

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 5**



Function dan Database

Oleh:

Aditya Oktaviari

NIM. 2210817110008

**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 : Aditya Oktaviari
NIM : 2210817110008

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

Ir. Eka Setya Wijaya, S.T., M.Kom.
NIP. 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	24

DAFTAR GAMBAR

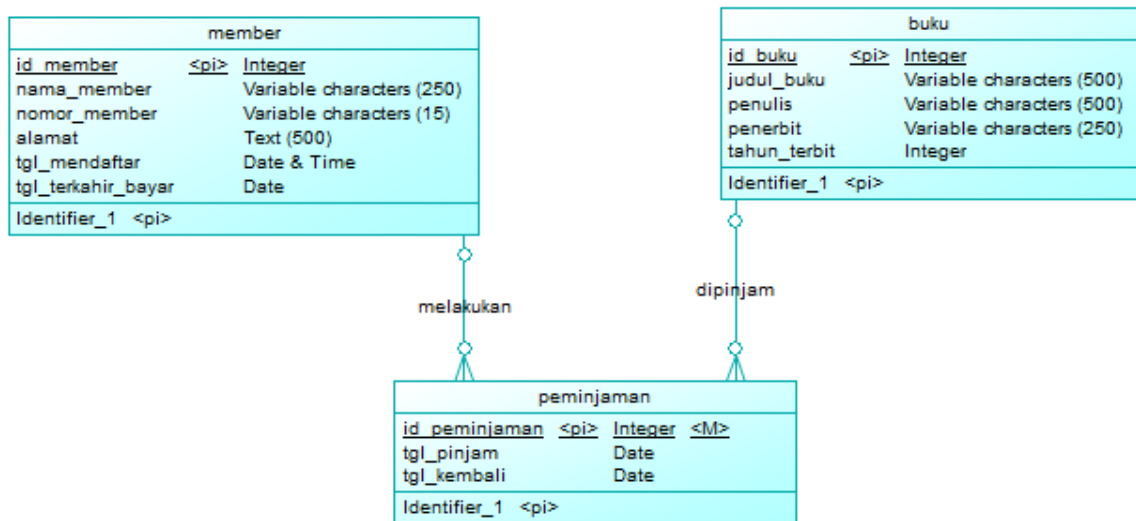
Gambar 1. Screenshot Hasil Jawaban Member.php	18
Gambar 2. Screenshot Hasil Jawaban FormMember.php	18
Gambar 3. Screenshot Hasil Jawaban Buku.php	18
Gambar 4. Screenshot Hasil Jawaban FormBuku.php	19
Gambar 5. Screenshot Hasil Jawaban Peminjaman.php.....	19
Gambar 6. Screenshot Hasil Jawaban FormPeminjaman.php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

DAFTAR TABEL

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

```

Tabel 1. Source Code Koneksi.php

```

1  <?php
2  include_once 'Koneksi.php';
3
4  class Model {
5      private $conn;
6
7      public function __construct() {
8          $database = new Koneksi();
9          $this->conn = $database->getConnection();
10     }
11
12     private function executeQuery($query, $params = [])
13     {
14         $stmt = $this->conn->prepare($query);
15         foreach ($params as $param => $value) {
16             $stmt->bindValue($param, $value);
17         }
18         return $stmt->execute();
19     }
20
21     public function getMembers() {
22         $query = "SELECT * FROM member";
23         $stmt = $this->conn->prepare($query);
24         $stmt->execute();
25         return $stmt->fetchAll(PDO::FETCH_ASSOC);
26     }
27
28     public function addMember($nama_member,
29                               $nomor_member, $tgl_mendaftar, $tgl_terakhir_bayar) {
30         $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      =
:tgl_terakhir_bayar WHERE id_member = :id_member";
39      return $this->executeQuery($query, [
40          ':id_member' => $id_member,
41          ':nama_member' => $nama_member,
42          ':nomor_member' => $nomor_member,
43          ':tgl_mendaftar' => $tgl_mendaftar,
44          ':tgl_terakhir_bayar' => $tgl_terakhir_bayar
45      ]);
46  }
47
48  public function deleteMember($id_member) {
49      $query = "DELETE FROM member WHERE id_member =
:id_member";
50      return $this->executeQuery($query, [':id_member'
=> $id_member]);
51  }
52
53  public function getBooks() {
54      $query = "SELECT * FROM buku";
55      $stmt = $this->conn->prepare($query);
56      $stmt->execute();
57      return $stmt->fetchAll(PDO::FETCH_ASSOC);
58  }
59
60  public function addBook($judul_buku, $penulis,
$tahun_terbit) {
61      $query = "INSERT INTO buku (judul_buku, penulis,
tahun_terbit)
VALUES (:judul_buku, :penulis,
:tahun_terbit)";
62      return $this->executeQuery($query, [
63          ':judul_buku' => $judul_buku,
64          ':penulis' => $penulis,
65          ':tahun_terbit' => $tahun_terbit
66      ]);
67  }
68
69  public function updateBook($id_buku, $judul_buku,
$penulis, $tahun_terbit) {
70      $query = "UPDATE buku SET judul_buku =
:judul_buku, penulis = :penulis, tahun_terbit =
:tahun_terbit WHERE id_buku = :id_buku";
71      return $this->executeQuery($query, [
72          ':id_buku' => $id_buku,
73          ':judul_buku' => $judul_buku,
74          ':penulis' => $penulis,

```

```

75         ':tahun_terbit' => $tahun_terbit
76     });
77 }
78
79     public function deleteBook($id_buku) {
80         $query = "DELETE FROM buku WHERE id_buku =
81         :id_buku";
82         return $this->executeQuery($query, ['id_buku'
83         => $id_buku]);
84     }
85
86     public function getLoans() {
87         $query = "SELECT * FROM peminjaman";
88         $stmt = $this->conn->prepare($query);
89         $stmt->execute();
90         return $stmt->fetchAll(PDO::FETCH_ASSOC);
91     }
92
93     public function addLoan($id_member, $id_buku,
94     $tgl_pinjam, $tgl_kembali) {
95         $query = "INSERT INTO peminjaman (id_member,
96     id_buku, tgl_pinjam, tgl_kembali) VALUES (:id_member,
97     :id_buku, :tgl_pinjam, :tgl_kembali)";
98         return $this->executeQuery($query, [
99     ':id_member' => $id_member,
100     ':id_buku' => $id_buku,
101     ':tgl_pinjam' => $tgl_pinjam,
102     ':tgl_kembali' => $tgl_kembali
103     ]);
104     }
105
106     public function updateLoan($id_peminjaman,
107     $id_member, $id_buku, $tgl_pinjam, $tgl_kembali) {
108         $query = "UPDATE peminjaman SET id_member =
109     :id_member, id_buku = :id_buku, tgl_pinjam =
110     :tgl_pinjam, tgl_kembali = :tgl_kembali WHERE
111     id_peminjaman = :id_peminjaman";
112         return $this->executeQuery($query, [
113     ':id_peminjaman' => $id_peminjaman,
114     ':id_member' => $id_member,
115     ':id_buku' => $id_buku,
116     ':tgl_pinjam' => $tgl_pinjam,
117     ':tgl_kembali' => $tgl_kembali
118     ]);
119     }
120
121     public function deleteLoan($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. 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. 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'],
26	\$_POST['nomor_member'],
27	\$_POST['tgl_mendaftar'], \$_POST['tgl_terakhir_bayar']);
28	} else {
29	\$model->addMember(\$_POST['nama_member'],
30	\$_POST['nomor_member'],
31	\$_POST['tgl_mendaftar'],
32	\$_POST['tgl_terakhir_bayar']);
	}
	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. 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. 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	<code>\$model->updateBook(\$_POST['id_buku'],</code>
	<code>\$_POST['judul_buku'],</code>
	<code>\$_POST['tahun_terbit']);</code>
23	<code> } else {</code>
24	<code> \$model->addBook(\$_POST['judul_buku'],</code>
	<code>\$_POST['penulis'], \$_POST['tahun_terbit']);</code>
25	<code> }</code>
26	<code> header('Location: Buku.php');</code>
27	<code>}</code>
28	<code>?></code>
29	
30	<code><!DOCTYPE html></code>
31	<code><html></code>
32	<code><head></code>
33	<code> <title>Form Buku</title></code>
34	<code></head></code>
35	<code><body></code>
36	<code> <h1><?php echo \$id_buku ? 'Edit' : 'Tambah'; ?></code>
	<code>Buku</h1></code>
37	<code> <form method="post"></code>
38	<code> <input type="hidden" name="id_buku" value="<?php</code>
	<code>echo \$id_buku; ?>"></code>
39	<code> <label>Judul Buku:</label></code>
40	<code> <input type="text" name="judul_buku" value="<?php</code>
	<code>echo \$judul_buku; ?>" required>
</code>
41	<code> <label>Penulis:</label></code>
42	<code> <input type="text" name="penulis" value="<?php</code>
	<code>echo \$penulis; ?>" required>
</code>
43	<code> <label>Tahun Terbit:</label></code>
44	<code> <input type="text" name="tahun_terbit"</code>
	<code>value="<?php echo \$tahun_terbit; ?>" required>
</code>
45	<code> <input type="submit" value="Simpan"></code>
46	<code> </form></code>
47	<code>
</code>
48	<code> Kembali</code>
49	<code></body></code>
50	<code></html></code>

Tabel 6. Source Code FormBuku.php

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

```

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. 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

```

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

Tabel 8. 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. Screenshot Hasil Jawaban Member.php

Tambah Member

Nama Member:

Nomor Member:

Tanggal Mendaftar:

Tanggal Terakhir Bayar:

[Kembali](#)

Gambar 2. Screenshot Hasil Jawaban FormMember.php

Data Buku

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

[Tambah Buku](#)

Gambar 3. Screenshot Hasil Jawaban Buku.php

Tambah Buku

Judul Buku:

Penulis:

Tahun Terbit:

[Kembali](#)

Gambar 4. 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. Screenshot Hasil Jawaban Peminjaman.php

Tambah Peminjaman

ID Member:

ID Buku:

Tanggal Pinjam:

Tanggal Kembali:

[Kembali](#)

Gambar 6. Screenshot Hasil Jawaban FormPeminjaman.php

C. Pembahasan

File Koneksi.php

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

File Model.php

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

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

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

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

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

File Member.php

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

diiterasi untuk membuat baris baru dalam tabel. Data anggota ditampilkan dalam masing-masing kolom menggunakan tag `<td>`. Di kolom terakhir, dua link disediakan untuk setiap anggota: satu untuk mengedit data anggota (dengan mengarahkan ke `FormMember.php` dan menyertakan `id_member` sebagai parameter) dan satu lagi untuk menghapus data anggota (mengarah ke `deleteMember.php` dengan `id_member` sebagai parameter, serta konfirmasi sebelum menghapus). Setelah tabel, sebuah link disediakan untuk menambah anggota baru, yang mengarah ke `FormMember.php`. Kode ini memberikan antarmuka sederhana untuk melihat, mengedit, dan menghapus data anggota, serta menambah anggota baru.

File FormMember.php

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

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

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

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

File Buku.php

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

File FormBuku.php

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

menambah buku baru atau mengedit buku yang sudah ada dengan mudah melalui formulir yang sama, tergantung pada keberadaan parameter id di URL.

File Peminjaman.php

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

File FormPeminjaman.php

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

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

Bagian HTML dari kode ini mendefinisikan struktur halaman web dengan judul "Form Peminjaman". Konten utama halaman ini berupa formulir menggunakan tag <form> dengan metode post. Terdapat input hidden untuk menyimpan nilai \$id_peminjaman. Formulir ini memungkinkan pengguna untuk memasukkan atau mengedit informasi

peminjaman, termasuk ID Member, ID Buku, Tanggal Pinjam, dan Tanggal Kembali. Nilai-nilai dari setiap input diambil dari variabel yang telah diinisialisasi sebelumnya. Sebuah tombol submit disediakan untuk mengirimkan formulir, serta link untuk kembali ke halaman Peminjaman.php setelah pengguna selesai melakukan operasi.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

[Pemrograman-Web-2/Modul5 at main · aditokta/Pemrograman-Web-2 \(github.com\)](https://github.com/aditokta/Pemrograman-Web-2/tree/main/Modul5)