

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 5**



Function dan Database

Oleh:

Nor Aina Eva Yanti NIM. 2210817120004

**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 : Nor Aina Eva Yanti
NIM : 2210817120004

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.....	6
A. Source Code	7
B. Output Program.....	18
C. Pembahasan	19
D. Tautan Git	28

DAFTAR GAMBAR

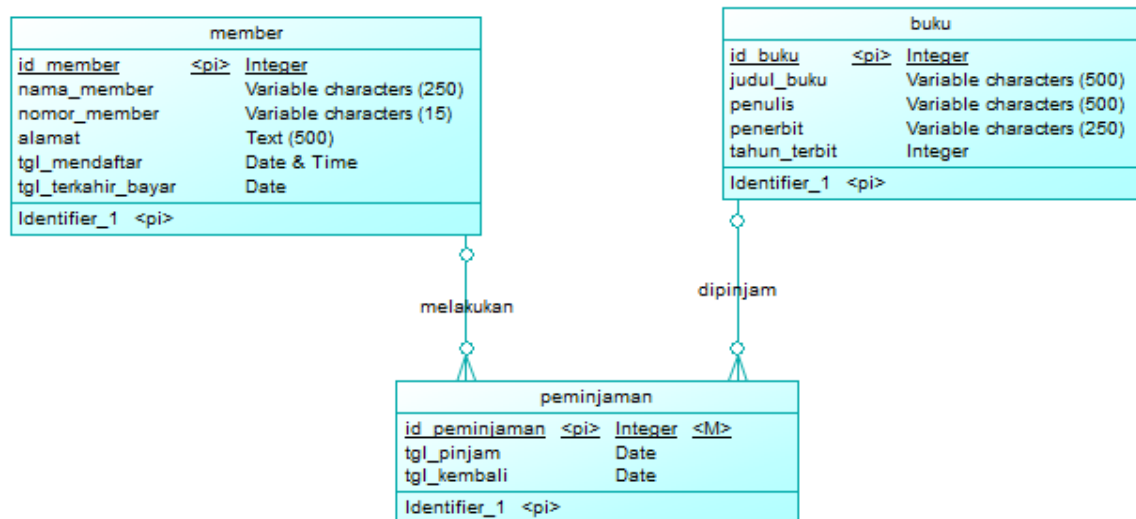
Gambar 1. MODUL 5 Screenshot Hasil Jawaban Member.php	18
Gambar 2. MODUL 5 Screenshot Hasil Jawaban FormMember.php.....	18
Gambar 3. MODUL 5 Screenshot Hasil Jawaban Buku.php	18
Gambar 4. MODUL 5 Screenshot Hasil Jawaban FormBuku.php.....	19
Gambar 5. MODUL 5 Screenshot Hasil Jawaban Peminjaman.php	19
Gambar 6. MODUL 5 Screenshot Hasil Jawaban FormPeminjaman.php	19

DAFTAR TABEL

Tabel 1. MODUL 5 Source Code Koneksi.php.....	8
Tabel 2. MODUL 5 Source Code Model.php	11
Tabel 3. MODUL 5 Source Code Member.php	12
Tabel 4. MODUL 5 Source Code FormMember.php.....	13
Tabel 5. MODUL 5 Source Code Buku.php	14
Tabel 6. MODUL 5 Source Code FormBuku.php.....	15
Tabel 7. MODUL 5 Source Code Peminjaman.php	16
Tabel 8. MODUL 5 Source Code FormPeminjaman.php	18

SOAL

- Diberikan desain konseptual sebuah basis data seperti berikut



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

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

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

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

Simpan dengan nama projek di Github: PRAK501

A. Source Code

```

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

```

```

1  <?php
2  include_once 'Koneksi.php';
3
4  class Model {
5      private $conn;
6
7      public function __construct() {
8          $database = new Koneksi();
9          $this->conn = $database->getConnection();
10     }
11
12     private function executeQuery($query, $params = [])
13 {
14         $stmt = $this->conn->prepare($query);
15         foreach ($params as $param => $value) {
16             $stmt->bindValue($param, $value);
17         }
18         return $stmt->execute();
19     }
20
21     public function getMembers() {
22         $query = "SELECT * FROM member";
23         $stmt = $this->conn->prepare($query);
24         $stmt->execute();
25         return $stmt->fetchAll(PDO::FETCH_ASSOC);
26     }
27
28     public function addMember($nama_member,
29     $nomor_member, $tgl_mendaftar, $tgl_terakhir_bayar) {
30         $query = "INSERT INTO member (nama_member,
31     nomor_member, tgl_mendaftar, tgl_terakhir_bayar) VALUES
32     (:nama_member, :nomor_member, :tgl_mendaftar,
33     :tgl_terakhir_bayar)";
34         return $this->executeQuery($query, [
35             ':nama_member' => $nama_member,
36             ':nomor_member' => $nomor_member,
37             ':tgl_mendaftar' => $tgl_mendaftar,
38             ':tgl_terakhir_bayar' => $tgl_terakhir_bayar
39         ]);
40     }
41
42     public function updateMember($id_member,
43     $nama_member, $nomor_member, $tgl_mendaftar,
44     $tgl_terakhir_bayar) {
45         $query = "UPDATE member SET nama member =

```



```

:nama_member,      nomor_member      =      :nomor_member,
tgl_mendaftar      =      :tgl_mendaftar,  tgl_terakhir_bayar      =
39 :tgl_terakhir_bayar WHERE id_member = :id_member";
40     return $this->executeQuery($query, [
41         ':id_member' => $id_member,
42         ':nama_member' => $nama_member,
43         ':nomor_member' => $nomor_member,
44         ':tgl_mendaftar' => $tgl_mendaftar,
45         ':tgl_terakhir_bayar' => $tgl_terakhir_bayar
46     ]);
47 }
48
49     public function deleteMember($id_member) {
        $query = "DELETE FROM member WHERE id_member =
50 :id_member";
        return $this->executeQuery($query, [':id_member'
51 => $id_member]);
52     }
53
54     public function getBooks() {
55         $query = "SELECT * FROM buku";
56         $stmt = $this->conn->prepare($query);
57         $stmt->execute();
58         return $stmt->fetchAll(PDO::FETCH_ASSOC);
59     }
60
61     public function addBook($judul_buku, $penulis,
    $tahun_terbit) {
        $query = "INSERT INTO buku (judul_buku, penulis,
    tahun_terbit)
62         VALUES (:judul_buku, :penulis,
    :tahun_terbit)";
63         return $this->executeQuery($query, [
64             ':judul_buku' => $judul_buku,
65             ':penulis' => $penulis,
66             ':tahun_terbit' => $tahun_terbit
67         ]);
68     }
69
70     public function updateBook($id_buku, $judul_buku,
    $penulis, $tahun_terbit) {
        $query = "UPDATE buku SET judul_buku =
    :judul_buku, penulis = :penulis, tahun_terbit =
71 :tahun_terbit WHERE id_buku = :id_buku";
72         return $this->executeQuery($query, [
73             ':id_buku' => $id_buku,
74             ':judul_buku' => $judul_buku,
75             ':penulis' => $penulis,

```

```

76         ':tahun_terbit' => $tahun_terbit
77     });
78 }
79
80     public function deleteBook($id_buku) {
81         $query = "DELETE FROM buku WHERE id_buku =
82         :id_buku";
83         return $this->executeQuery($query, ['id_buku'
84         => $id_buku]);
85     }
86
87     public function getLoans() {
88         $query = "SELECT * FROM peminjaman";
89         $stmt = $this->conn->prepare($query);
90         $stmt->execute();
91         return $stmt->fetchAll(PDO::FETCH_ASSOC);
92     }
93
94     public function addLoan($id_member, $id_buku,
95     $tgl_pinjam, $tgl_kembali) {
96         $query = "INSERT INTO peminjaman (id_member,
97     id_buku, tgl_pinjam, tgl_kembali) VALUES (:id_member,
98     :id_buku, :tgl_pinjam, :tgl_kembali)";
99         return $this->executeQuery($query, [
100             ':id_member' => $id_member,
101             ':id_buku' => $id_buku,
102             ':tgl_pinjam' => $tgl_pinjam,
103             ':tgl_kembali' => $tgl_kembali
104         ]);
105     }
106
107     public function updateLoan($id_peminjaman,
108     $id_member, $id_buku, $tgl_pinjam, $tgl_kembali) {
109         $query = "UPDATE peminjaman SET id_member =
110     :id_member, id_buku = :id_buku, tgl_pinjam =
111     :tgl_pinjam, tgl_kembali = :tgl_kembali WHERE
112     id_peminjaman = :id_peminjaman";
113         return $this->executeQuery($query, [
114             ':id_peminjaman' => $id_peminjaman,
115             ':id_member' => $id_member,
116             ':id_buku' => $id_buku,
117             ':tgl_pinjam' => $tgl_pinjam,
118             ':tgl_kembali' => $tgl_kembali
119         ]);
120     }
121
122     public function deleteLoan($id_peminjaman) {

```

113	\$query = "DELETE FROM peminjaman WHERE id_peminjaman = :id_peminjaman";
114	return \$this->executeQuery(\$query, [':id_peminjaman' => \$id_peminjaman]);
115	}
116	}
117	?>

Tabel 2. MODUL 5 Source Code Model.php

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

33	confirm('Yakin ingin menghapus data?')">Hapus
34	</td>
35	</tr>
36	<?php } ?>
37	</table>
38	
39	Tambah Member
40	</body>
	</html>

Tabel 3. MODUL 5 Source Code Member.php

1	<?php
2	include_once 'Model.php';
3	\$model = new Model();
4	
5	\$id_member = '';
6	\$nama_member = '';
7	\$nomor_member = '';
8	\$tgl_mendaftar = '';
9	\$tgl_terakhir_bayar = '';
10	
11	if (isset(\$_GET['id'])) {
12	\$id_member = \$_GET['id'];
13	\$member = \$model->getMemberById(\$id_member);
14	if (\$member) {
15	\$nama_member = \$member['nama_member'];
16	\$nomor_member = \$member['nomor_member'];
17	\$tgl_mendaftar = \$member['tgl_mendaftar'];
18	\$tgl_terakhir_bayar = \$member['tgl_terakhir_bayar'];
19	}
20	}
21	
22	if (\$_POST) {
23	if (\$id_member) {
24	\$model->updateMember(\$_POST['id_member'],
25	\$_POST['nama_member'], \$_POST['nomor_member'],
26	\$_POST['tgl_mendaftar'], \$_POST['tgl_terakhir_bayar']);
27	} else {
28	\$model->addMember(\$_POST['nama_member'],
29	\$_POST['nomor_member'], \$_POST['tgl_mendaftar'],
30	\$_POST['tgl_terakhir_bayar']);
31	}
32	header('Location: Member.php');
	}
	?>
	<!DOCTYPE html>

```

33 <html>
34 <head>
35     <title>Form Member</title>
36 </head>
37 <body>
38     <h1><?php echo $id_member ? 'Edit' : 'Tambah'; ?>
Member</h1>
39     <form method="post">
40         <input type="hidden" name="id_member"
value="<?php echo $id_member; ?>">
41         <label>Nama Member:</label>
42         <input type="text" name="nama_member"
value="<?php echo $nama_member; ?>" required><br>
43         <label>Nomor Member:</label>
44         <input type="text" name="nomor_member"
value="<?php echo $nomor_member; ?>" required><br>
45         <label>Tanggal Mendaftar:</label>
46         <input type="date" name="tgl_mendaftar"
value="<?php echo $tgl_mendaftar; ?>" required><br>
47         <label>Tanggal Terakhir Bayar:</label>
48         <input type="date" name="tgl_terakhir_bayar"
value="<?php echo $tgl_terakhir_bayar; ?>" required><br>
49         <input type="submit" value="Simpan">
50     </form>
51     <br>
52     <a href="Member.php">Kembali</a>
53 </body>
54 </html>

```

Tabel 4. MODUL 5 Source Code FormMember.php

```

1 <?php
2 include_once 'Model.php';
3 $model = new Model();
4 $books = $model->getBooks();
5 ?>
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10     <title>Buku</title>
11 </head>
12 <body>
13     <h1>Data Buku</h1>
14     <table border="1">
15         <tr>
16             <th>ID Buku</th>
17             <th>Judul Buku</th>
18             <th>Penulis</th>

```

```

19         <th>Tahun Terbit</th>
20         <th>Aksi</th>
21     </tr>
22     <?php foreach ($books as $book) { ?>
23         <tr>
24             <td><?php echo $book['id_buku']; ?></td>
25             <td><?php echo $book['judul_buku'];
?></td>
26             <td><?php echo $book['penulis']; ?></td>
27             <td><?php echo $book['tahun_terbit'];
?></td>
28             <td>
29                 <a href="FormBuku.php?id=<?php echo
$book['id_buku']; ?>">Edit</a>
30                 <a href="deleteBuku.php?id=<?php echo
$book['id_buku']; ?>" onclick="return confirm('Yakin
ingin menghapus data?')">Hapus</a>
31             </td>
32         </tr>
33     <?php } ?>
34 </table>
35 <br>
36 <a href="FormBuku.php">Tambah Buku</a>
37 </body>
38 </html>

```

Tabel 5. MODUL 5 Source Code Buku.php

```

1 <?php
2     include_once 'Model.php';
3     $model = new Model();
4
5     $id_buku = '';
6     $judul_buku = '';
7     $penulis = '';
8     $tahun_terbit = '';
9
10    if (isset($_GET['id'])) {
11        $id_buku = $_GET['id'];
12        $book = $model->getBookById($id_buku);
13        if ($book) {
14            $judul_buku = $book['judul_buku'];
15            $penulis = $book['penulis'];
16            $tahun_terbit = $book['tahun_terbit'];
17        }
18    }
19
20    if ($_POST) {
21        if ($id_buku) {

```

```

22         $model->updateBook($_POST['id_buku'],
$_POST['judul_buku'],          $_POST['penulis'],
$_POST['tahun_terbit']);
23     } else {
24         $model->addBook($_POST['judul_buku'],
$_POST['penulis'], $_POST['tahun_terbit']);
25     }
26     header('Location: Buku.php');
27 }
28 ?>
29
30 <!DOCTYPE html>
31 <html>
32 <head>
33     <title>Form Buku</title>
34 </head>
35 <body>
36     <h1><?php echo $id_buku ? 'Edit' : 'Tambah'; ?>
Buku</h1>
37     <form method="post">
38         <input type="hidden" name="id_buku" value="<?php
echo $id_buku; ?>">
39         <label>Judul Buku:</label>
40         <input type="text" name="judul_buku" value="<?php
echo $judul_buku; ?>" required><br>
41         <label>Penulis:</label>
42         <input type="text" name="penulis" value="<?php
echo $penulis; ?>" required><br>
43         <label>Tahun Terbit:</label>
44         <input type="text" name="tahun_terbit"
value="<?php echo $tahun_terbit; ?>" required><br>
45         <input type="submit" value="Simpan">
46     </form>
47     <br>
48     <a href="Buku.php">Kembali</a>
49 </body>
50 </html>

```

Tabel 6. MODUL 5 Source Code FormBuku.php

```

1 <?php
2 include_once 'Model.php';
3 $model = new Model();
4 $loans = $model->getLoans();
5 ?>
6
7 <!DOCTYPE html>
8 <html>
9 <head>

```

```

10     <title>Peminjaman</title>
11 </head>
12 <body>
13     <h1>Data Peminjaman</h1>
14     <table border="1">
15         <tr>
16             <th>ID Peminjaman</th>
17             <th>ID Member</th>
18             <th>ID Buku</th>
19             <th>Tanggal Pinjam</th>
20             <th>Tanggal Kembali</th>
21             <th>Aksi</th>
22         </tr>
23         <?php foreach ($loans as $loan) { ?>
24             <tr>
25                 <td><?php echo $loan['id_peminjaman'];
26                 ?></td>
27                 <td><?php echo $loan['id_member'];
28                 ?></td>
29                 <td><?php echo $loan['id_buku']; ?></td>
30                 <td><?php echo $loan['tgl_pinjam'];
31                 ?></td>
32                 <td><?php echo $loan['tgl_kembali'];
33                 ?></td>
34                 <td>
35                     <a href="FormPeminjaman.php?id=<?php
36                     echo $loan['id_peminjaman']; ?>">Edit</a>
37                     <a
38                     href="deletePeminjaman.php?id=<?php
39                     $loan['id_peminjaman']; ?>"
40                     onclick="return
41                     confirm('Yakin ingin menghapus data?')">Hapus</a>
42                 </td>
43             </tr>
44             <?php } ?>
45         </table>
46         <br>
47         <a href="FormPeminjaman.php">Tambah Peminjaman</a>
48     </body>
49 </html>

```

Tabel 7. MODUL 5 Source Code Peminjaman.php

```

1 <?php
2     include_once 'Model.php';
3     $model = new Model();
4
5     $id_peminjaman = '';
6     $id_member = '';
7     $id_buku = '';

```



```

8      $tgl_pinjam = '';
9      $tgl_kembali = '';
10
11     if (isset($_GET['id'])) {
12         $id_peminjaman = $_GET['id'];
13         $loan = $model->getLoanById($id_peminjaman);
14         if ($loan) {
15             $id_member = $loan['id_member'];
16             $id_buku = $loan['id_buku'];
17             $tgl_pinjam = $loan['tgl_pinjam'];
18             $tgl_kembali = $loan['tgl_kembali'];
19         }
20     }
21
22     if ($_POST) {
23         if ($id_peminjaman) {
24             $model->updateLoan($_POST['id_peminjaman'],
25 $_POST['id_member'], $_POST['id_buku'],
26 $_POST['tgl_pinjam'], $_POST['tgl_kembali']);
27         } else {
28             $model->addLoan($_POST['id_member'],
29 $_POST['id_buku'], $_POST['tgl_pinjam'],
30 $_POST['tgl_kembali']);
31         }
32         header('Location: Peminjaman.php');
33     }
34     ?>
35
36 <!DOCTYPE html>
37 <html>
38 <head>
39     <title>Form Peminjaman</title>
40 </head>
41 <body>
42     <h1><?php echo $id_peminjaman ? 'Edit' : 'Tambah'; ?>
43 Peminjaman</h1>
44     <form method="post">
45         <input type="hidden" name="id_peminjaman"
46 value="<?php echo $id_peminjaman; ?>">
47         <label>ID Member:</label>
48         <input type="number" name="id_member"
49 value="<?php echo $id_member; ?>" required><br>
50         <label>ID Buku:</label>
51         <input type="number" name="id_buku" value="<?php
52 echo $id_buku; ?>" required><br>
53         <label>Tanggal Pinjam:</label>
54         <input type="date" name="tgl_pinjam" value="<?php

```

47	echo \$tgl_pinjam; ?>" required>
48	<label>Tanggal Kembali:</label>
49	<input type="date" name="tgl_kembali" value="<?php echo \$tgl_kembali; ?>" required>
50	<input type="submit" value="Simpan">
51	</form>
52	
53	Kembali
54	</body>
	</html>

Tabel 8. MODUL 5 Source Code FormPeminjaman.php

B. Output Program

Data Member

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

[Tambah Member](#)

Gambar 1. MODUL 5 Screenshot Hasil Jawaban Member.php

Tambah Member

Nama Member:

Nomor Member:

Tanggal Mendaftar:

Tanggal Terakhir Bayar:

[Kembali](#)

Gambar 2. MODUL 5 Screenshot Hasil Jawaban FormMember.php

Data Buku

ID Buku	Judul Buku	Penulis	Tahun Terbit	Aksi
1	bulan	eva	2024	Edit Hapus

[Tambah Buku](#)

Gambar 3. MODUL 5 Screenshot Hasil Jawaban Buku.php

Tambah Buku

Judul Buku:

Penulis:

Tahun Terbit:

[Kembali](#)

Gambar 4. MODUL 5 Screenshot Hasil Jawaban FormBuku.php

Data Peminjaman

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

[Tambah Peminjaman](#)

Gambar 5. MODUL 5 Screenshot Hasil Jawaban Peminjaman.php

Tambah Peminjaman

ID Member:

ID Buku:

Tanggal Pinjam:

Tanggal Kembali:

[Kembali](#)

Gambar 6. MODUL 5 Screenshot Hasil Jawaban FormPeminjaman.php

C. Pembahasan

File Koneksi.php

Pada baris [1] terdapat `<?php` yang menandakan bahwa kode ini adalah script PHP. PHP adalah bahasa pemrograman server-side yang digunakan untuk membuat halaman web dinamis.

Pada baris [2] terdapat `<class Koneksi>` yang mendefinisikan sebuah kelas bernama Koneksi. Kelas ini digunakan untuk mengelola koneksi ke database MySQL.

Pada baris [3] terdapat `<private $host = "localhost";>`. Baris ini mendefinisikan variabel `host` dengan akses `private`, yang berarti variabel ini hanya bisa diakses dari dalam kelas. Nilai variabel ini adalah `"localhost"`, yang menunjukkan server database berada di mesin lokal.

Pada baris [4] terdapat `<private $dbname = "perpustakaan";>`. Baris ini mendefinisikan variabel `dbname` dengan akses `private`. Nilai variabel ini adalah `"perpustakaan"`, yang menunjukkan nama database yang akan dihubungkan.

Pada baris [5] terdapat `<private $username = "root";>`. Baris ini mendefinisikan variabel username dengan akses private. Nilai variabel ini adalah "root", yang merupakan nama pengguna default untuk database MySQL.

Pada baris [6] terdapat `<private $password = ">`. Baris ini mendefinisikan variabel password dengan akses private. Nilai variabel ini adalah string kosong, yang merupakan kata sandi default untuk pengguna "root".

Pada baris [7] terdapat `<public $conn;>`. Baris ini mendefinisikan variabel conn dengan akses public. Variabel ini akan menyimpan objek koneksi ke database.

Pada baris [9] terdapat `<public function getConnection(>`. Baris ini mendefinisikan sebuah fungsi public bernama getConnection. Fungsi ini bertanggung jawab untuk membuat koneksi ke database.

Pada baris [10] terdapat `$this->conn = null;`. Baris ini menginisialisasi variabel conn dengan nilai null. Ini memastikan bahwa variabel conn kosong sebelum mencoba membuat koneksi.

Pada baris [12] terdapat `try {`. Baris ini memulai blok try-catch, yang digunakan untuk menangani kesalahan saat mencoba menghubungkan ke database.

Pada baris [13] terdapat `$this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->dbname, $this->username, $this->password);`. Baris ini mencoba membuat objek PDO baru dengan menggunakan informasi koneksi yang disimpan dalam variabel host, dbname, username, dan password. PDO (PHP Data Objects) adalah ekstensi PHP yang menyediakan antarmuka yang konsisten untuk mengakses berbagai database.

Pada baris [14] terdapat `$this->conn->exec("set names utf8");`. Baris ini memastikan bahwa koneksi menggunakan set karakter UTF-8, yang penting untuk menangani berbagai karakter internasional dengan benar.

Pada baris [15] terdapat `} catch(PDOException $exception) {`. Baris ini menangkap setiap kesalahan yang terjadi saat mencoba membuat koneksi dan memulai blok catch.

Pada baris [16] terdapat `echo "Connection error: " . $exception->getMessage();`. Jika terjadi kesalahan, pesan error akan ditampilkan dengan menggunakan echo yang diikuti oleh pesan kesalahan yang diambil dari objek exception.

Pada baris [18] terdapat `return $this->conn;`. Baris ini mengembalikan objek koneksi jika berhasil dibuat, atau null jika terjadi kesalahan.

Pada baris [19] terdapat `}` yang menandakan akhir dari fungsi getConnection.

Pada baris [20] terdapat `}` yang menandakan akhir dari kelas Koneksi.

File Model.php

Pada baris [1], terdapat deklarasi `<?php`, menandakan bahwa kode ini adalah script PHP, dimulai dari sini.

Pada baris [2], `include_once 'Koneksi.php';` merupakan perintah untuk memasukkan (include) file `Koneksi.php`, yang kemungkinan berisi kelas atau fungsi untuk mengelola koneksi ke database.

Pada baris [4-6], terdapat definisi kelas `Model`, yang merupakan kelas untuk mengelola data dari database. Kelas ini memiliki properti `$conn` yang merupakan koneksi ke database.

Pada baris [8-11], terdapat konstruktor `__construct()`, yang akan dijalankan setiap kali objek dari kelas `Model` dibuat. Konstruktor ini membuat objek koneksi ke database dengan memanggil kelas `Koneksi` dan menyimpannya di dalam properti `$conn`.

Pada baris [13-17], terdapat method `executeQuery($query, $params = [])`, yang merupakan method private untuk mengeksekusi query ke database. Method ini menggunakan prepared statement untuk menghindari SQL injection.

Pada baris [19-23], terdapat method `getMembers()`, yang digunakan untuk mengambil data dari tabel member dalam database.

Pada baris [25-31], terdapat method `addMember($nama_member, $nomor_member, $tgl_mendaftar, $tgl_terakhir_bayar)`, yang digunakan untuk menambahkan data member baru ke dalam tabel member.

Pada baris [33-39], terdapat method `updateMember($id_member, $nama_member, $nomor_member, $tgl_mendaftar, $tgl_terakhir_bayar)`, yang digunakan untuk memperbarui data member yang sudah ada dalam tabel member.

Pada baris [41-47], terdapat method `deleteMember($id_member)`, yang digunakan untuk menghapus data member dari tabel member.

Pada baris [49-53], terdapat method `getBooks()`, yang digunakan untuk mengambil data dari tabel buku dalam database.

Pada baris [55-61], terdapat method `addBook($judul_buku, $penulis, $tahun_terbit)`, yang digunakan untuk menambahkan data buku baru ke dalam tabel buku.

Pada baris [63-69], terdapat method `updateBook($id_buku, $judul_buku, $penulis, $tahun_terbit)`, yang digunakan untuk memperbarui data buku yang sudah ada dalam tabel buku.

Pada baris [71-75], terdapat method `deleteBook($id_buku)`, yang digunakan untuk menghapus data buku dari tabel buku.

Pada baris [77-81], terdapat method `getLoans()`, yang digunakan untuk mengambil data dari tabel peminjaman dalam database.

Pada baris [83-89], terdapat method `addLoan($id_member, $id_buku, $tgl_pinjam, $tgl_kembali)`, yang digunakan untuk menambahkan data peminjaman baru ke dalam tabel peminjaman.

Pada baris [91-97], terdapat method `updateLoan($id_peminjaman, $id_member, $id_buku, $tgl_pinjam, $tgl_kembali)`, yang digunakan untuk memperbarui data peminjaman yang sudah ada dalam tabel peminjaman.

Pada baris [99-103], terdapat method `deleteLoan($id_peminjaman)`, yang digunakan untuk menghapus data peminjaman dari tabel peminjaman.

Pada baris [105], penutup blok kelas `Model`, menandakan akhir dari definisi kelas `Model`.

Pada baris [106-107], penutup blok PHP `?`>, menandakan akhir dari script PHP.

File Member.php

Pada baris [1], `include_once 'Model.php';` digunakan untuk memasukkan file `Model.php`, yang berisi definisi kelas `Model` untuk mengelola data dari database.

Pada baris [2], `$model = new Model();` membuat objek baru dari kelas `Model`, yang akan digunakan untuk mengakses method-method yang ada dalam kelas tersebut.

Pada baris [3], `$members = $model->getMembers();` memanggil method `getMembers()` dari objek `$model` untuk mengambil data member dari database dan menyimpannya dalam variabel `$members`.

Berikutnya, kode HTML dimulai dari `<!DOCTYPE html>` hingga `</html>`, yang merupakan struktur dasar halaman web.

Pada baris [9-14], terdapat sebuah tabel HTML untuk menampilkan data member. Dalam tabel ini, menggunakan perulangan `foreach` untuk menampilkan setiap baris data member yang ada dalam variabel `$members`.

Pada baris [15-23], setiap baris data member ditampilkan dalam sebuah baris tabel. Kolom "Aksi" memiliki dua link, yaitu "Edit" yang mengarahkan ke halaman `FormMember.php` dengan parameter `id`, dan "Hapus" yang mengarahkan ke `deleteMember.php` dengan parameter `id` juga, dengan konfirmasi melalui fungsi `confirm()` JavaScript.

Pada baris [24], terdapat link "Tambah Member" yang mengarahkan ke halaman FormMember.php untuk menambahkan member baru.

File FormMember.php

Pada baris [1] kita mulai dengan tag PHP `<?php`. Ini menandakan awal dari kode PHP.

Pada baris [2] dilakukan inclusion file Model.php, yang berisi kelas Model. File ini kemungkinan berisi definisi kelas dan fungsi-fungsi yang dibutuhkan dalam skrip ini.

Pada baris [3] kita membuat objek `$model` dari kelas Model yang telah di-include sebelumnya. Objek ini akan digunakan untuk mengakses metode-metode dalam kelas Model.

Pada baris [5-9] dilakukan inisialisasi variabel `$id_member`, `$nama_member`, `$nomor_member`, `$tgl_mendaftar`, dan `$tgl_terakhir_bayar` dengan string kosong. Ini dilakukan untuk menghindari error ketika variabel ini digunakan sebelum di-set.

Pada baris [11] kita cek apakah ada parameter GET dengan nama 'id'. Jika ada, kita simpan nilainya ke dalam variabel `$id_member`.

Pada baris [12] kita panggil metode `getMemberById` dari objek `$model` untuk mendapatkan data member berdasarkan `$id_member`. Jika data ditemukan, kita set nilai variabel `$nama_member`, `$nomor_member`, `$tgl_mendaftar`, dan `$tgl_terakhir_bayar` dengan data yang didapatkan.

Pada baris [15] kita cek apakah terdapat data yang disubmit melalui metode POST.

Pada baris [16-17] jika `$id_member` tidak kosong, kita panggil metode `updateMember` dari objek `$model` dengan parameter yang sesuai. Jika `$id_member` kosong, kita panggil metode `addMember` untuk menambahkan member baru.

Pada baris [18] setelah proses penambahan atau pembaruan data selesai, kita redirect pengguna ke halaman `Member.php`.

Pada baris [21-31] kita memiliki struktur HTML untuk menampilkan form.

Pada baris [23] terdapat pengecekan apakah `$id_member` kosong. Jika tidak kosong, maka form akan menampilkan data member yang akan diedit. Jika kosong, maka form akan menampilkan form kosong untuk menambahkan member baru.

Pada baris [24-29] terdapat elemen-elemen input yang masing-masing memiliki nilai awal sesuai dengan data member yang akan diedit (jika ada).

Pada baris [32] kita memiliki tombol submit dengan teks 'Simpan' yang akan digunakan untuk menyimpan data.

Pada baris [34] terdapat link 'Kembali' yang akan mengarahkan pengguna kembali ke halaman Member.php setelah selesai mengisi atau mengedit data member.

File Buku.php

Pada baris [1], kita meng-include file Model.php yang kemungkinan berisi definisi kelas Model dan fungsi-fungsi terkait.

Pada baris [2], kita membuat objek \$model dari kelas Model yang sudah di-include sebelumnya. Objek ini akan digunakan untuk mengakses metode-metode dalam kelas Model.

Pada baris [3], kita mengambil data buku dengan memanggil metode getBooks dari objek \$model dan menyimpannya dalam variabel \$books.

Pada baris [7], kita mulai struktur HTML dengan tag <!DOCTYPE html>, menandakan bahwa ini adalah dokumen HTML5.

Pada baris [8-10], terdapat elemen-elemen <head> yang berisi informasi tentang dokumen seperti judul halaman.

Pada baris [11-26], terdapat elemen-elemen <body> yang berisi tampilan data buku dalam bentuk tabel. Setiap buku ditampilkan dalam sebuah baris tabel.

Pada baris [12], kita memiliki judul halaman <h1>Data Buku</h1>.

Pada baris [13-17], kita memiliki tabel dengan border 1 dan header tabel yang berisi kolom-kolom: ID Buku, Judul Buku, Penulis, Tahun Terbit, dan Aksi.

Pada baris [18-25], kita menggunakan loop foreach untuk mengulang data buku dari variabel \$books. Setiap buku akan ditampilkan dalam sebuah baris tabel.

Pada baris [19-22], kita menampilkan data buku dalam kolom-kolom sesuai dengan struktur tabel yang telah ditentukan.

Pada baris [23], kita memiliki kolom Aksi yang berisi dua link, yaitu Edit dan Hapus. Link Edit akan mengarahkan pengguna ke halaman FormBuku.php dengan membawa parameter id buku yang akan diedit. Link Hapus akan mengarahkan pengguna ke deleteBuku.php dengan membawa parameter id buku yang akan dihapus. Sebelum menghapus, pengguna akan diminta konfirmasi terlebih dahulu.

Pada baris [27-29], terdapat link 'Tambah Buku' yang mengarahkan pengguna ke halaman FormBuku.php untuk menambahkan buku baru.

Pada baris [30], penutup tag `</body>` menandakan akhir dari bagian body HTML.

Pada baris [31], penutup tag `</html>` menandakan akhir dari dokumen HTML.

File FormBuku.php

Pada baris [1], kita meng-include file `Model.php` yang kemungkinan berisi definisi kelas Model dan fungsi-fungsi terkait.

Pada baris [2], kita membuat objek `$model` dari kelas Model yang sudah di-include sebelumnya. Objek ini akan digunakan untuk mengakses metode-metode dalam kelas Model.

Pada baris [4-7], variabel `$id_buku`, `$judul_buku`, `$penulis`, dan `$tahun_terbit` diinisialisasi dengan string kosong. Ini dilakukan agar variabel-variabel ini tidak menyebabkan error ketika digunakan sebelum di-set nilainya.

Pada baris [9], kita cek apakah ada parameter GET dengan nama `'id'`. Jika ada, nilai dari parameter tersebut disimpan ke dalam variabel `$id_buku`.

Pada baris [10], kita memanggil metode `getBookById` dari objek `$model` untuk mendapatkan data buku berdasarkan `$id_buku`. Jika data buku ditemukan, kita mengisi variabel `$judul_buku`, `$penulis`, dan `$tahun_terbit` dengan nilai yang sesuai.

Pada baris [12], kita cek apakah ada data yang disubmit melalui metode POST.

Pada baris [13-14], jika `$id_buku` tidak kosong (artinya kita sedang dalam mode edit), maka kita memanggil metode `updateBook` dari objek `$model` dengan parameter yang sesuai. Jika `$id_buku` kosong (artinya kita sedang dalam mode tambah), maka kita memanggil metode `addBook` untuk menambahkan buku baru.

Pada baris [15], setelah proses penambahan atau pembaruan data selesai, kita redirect pengguna ke halaman `Buku.php`.

Pada baris [18-28], terdapat struktur HTML untuk menampilkan form.

Pada baris [20], kita menggunakan ternary operator `(? :)` untuk menampilkan judul form sesuai dengan mode (Edit/Tambah).

Pada baris [21], kita memiliki hidden input untuk menyimpan nilai `$id_buku` agar dapat dikirimkan bersamaan dengan form.

Pada baris [22-24], terdapat input untuk judul buku, penulis, dan tahun terbit. Nilai awal masing-masing diambil dari variabel yang sesuai.

Pada baris [25], terdapat tombol submit dengan teks 'Simpan' yang akan digunakan untuk menyimpan data.

Pada baris [29], terdapat link 'Kembali' yang mengarahkan pengguna kembali ke halaman Buku.php setelah selesai mengisi atau mengedit data buku.

Pada baris [30], penutup tag `</body>` menandakan akhir dari bagian body HTML.

Pada baris [31], penutup tag `</html>` menandakan akhir dari dokumen HTML.

File Peminjaman.php

Pada baris [1], kita meng-include file `Model.php` yang kemungkinan berisi definisi kelas Model dan fungsi-fungsi terkait.

Pada baris [2], kita membuat objek `$model` dari kelas Model yang sudah di-include sebelumnya. Objek ini akan digunakan untuk mengakses metode-metode dalam kelas Model.

Pada baris [4-6], kita mengambil data peminjaman dengan memanggil metode `getLoans` dari objek `$model` dan menyimpannya dalam variabel `$loans`.

Pada baris [8], kita mulai struktur HTML dengan tag `<!DOCTYPE html>`, menandakan bahwa ini adalah dokumen HTML5.

Pada baris [9-11], terdapat elemen-elemen `<head>` yang berisi informasi tentang dokumen seperti judul halaman.

Pada baris [12-22], terdapat elemen-elemen `<body>` yang berisi tampilan data peminjaman dalam bentuk tabel. Setiap data peminjaman ditampilkan dalam sebuah baris tabel.

Pada baris [13], kita memiliki judul halaman `<h1>Data Peminjaman</h1>`.

Pada baris [14-18], kita memiliki tabel dengan border 1 dan header tabel yang berisi kolom-kolom: ID Peminjaman, ID Member, ID Buku, Tanggal Pinjam, Tanggal Kembali, dan Aksi.

Pada baris [19-21], kita menggunakan loop `foreach` untuk mengulang data peminjaman dari variabel `$loans`. Setiap data peminjaman akan ditampilkan dalam sebuah baris tabel.

Pada baris [20-24], kita menampilkan data peminjaman dalam kolom-kolom sesuai dengan struktur tabel yang telah ditentukan.

Pada baris [25], kita memiliki kolom Aksi yang berisi dua link, yaitu Edit dan Hapus. Link Edit akan mengarahkan pengguna ke halaman `FormPeminjaman.php` dengan membawa

parameter id peminjaman yang akan diedit. Link Hapus akan mengarahkan pengguna ke `deletePeminjaman.php` dengan membawa parameter id peminjaman yang akan dihapus. Sebelum menghapus, pengguna akan diminta konfirmasi terlebih dahulu.

Pada baris [27-29], terdapat link 'Tambah Peminjaman' yang mengarahkan pengguna ke halaman `FormPeminjaman.php` untuk menambahkan data peminjaman baru.

Pada baris [30], penutup tag `</body>` menandakan akhir dari bagian body HTML.

Pada baris [31], penutup tag `</html>` menandakan akhir dari dokumen HTML.

File FormPeminjaman.php

Pada baris [1], kita meng-include file `Model.php` yang kemungkinan berisi definisi kelas Model dan fungsi-fungsi terkait.

Pada baris [2], kita membuat objek `$model` dari kelas Model yang sudah di-include sebelumnya. Objek ini akan digunakan untuk mengakses metode-metode dalam kelas Model.

Pada baris [4-8], variabel `$id_peminjaman`, `$id_member`, `$id_buku`, `$tgl_pinjam`, dan `$tgl_kembali` diinisialisasi dengan string kosong atau nilai default. Hal ini dilakukan agar variabel-variabel ini tidak menyebabkan error ketika digunakan sebelum di-set nilainya.

Pada baris [10], kita cek apakah ada parameter GET dengan nama 'id'. Jika ada, nilai dari parameter tersebut disimpan ke dalam variabel `$id_peminjaman`.

Pada baris [11], kita memanggil metode `<getLoanById>` dari objek `<$model>` untuk mendapatkan data peminjaman berdasarkan `$id_peminjaman`. Jika data peminjaman ditemukan, kita mengisi variabel `$id_member`, `$id_buku`, `$tgl_pinjam`, dan `$tgl_kembali` dengan nilai yang sesuai.

Pada baris [13], kita cek apakah ada data yang disubmit melalui metode POST.

Pada baris [14-15], jika `<$id_peminjaman>` tidak kosong (artinya kita sedang dalam mode edit), maka kita memanggil metode `updateLoan` dari objek `<$model>` dengan parameter yang sesuai. `<$id_peminjaman>` kosong (artinya kita sedang dalam mode tambah), maka kita memanggil metode `addLoan` untuk menambahkan data peminjaman baru.

Pada baris [16], setelah proses penambahan atau pembaruan data selesai, kita redirect pengguna ke halaman `Peminjaman.php`.

Pada baris [19-29], terdapat struktur HTML untuk menampilkan form.

Pada baris [21], kita menggunakan ternary operator (`? :`) untuk menampilkan judul form sesuai dengan mode (Edit/Tambah) .

Pada baris [22], kita memiliki hidden input untuk menyimpan `<nilai $id_peminjaman>` agar dapat dikirimkan bersamaan dengan form.

Pada baris [23-26], terdapat input untuk ID member, ID buku, tanggal pinjam, dan tanggal kembali. Nilai awal masing-masing diambil dari variabel yang sesuai.

Pada baris [27], terdapat tombol submit dengan teks 'Simpan' yang akan digunakan untuk menyimpan data.

Pada baris [30], terdapat link 'Kembali' yang mengarahkan pengguna kembali ke halaman Peminjaman.php setelah selesai mengisi atau mengedit data peminjaman.

Pada baris [31], penutup tag `</body>` menandakan akhir dari bagian body HTML.

Pada baris [32], penutup tag `</html>` menandakan akhir dari dokumen HTML.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/norainaevayanti/Praktikum-Pemrograman-Web-II.git>