambah buku. Formulir ini memiliki field untuk judul buku, pengarang, penerbit, dan tahun terbit. Setiap field dilengkapi dengan label dan input untuk memasukkan data. Ketika tombol submit ditekan, data akan dikirim ke halaman ini sendiri (`$_SERVER["PHP_SELF"]`)

Peminjaman.php

1. `<?php`: Tag pembuka untuk menandakan bahwa ini adalah kode PHP.
2. `require "Koneksi.php";`: Menggunakan pernyataan `require` untuk menyertakan file `"Koneksi.php"`, yang kemungkinan berisi kode untuk membuat koneksi ke database.
3. `require "Model.php";`: Menggunakan pernyataan `require` untuk menyertakan file `"Model.php"`, yang kemungkinan berisi definisi dari kelas Model.
4. `$model = new Model($pdo);`: Membuat objek dari kelas Model dengan parameter `$pdo`, yang kemungkinan adalah objek PDO yang digunakan untuk berinteraksi dengan database.
5. `if (!empty($_GET['id_peminjaman'])) { ... }`: Memeriksa apakah ada parameter `id_peminjaman` yang diterima melalui metode GET. Jika ada, maka kode di dalam blok ini akan dijalankan. Ini digunakan untuk menghapus data peminjaman berdasarkan `id_peminjaman` yang diberikan.
6. `$id_peminjaman = $_GET['id_peminjaman'];`: Mengambil nilai `id_peminjaman` dari parameter GET.
7. `$model->deleteDataByStruktur("peminjaman", $id_peminjaman, "id_peminjaman");`: Memanggil metode `deleteDataByStruktur` dari objek `$model` untuk menghapus data peminjaman berdasarkan `id_peminjaman` yang diberikan.

8. ``header('location:Peminjaman.php');``: Mengarahkan pengguna kembali ke halaman "Peminjaman.php" setelah menghapus data.
9. ``exit();``: Menghentikan eksekusi skrip PHP saat ini.
10. ``if (isset($_POST['submit']) && empty($_POST['id_peminjaman'])) { ... }``: Memeriksa apakah ada data yang dikirim melalui metode POST dan apakah ``id_peminjaman`` tidak kosong. Jika kondisi terpenuhi, maka kode di dalam blok ini akan dijalankan. Ini digunakan untuk menambahkan data peminjaman baru.
11. ``$data = array(...)``: Membuat array ``$data`` yang berisi data peminjaman baru yang akan dimasukkan ke dalam database.
12. ``if ($model->insertPeminjaman($data)) { ... }``: Memanggil metode ``insertPeminjaman`` dari objek ``$model`` untuk menambahkan data peminjaman baru ke dalam database.
13. ``echo "Data Peminjaman berhasil ditambahkan."``: Pesan yang akan ditampilkan jika data peminjaman berhasil ditambahkan.
14. ``else { echo "Error: Gagal menambahkan data peminjaman."; }``: Pesan yang akan ditampilkan jika terjadi kesalahan saat menambahkan data peminjaman.
15. ``<!DOCTYPE html> ... </html>``: Bagian HTML dari halaman web, yang digunakan untuk menampilkan data peminjaman dan formulir untuk menambahkan atau mengedit data.
16. ``<style> ... </style>``: Bagian CSS untuk styling halaman web.
17. ``<div class="container"> ... </div>``: Membuat kontainer untuk menempatkan konten halaman web.
18. ``<table> ... </table>``: Mendefinisikan tabel untuk menampilkan data peminjaman.
19. ``<thead> ... </thead>``: Mendefinisikan bagian kepala tabel.
20. ``<tbody> ... </tbody>``: Mendefinisikan bagian tubuh tabel, di mana data peminjaman akan ditampilkan.
21. ``<?php ... ?>``: Tag PHP digunakan di dalam elemen HTML untuk mengeksekusi kode PHP yang menghasilkan data dinamis.
22. ``echo $data['id_peminjaman']``: Menampilkan nilai ``id_peminjaman`` dari setiap data peminjaman di dalam sel tabel.

23. ` ... `: Membuat tautan untuk mengedit data peminjaman, dengan menambahkan parameter `id_peminjaman` ke URL.
24. ` ... `: Membuat tautan untuk menghapus data peminjaman. Fungsi `confirm()` akan menampilkan dialog konfirmasi sebelum penghapusan.
25. ` ... `: Membuat tautan untuk menuju ke halaman "FormPeminjaman.php" untuk menambahkan data peminjaman baru.
26. `</body>` dan `</html>`: Tag penutup untuk menutup elemen HTML dan dokumen HTML secara keseluruhan.

FormPeminjaman.php

1. `require "koneksi.php";` dan `require "model.php";`: Menggunakan pernyataan `require` untuk menyertakan file "koneksi.php" dan "model.php" yang kemungkinan berisi kode untuk koneksi ke database dan definisi kelas Model.
2. `$model = new Model($pdo);`: Membuat objek Model dengan parameter `$pdo`, yang kemungkinan adalah objek PDO yang digunakan untuk berinteraksi dengan database.
3. `if (isset($_POST['submit'])) { ... }`: Memeriksa apakah tombol "submit" telah ditekan pada formulir. Jika ya, kode di dalam blok ini akan dijalankan, yang mengindikasikan bahwa pengguna sedang mencoba untuk menambahkan data peminjaman baru.
4. `if (isset($_POST['edit'])) { ... }`: Memeriksa apakah tombol "edit" telah ditekan pada formulir. Jika ya, kode di dalam blok ini akan dijalankan, yang mengindikasikan bahwa pengguna sedang mencoba untuk mengedit data peminjaman yang ada.
5. `if (isset($_POST['kembali'])) { ... }`: Memeriksa apakah tombol "kembali" telah ditekan pada formulir. Jika ya, kode di dalam blok ini akan dijalankan, yang mengarahkan pengguna kembali ke halaman "Peminjaman.php".
6. `$id = isset($_GET['id_peminjaman']) ? $_GET['id_peminjaman'] : "";`: Mendapatkan nilai dari parameter GET 'id_peminjaman' jika ada, jika tidak, nilai defaultnya adalah string kosong.

7. ``$data = $model->selectDataById('peminjaman', $id, "id_peminjaman");``: Memanggil metode ``selectDataById`` dari objek ``$model`` untuk mendapatkan data peminjaman berdasarkan ID yang diberikan.
8. ``<!doctype html> ... </html>``: Bagian HTML dari halaman web, yang digunakan untuk menampilkan formulir untuk menambahkan atau mengedit data peminjaman.
9. ``<div class="container"> ... </div>``: Membuat kontainer untuk menempatkan konten halaman web.
10. ``<?php if (isset($_GET['id_peminjaman'])) : ?> ... <?php else : ?> ... <?php endif; ?>``: Menggunakan sintaks alternatif PHP untuk mengecek apakah parameter GET `'id_peminjaman'` telah disediakan. Jika ya, tampilkan formulir untuk mengedit data peminjaman yang ada, jika tidak, tampilkan formulir untuk menambahkan data peminjaman baru.
11. ``<?php echo isset($data['id_peminjaman']) ? htmlspecialchars($data['id_peminjaman']) : ""; ?>``: Menampilkan nilai ID Peminjaman yang mungkin sudah ada dalam formulir, dengan menggunakan ``htmlspecialchars`` untuk mencegah serangan XSS.
12. ``if (isset($_GET['id_peminjaman'])) { ... } else { ... }``: Menampilkan formulir sesuai kondisi apakah pengguna sedang dalam mode edit atau tambah data baru.
13. ``<button type="submit" name="edit">Simpan Perubahan</button>`` dan ``<button type="submit" name="submit">Tambah Data</button>``: Tombol untuk menyimpan perubahan data yang diedit atau menambahkan data baru.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/aisyainind/PRAKTIKUM-PEMROGRAMAN-WEB-II/tree/main/Modul%205>