

**LAPORAN PRAKTIKUM
PEMROGRAMAN II
MODUL 5**



FUNCTION DAN DATABASE

Oleh:

Siti Ainur Rahmawati NIM. 2210817220029

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN 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 : Siti Ainur Rahmawati

NIM : 2210817220029

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.....	24
C. Pembahasan	27
Tautan Git	41

DAFTAR GAMBAR

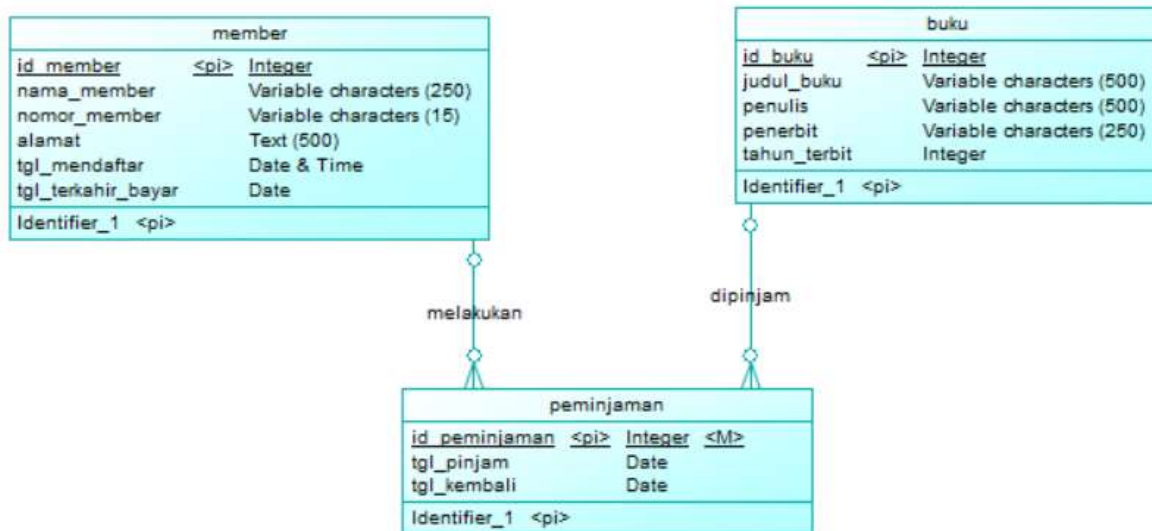
Gambar 1. Screenshot Output Soal 1 FormBuku.php	24
Gambar 2. Screenshot Output Soal 1 Buku.php.....	25
Gambar 3. Screenshot Output Soal 1 FormMember.php	25
Gambar 4. Screenshot Output Soal 1Member.php.....	26
Gambar 5. Screenshot Output Soal 1 FormPeminjaman.php	26
Gambar 6. Screenshot Output Soal 1 Peminjaman.php.....	27

DAFTAR TABEL

Tabel 1. Source Code Soal 1 Koneksi.php	7
Tabel 2. Source Code Soal 1 Model.php	8
Tabel 3. Source Code Soal 1 Member.php	11
Tabel 4. Source Code Soal 1 FormMember.php	13
Tabel 5. Source Code Soal 1 Buku.php	15
Tabel 6. Source Code Soal 1 FormBuku.php	17
Tabel 7. Source Code Soal 1 Peminjaman.php	19
Tabel 8. Source Code Soal 1 FormPeminjaman.php	22

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.

1. Adapun ketentuan pembuatannya sebagai berikut:
2. Koneksi database dibuat menjadi satu file sendiri yaitu Koneksi.php, kemudian gunakan fungsi require ketika ingin melakukan operasi ke basis data.
3. Operasi data seperti Insert, Update, Delete, Get Data dibuat menjadi fungsi sendiri masingmasing dan disimpan di dalam satu file khusus yaitu Model.php
4. Minimal terdapat file-file berikut di dalam projekContoh Output 1:

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.

A. Source Code

Tabel 1. Source Code Soal 1 Koneksi.php

1	<?php
2	try {
3	\$koneksi = new
	PDO('mysql:host=localhost;dbname=perpusdb', 'root', '',
4	array(PDO::ATTR_ERRMODE,
	PDO::ERRMODE_EXCEPTION, PDO::ATTR_PERSISTENT => true));
5	}
6	catch(PDOException \$e) {

7	echo "Koneksi atau query bermasalah: ".\$e->getMessage(). " ";
8	die();
9	}
10	?>

Tabel 2. Source Code Soal 1 Model.php

1	<?php
2	require_once 'Koneksi.php';
3	function getData(\$koneksi, \$table, \$orderBy) {
4	try {
5	\$query = "SELECT * FROM \$table ORDER BY \$orderBy DESC";
6	if (\$table == 'peminjaman') {
7	\$query = "SELECT p.id_peminjaman,
8	m.nama_member, b.judul_buku, p.tgl_pinjam, p.tgl_kembali
9	FROM peminjaman p
10	JOIN member m ON p.id_member =
11	m.id_member
12	JOIN buku b ON p.id_buku =
13	b.id_buku
14	ORDER BY \$orderBy DESC";
15	}
16	\$stmt = \$koneksi->query(\$query);
17	return \$stmt->fetchAll(PDO::FETCH_OBJ);
18	} catch (PDOException \$e) {
19	echo "Error: " . \$e->getMessage();
20	return [];
21	} finally {
22	\$koneksi = null;
23	}
24	}
25	function deleteData(\$koneksi, \$table, \$column, \$id) {
26	try {
27	\$stmt = \$koneksi->prepare("DELETE FROM \$table
28	WHERE \$column = :id");
29	\$stmt->bindParam(':id', \$id, PDO::PARAM_INT);
30	\$stmt->execute();


```

27     } catch (PDOException $e) {
28         echo "Error: " . $e->getMessage();
29     } finally {
30         $koneksi = null;
31     }
32 }

33 function updateData($koneksi, $tabel, $data,
34 $whereColumn, $whereValue) {
35     try {
36         $setClause = [];
37         foreach ($data as $key => $value) {
38             $setClause[] = "$key = :$key";
39         }
40         $setClauseStr = implode(', ', $setClause);
41         $stmt = $koneksi->prepare("UPDATE $tabel SET
42 $setClauseStr WHERE $whereColumn = :whereValue");
43         foreach ($data as $key => $value) {
44             $stmt->bindValue(":$key", $value);
45         }
46         $stmt->bindValue(':whereValue', $whereValue);
47         $stmt->execute();
48         return true;
49     } catch (PDOException $e) {
50         echo "Error: " . $e->getMessage();
51         return false;
52     } finally {
53         $koneksi = null;
54     }
55 }

56 function insertData($koneksi, $tabel, $data) {
57     try {
58         $kolom = implode(', ', array_keys($data));
59         $placeholders = ':' . implode(', ',
60 array_keys($data));
61         $stmt = $koneksi->prepare("INSERT INTO $tabel
62 ($kolom) VALUES ($placeholders)");
63         foreach ($data as $key => $value) {
64             $stmt->bindValue(":$key", $value);
65         }

```

```

62         $stmt->execute();
63         return true;
64     } catch (PDOException $e) {
65         echo "Error: " . $e->getMessage();
66         return false;
67     } finally {
68         $koneksi = null;
69     }
70 }

71 function isExist($koneksi, $column_id, $id_member,
72 $table) {
73     try {
74         $stmt = $koneksi->prepare("SELECT 1 FROM $table
75 WHERE $column_id = $id_member LIMIT 1");
76         $stmt->execute();
77         return $stmt->fetchColumn() > 0;
78     } catch (PDOException $e) {
79         echo "Error: " . $e->getMessage();
80         return false;
81     }
82 }

83 function getById($koneksi, $table, $keyColumn,
84 $keyValue) {
85     try {
86         if ($table == 'peminjaman') {
87             $query = "SELECT p.id_peminjaman,
88 m.id_member, b.id_buku, p.tgl_pinjam, p.tgl_kembali
89 FROM peminjaman p
90 JOIN member m ON p.id_member =
91 m.id_member
92 JOIN buku b ON p.id_buku =
93 b.id_buku
94 WHERE p.$keyColumn = :keyValue";
95         } else {
96             $query = "SELECT * FROM $table WHERE
97 $keyColumn = :keyValue";
98         }
99         $stmt = $koneksi->prepare($query);

```

95	\$stmt->bindParam(':keyValue', \$keyValue,
	PDO::PARAM_INT);
96	\$stmt->execute();
97	\$result = \$stmt->fetch(PDO::FETCH_OBJ);
98	return \$result;
99	} catch (PDOException \$e) {
100	echo "Error: " . \$e->getMessage();
101	return null;
102	}
103	}
104	?>

Tabel 3. Source Code Soal 1 Member.php

1	<?php
2	require_once 'model.php';
3	\$members = getData(\$koneksi, 'member', 'id_member');
4	if (\$_SERVER['REQUEST_METHOD'] == 'POST' &&
	isset(\$_POST['hapus'])) {
5	\$id = \$_POST['id'];
6	deleteData(\$koneksi, 'member', 'id_member', \$id);
7	header('Location: Member.php');
8	}
9	?>
10	<!DOCTYPE html>
11	<html>
12	<head>
13	<title>Daftar Member</title>
14	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
15	<link rel="preconnect"
	href="https://fonts.googleapis.com">
16	<link rel="preconnect"
	href="https://fonts.gstatic.com" crossorigin>
17	<link
	href="https://fonts.googleapis.com/css2?family=Spectral:i
	tal,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;

```

1,300;1,400;1,500;1,600;1,700;1,800&display=swap"
rel="stylesheet">
18     <link rel="stylesheet" href="styleslists.css">
19 </head>
20 <body>
21     <h1>Daftar Member</h1>
22     <a href="FormMember.php">Tambah Member</a>

23         <table border="1">
24             <tr>
25                 <th>ID</th>
26                 <th>Nama</th>
27                 <th>Nomor</th>
28                 <th>Alamat</th>
29                 <th>Tanggal Daftar</th>
30                 <th>Transaksi Terakhir</th>
31                 <th>Aksi</th>
32             </tr>
33             <?php if (count($members) > 0): ?>
34             <?php foreach ($members as $row): ?>
35                 <tr>
36                     <td><?=          htmlspecialchars($row-
>id_member) ?></td>
37                     <td><?=          htmlspecialchars($row-
>nama_member) ?></td>
38                     <td><?=          htmlspecialchars($row-
>nomor_member) ?></td>
39                     <td><?=          htmlspecialchars($row-
>alamat) ?></td>
40                     <td><?=          htmlspecialchars($row-
>tgl_mendaftar) ?></td>
41                     <td><?=          htmlspecialchars($row-
>tgl_terakhir_bayar) ?></td>
42                     <td>
43                         <form          action="FormMember.php"
method="post" style="display:inline;">
44                             <input type="hidden" name="id"
value="<?= htmlspecialchars($row->id_member) ?>">
45                             <button          type="submit"
name="Edit">Edit</button>
46                     </form>

```

47	<code><form method="post"></code>
45	<code> <input type="hidden" name="id"</code>
	<code>value="<?= htmlspecialchars(\$row->id_member) ?>"></code>
49	<code> <button type="submit"</code>
	<code>name="hapus" class="button edit-button">Hapus</button></code>
50	<code> </form></code>
51	<code> </td></code>
52	<code> </td></code>
53	<code> </tr></code>
54	<code> <?php endforeach; ?></code>
55	<code> <?php else: ?></code>
56	<code> <tr><td colspan="6">Tidak ada member yang</code>
	<code>terdaftar</td></tr></code>
57	<code> <?php endif; ?></code>
58	<code> </table></code>
59	<code></body></code>
60	<code></html></code>

Tabel 4. Source Code Soal 1 FormMember.php

1	<code><?php</code>
2	<code> require 'Model.php';</code>
3	<code> \$id_member = isset(\$_POST['id']) ? \$_POST['id'] : null;</code>
	<code> \$data = null;</code>
4	<code> if (\$id_member) {</code>
5	<code> \$data = getById(\$koneksi, 'member', 'id_member',</code>
6	<code> \$id_member);</code>
7	<code> }</code>
8	
9	<code> if ((\$_SERVER['REQUEST_METHOD'] == 'POST' &&</code>
	<code> isset(\$_POST['submit']))) {</code>
10	<code> \$arr_data = [</code>
11	<code> 'nama_member' => \$_POST['nama'],</code>
12	<code> 'nomor_member' => \$_POST['nomor'],</code>
13	<code> 'alamat' => \$_POST['alamat'],</code>
14	<code> 'tgl_mendaftar' => date('Y-m-d H:i:s'),</code>
	<code> 'tgl_terakhir_bayar' => \$_POST['tgl_terakhir']</code>
15	<code>];</code>
	<code> \$whereColumn = 'id_member';</code>

```

16     $whereValue = $_POST['id_member'];
17     if (isExist($koneksi, 'id_member',
$_POST['id_member'], 'member')) {
        updateData($koneksi, 'member', $arr_data,
$whereColumn, $whereValue);
    } else {
        insertData($koneksi, 'member', $arr_data);
    }
18
19     header('Location: Member.php');
20     exit();
21 }
22 ?>

23 <!DOCTYPE html>
24 <html>
25 <head>
26     <title>Form Member</title>
27     <meta charset="UTF-8">
28     <meta name="viewport" content="width=device-width,
29 initial-scale=1.0">
30     <title>Form Member</title>
31     <link rel="preconnect"
32 href="https://fonts.googleapis.com">
33     <link rel="preconnect"
34 href="https://fonts.gstatic.com" crossorigin>
35     <link
36 href="https://fonts.googleapis.com/css2?family=Spectral:i
tal,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;
37 1,300;1,400;1,500;1,600;1,700;1,800&display=swap"
rel="stylesheet">
38     <link rel="stylesheet" href="styles.css">
39 </head>
40 <body>
    <h1>Form Member</h1>
    <form method="post">
        <input type="hidden" name="id_member" value="<?=$data ? htmlspecialchars($data->id_member) : '' ?>"><br>
        <Label for="nama">Nama</Label>
41
42
43

```

44	<pre> <input type="text" name="nama" id="nama" max=250 required value="<?= \$data ? htmlspecialchars(\$data- >nama_member) : ' ' ?>">
 <Label for="nomor">Nomor Telepon</Label> <input type="num" name="nomor" id="nomor" max=15 required value="<?= \$data ? htmlspecialchars(\$data- >nomor_member) : ' ' ?>">
 <Label for="alamat">Alamat</Label> <input type="text" name="alamat" id="alamat" max=500 required value="<?= \$data ? htmlspecialchars(\$data->alamat) : ' ' ?>">
 <Label for="alamat">Tanggal Terakhir Transaksi</Label> <input type="date" name="tgl_terakhir" id="tgl_terakhir" max=500 required value="<?= \$data ? htmlspecialchars(\$data->tgl_terakhir_bayar) : ' ' ?>">
 <button type="submit" name="submit">Simpan</button> </form> </body> </html> </pre>
57	
58	
59	
60	

Tabel 5. Source Code Soal 1 Buku.php

1	<pre><?php</pre>
2	<pre>require_once 'model.php';</pre>
3	<pre>\$books = getData(\$koneksi, 'buku', 'id_buku');</pre>
4	<pre>if (\$_SERVER['REQUEST_METHOD'] == 'POST' &&</pre>
5	<pre>isset(\$_POST['hapus'])) {</pre>
6	<pre> \$id = \$_POST['id'];</pre>
7	<pre> deleteData(\$koneksi, 'buku', 'id_buku', \$id);</pre>
8	<pre> header('Location: Buku.php');</pre>
9	<pre>}</pre>
10	<pre>?></pre>
11	<pre><!DOCTYPE html></pre>
12	<pre><html></pre>

```

12 <head>
13     <title>Daftar Buku</title>
14     <meta name="viewport" content="width=device-width,
initial-scale=1.0">
15     <link rel="preconnect"
href="https://fonts.googleapis.com">
16     <link rel="preconnect"
href="https://fonts.gstatic.com" crossorigin>
17     <link
href="https://fonts.googleapis.com/css2?family=Spectral:i
tal,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;
1,300;1,400;1,500;1,600;1,700;1,800&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="styleslists.css">
18 </head>
19 <body>
20     <h1>Daftar Buku</h1>
21     <a href="FormBuku.php">Tambah Buku</a>
22
    <table border="1">
23         <tr>
24             <th>ID</th>
25             <th>Judul</th>
26             <th>Penerbit</th>
27             <th>Penulis</th>
28             <th>Tahun Terbit</th>
29             <th>Aksi</th>
30         </tr>
31         <?php if (count($books) > 0): ?>
32             <?php foreach ($books as $row): ?>
33                 <tr>
34                     <td><?=          htmlspecialchars($row-
35 >id_buku) ?></td>
36                     <td><?=          htmlspecialchars($row-
37 >judul_buku) ?></td>
38                     <td><?=          htmlspecialchars($row-
39 >penerbit) ?></td>
40                     <td><?=          htmlspecialchars($row-
41 >penulis) ?></td>
42                     <td><?=          htmlspecialchars($row-
43 >tahun_terbit) ?></td>

```


40	<pre> <td> <form action="FormBuku.php" 41 method="post"> <input type="hidden" name="id" 42 value="<?= htmlspecialchars(\$row->id_buku) ?>"> 43 <button type="submit" name="Edit">Edit</button> 44 </form> <form method="post"> 45 <input type="hidden" name="id" value="<?= htmlspecialchars(\$row->id_buku) ?>"> 46 <button type="submit" 47 name="hapus" class="button edit-button">Hapus</button> 45 </form> </td> 49 </tr> <?php endforeach; ?> 50 <?php else: ?> 51 <tr><td colspan="6">Tidak ada buku yang 52 terdaftar</td></tr> 53 <?php endif; ?> 54 </table> 55 </body> 56 </html> 57 58 59 60 </pre>
----	--

Tabel 6. Source Code Soal 1 FormBuku.php

1	<pre><?php</pre>
2	<pre>require 'Model.php';</pre>
3	<pre> \$id_buku = isset(\$_POST['id']) ? \$_POST['id'] : null; 4 \$data = null; 5 if (\$id_buku) { 6 echo \$id_buku; </pre>

```

7         $data = getById($koneksi, 'buku', 'id_buku',
8 $id_buku);
9     }

10     if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
11 isset($_POST['submit'])) {
12         $arr_data = [
13             'judul_buku' => $_POST['judul'],
14             'penulis' => $_POST['penulis'],
15             'penerbit' => $_POST['penerbit'],
16             'tahun_terbit' => $_POST['tahun']
17         ];

18         $whereColumn = 'id_buku';
19         $whereValue = $_POST['id_buku'];
20         if (isExist($koneksi, 'id_buku',
21 $_POST['id_buku'], 'buku')) {
22             updateData($koneksi, 'buku', $arr_data,
23 $whereColumn, $whereValue);
24         } else {
25             insertData($koneksi, 'buku', $arr_data);
26         }
27         header('Location: Buku.php');
28         exit();
29     }
30     ?>

31 <!DOCTYPE html>
32 <html>
33 <head>
34     <title>Form Buku</title>
35     <link rel="preconnect"
36 href="https://fonts.googleapis.com">
37     <link rel="preconnect"
38 href="https://fonts.gstatic.com" crossorigin>
39     <link
40 href="https://fonts.googleapis.com/css2?family=Spectral:i
41 tal,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;
42 1,300;1,400;1,500;1,600;1,700;1,800&display=swap"
43 rel="stylesheet">
44     <link rel="stylesheet" href="styles.css">

```

37	</head>
	<body>
38	<h1>Form Buku</h1>
	<form method="post">
39	<input type="hidden" name="id_buku" value="<?= \$data
	? htmlspecialchars(\$data->id_buku) : ' ' ?>">
40	<Label for="judul">Judul</Label>
	<input type="text" name="judul" id="judul" max=500
41	required value="<?= \$data ? htmlspecialchars(\$data-
	>judul_buku) : ' ' ?>">
42	<Label for="penulis">Penulis</Label>
43	<input type="text" name="penulis" id="penulis"
	max=500 required value="<?= \$data ?
44	htmlspecialchars(\$data->penulis) : ' ' ?>">
	<Label for="penerbit">Penerbit</Label>
45	<input type="text" name="penerbit" id="penerbit"
	max=250 required value="<?= \$data ?
46	htmlspecialchars(\$data->penerbit) : ' ' ?>">
47	<Label for="tahun">Tahun Terbit</Label>
45	<input type="num" name="tahun" id="tahun" required
	value="<?= \$data ? htmlspecialchars(\$data->tahun_terbit) :
49	' ' ?>">
	<button type="submit"
50	name="submit">Simpan</button>
51	</form>
52	</body>
53	</html>
54	
55	
56	
57	
58	
59	
60	

Tabel 7. Source Code Soal 1 Peminjaman.php

1	<?php
2	require_once 'model.php';

```

3  $lends = getData($koneksi, 'peminjaman', 'id_peminjaman');

4  if      ($_SERVER['REQUEST_METHOD']      ==      'POST'      &&
isset($_POST['hapus'])) {
5      $id = $_POST['id'];
6      deleteData($koneksi,  'peminjaman',  'id_peminjaman',
7  $id);
8      header('Location: Peminjaman.php');
9  }
10  ?>
11  <!DOCTYPE html>
12  <html>
13  <head>
14      <title>Daftar Peminjaman</title>
15      <meta  name="viewport"  content="width=device-width,
initial-scale=1.0">
16      <link                                rel="preconnect"
href="https://fonts.googleapis.com">
17      <link                                rel="preconnect"
href="https://fonts.gstatic.com" crossorigin>
18      <link
href="https://fonts.googleapis.com/css2?family=Spectral:i
tal,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;
1,300;1,400;1,500;1,600;1,700;1,800&display=swap"
rel="stylesheet">
19      <link rel="stylesheet" href="styleslists.css">
20  </head>
21  <body>
22      <h1>Daftar Peminjaman</h1>
23      <a href="FormPeminjaman.php">Tambah Peminjaman</a>
24
25      <table border="1">
26          <tr>
27              <th>ID</th>
28              <th>Nama</th>
29              <th>Judul Buku</th>
30              <th>Tanggal Pinjam</th>
31              <th>Tanggal Kembali</th>
32              <th>Aksi</th>
33          </tr>

```

```

32         <?php if (count($lends) > 0): ?>
33         <?php foreach ($lends as $row): ?>
34             <tr>
35                 <td><?=          htmlspecialchars($row-
36 >id_peminjaman) ?></td>
37                 <td><?=          htmlspecialchars($row-
37 >nama_member) ?></td>
38                 <td><?=          htmlspecialchars($row-
38 >judul_buku) ?></td>
39                 <td><?=          htmlspecialchars($row-
39 >tgl_pinjam) ?></td>
40                 <td><?=          htmlspecialchars($row-
40 >tgl_kembali) ?></td>
41                 <td>
42                     <form action="FormPeminjaman.php"
method="post">
43                         <input type="hidden" name="id"
value="<?= htmlspecialchars($row->id_peminjaman) ?>">
44                         <button          type="submit"
name="Edit">Edit</button>
45                     </form>
46                     <br>
47                     <form method="post">
48                         <input type="hidden" name="id"
value="<?= htmlspecialchars($row->id_peminjaman) ?>">
49                         <button          type="submit"
name="hapus" class="button edit-button">Hapus</button>
50                     </form>
51                 </td>
52             </tr>
53         <?php endforeach; ?>
54         <?php else: ?>
55             <tr><td colspan="6">Tidak ada peminjaman
yang terdaftar</td></tr>
56         <?php endif; ?>
57     </table>
58 </body>
59 </html>
60

```

Tabel 8. Source Code Soal 1 FormPeminjaman.php

1	<?php
2	require 'model.php';
3	\$id_peminjaman = isset(\$_POST['id']) ? \$_POST['id'] :
4	null;
5	\$data = null;
6	if (\$id_peminjaman) {
7	\$data = getById(\$koneksi, 'peminjaman',
8	'id_peminjaman', \$id_peminjaman);
9	}
10	if (\$_SERVER['REQUEST_METHOD'] == 'POST' &&
11	isset(\$_POST['submit'])) {
12	\$arr_data = [
13	'id_member' => \$_POST['id_member'],
14	'id_buku' => \$_POST['id_buku'],
15	'tgl_pinjam' => \$_POST['tgl_pinjam'],
16	'tgl_kembali' => \$_POST['tgl_kembali']
17];
18	\$whereColumn = 'id_peminjaman';
19	\$whereValue = \$_POST['id_peminjaman'];
20	if (isExist(\$koneksi, 'id_member',
21	\$_POST['id_member'], 'member') && isExist(\$koneksi,
22	'id_buku', \$_POST['id_buku'], 'buku')) {
23	if (isExist(\$koneksi, 'id_peminjaman',
24	\$_POST['id_peminjaman'], 'peminjaman')) {
25	updateData(\$koneksi, 'peminjaman',
26	\$arr_data, \$whereColumn, \$whereValue);
27	} else {
28	insertData(\$koneksi, 'peminjaman',
29	\$arr_data);
30	}
31	header('Location: Peminjaman.php');
32	exit();
33	} else {


```

26         echo "Id Member atau Id Buku tidak terdaftar";
27     }
28 }
29 ?>
30
31 <!DOCTYPE html>
32 <html lang="en">
33 <head>
34     <meta charset="UTF-8">
35     <meta name="viewport" content="width=device-width,
36 initial-scale=1.0">
37     <link rel="preconnect"
href="https://fonts.googleapis.com">
38     <link rel="preconnect"
href="https://fonts.gstatic.com" crossorigin>
39     <link
href="https://fonts.googleapis.com/css2?family=Spectral:i
40 tal,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;
1,300;1,400;1,500;1,600;1,700;1,800&display=swap"
41 rel="stylesheet">
42     <link rel="stylesheet" href="styles.css">
43 </head>
44 <body>
45     <h1>Form Peminjaman</h1>
46     <form method="post">
47         <input type="hidden" name="id_peminjaman"
48 value="<?= $data ? htmlspecialchars($data->id_peminjaman)
49 : '' ?>">
50         <label for="id_member">ID Member</label>
51         <input type="text" name="id_member" id="id_member"
52 value="<?= $data ? htmlspecialchars($data->id_member) : ''
53 ?>" required><br>
54         <label for="id_buku">ID Buku</label>
55         <input type="text" name="id_buku" id="id_buku"
56 value="<?= $data ? htmlspecialchars($data->id_buku) : ''
57 ?>" required><br>
58         <label for="tgl_pinjam">Tanggal Pinjam</label>
59         <input type="date" name="tgl_pinjam"
60 id="tgl_pinjam" value="<?= $data ? htmlspecialchars($data-
61 >tgl_pinjam) : '' ?>" required><br>

```

56	<pre> <label for="tgl_kembali">Tanggal Kembali</label> <input type="date" name="tgl_kembali" 57 id="tgl_kembali" value="<?= \$data ? 58 htmlspecialchars(\$data->tgl_kembali) : ' ?>" 59 required>
 60 <button type="submit" name="submit">Simpan</button> </form> </body> </html> </pre>
----	---

B. Output Program



The screenshot shows a web application with a dark blue background. At the top center, the text "Form Buku" is displayed in a white, serif font. Below this, a white rectangular form is centered. The form contains four input fields, each with a label above it: "Judul", "Penulis", "Penerbit", and "Tahun Terbit". The "Tahun Terbit" field is a smaller, single-line input. At the bottom of the form is a red button with the word "Simpan" in white text.

Gambar 1. Screenshot Output Soal 1 FormBuku.php

Daftar Buku

Tambah Buku

ID	Judul	Penerbit	Penulis	Tahun Terbit	Aksi
11	Jodoh Saya Tuhan yang Tau	Rumah Teranga	Kambing Hitam	2017	Edit Hapus
2	Jodoh Ain Siapa Geraagan?	Rumah Ain	Ain-Cantik	1997	Edit Hapus
1	Ayam Goreng Wenak	Mizan	Ain	2007	Edit Hapus

Gambar 2. Screenshot Output Soal 1 Buku.php

Form Member

Nama

Nomor Telepon

Alamat

Tanggal Terakhir Transaksi

Simpan

Gambar 3. Screenshot Output Soal 1 FormMember.php

Daftar Member

Tambah Member

ID	Nama	Nomor	Alamat	Tanggal Daftar	Transaksi Terakhir	Aksi
3	Nini	098767	jalan negara	2024-05-22 16:41:46	2024-05-22	<div>Edit</div> <div>Hapus</div>
2	Siti	123	jalan surya	2024-05-22 16:41:53	2024-05-07	<div>Edit</div> <div>Hapus</div>
1	Jane Doe	0897979797997	123 Main St	2024-05-22 16:38:34	2024-01-01	<div>Edit</div> <div>Hapus</div>

Gambar 4. Screenshot Output Soal 1Member.php

Form Peminjaman

ID Member

ID Buku

Tanggal Pinjam

mm/dd/yyyy

Tanggal Kembali

mm/dd/yyyy

Simpan

Gambar 5. Screenshot Output Soal 1 FormPeminjaman.php

Daftar Peminjaman					
<div>Tambah Peminjaman</div>					
ID	Nama	Judul Buku	Tanggal Pinjam	Tanggal Kembali	Aksi
21	Jane Doe	Jedoh: Aio Siapa Gerakan?	2024-05-09	2024-05-04	<div>Edit</div> <div>Hapus</div>
19	Jane Doe	Ayam Goreng Wenak	2024-05-08	2024-05-10	<div>Edit</div> <div>Hapus</div>

Gambar 6. Screenshot Output Soal 1 Peminjaman.php

C. Pembahasan

Koneksi.php

Baris 1: <?php - Memulai kode PHP

Baris 2-3: try { - Blok untuk mencoba koneksi

Baris 4: \$koneksi = new PDO(...) - Membuat objek koneksi PDO

Baris 4 (per bagian):

'mysql:host=localhost;dbname=perpusdb' - Informasi koneksi database

'root': Username untuk koneksi

": Password untuk koneksi (kosong)

array(...): Opsi PDO:

PDO::ATTR_ERRMODE: Mode error (exception)

PDO::ATTR_PERSISTENT: Koneksi persisten

Baris 5: } - Penutup blok try

Baris 6-7: catch(PDOException \$e) { - Blok untuk menangkap kesalahan koneksi

Baris 8: echo "Koneksi atau query bermasalah:..." - Mencetak pesan error

Baris 8 (per bagian):

\$e->getMessage(): Mendapatkan pesan detail error

Baris 9 (komentar): // die(); - Mematikan program jika error (tidak aktif)

Baris 10: ?> - Mengakhiri kode PHP

Model.php

Baris 1: <?php - Memulai kode PHP

Baris 2: require_once 'Koneksi.php'; - Memuat file 'Koneksi.php' yang diasumsikan berisi informasi koneksi database.

Baris 4-22: Fungsi getData(\$koneksi, \$table, \$orderBy)

Mendapatkan semua data dari tabel dengan order by tertentu.

Baris 4-5: Blok try untuk mencoba eksekusi query.

Baris 6: Membangun query SELECT * dasar.

Baris 7-13: Menyesuaikan query untuk tabel 'peminjaman' (join member dan buku).

Baris 14: Eksekusi query menggunakan \$koneksi->query().

Baris 15: Mengembalikan hasil query sebagai array objek.

Baris 16-18: Blok catch untuk menangani kesalahan query.

Baris 19: Blok finally untuk menutup koneksi (jika ada).

Baris 24-34: Fungsi deleteData(\$koneksi, \$table, \$column, \$id)

Menghapus data dari tabel berdasarkan kolom dan id tertentu.

Baris 24-25: Blok try untuk mencoba eksekusi query.

Baris 26: Membangun query DELETE dengan prepared statement.

Baris 27: Memasang parameter id ke statement.

Baris 28: Eksekusi query menggunakan \$stmt->execute().

Baris 29-31: Blok catch untuk menangani kesalahan query.

Baris 32: Blok finally untuk menutup koneksi (jika ada).

Baris 36-60: Fungsi updateData(\$koneksi, \$tabel, \$data, \$whereColumn, \$whereValue)

Memperbarui data di tabel berdasarkan data baru, kolom kondisi, dan nilai kondisi.

Baris 36-37: Blok try untuk mencoba eksekusi query.

Baris 38-40: Membangun string untuk set clause (kolom dan nilai yang akan diubah).

Baris 41: Membangun prepared statement untuk update.

Baris 42-44: Memasang parameter untuk setiap data yang akan diubah.

Baris 45: Memasang parameter untuk nilai kondisi.

Baris 46: Eksekusi query menggunakan \$stmt->execute().

Baris 47: Mengembalikan True jika berhasil.

Baris 48-50: Blok catch untuk menangani kesalahan query.

Baris 51: Mengembalikan False jika gagal.

Baris 52: Blok finally untuk menutup koneksi (jika ada).

Baris 62-82: Fungsi insertData(\$koneksi, \$tabel, \$data)

Memasukkan data baru ke dalam tabel.

Baris 62-63: Blok try untuk mencoba eksekusi query.

Baris 64-65: Membangun string untuk kolom dan placeholder.

Baris 66: Membangun prepared statement untuk insert.

Baris 67-69: Memasang parameter untuk setiap data yang akan dimasukkan.

Baris 70: Eksekusi query menggunakan `$stmt->execute()`.

Baris 71: Mengembalikan True jika berhasil.

Baris 72-74: Blok catch untuk menangani kesalahan query.

Baris 75: Mengembalikan False jika gagal.

Baris 76: Blok finally untuk menutup koneksi (jika ada).

Baris 84-94: Fungsi `isExist($koneksi, $column_id, $id_member, $table)`

Mengecek keberadaan data di tabel berdasarkan kolom ID dan nilai ID.

Baris 84-85: Blok try untuk mencoba eksekusi query.

Baris 86: Membangun prepared statement untuk mengecek keberadaan data.

Baris 87: Eksekusi query menggunakan `$stmt->execute()`.

Baris 88: Mengembalikan True jika data ada (jumlah baris > 0).

Baris 89-91: Blok catch untuk menangani kesalahan query.

Baris 92: Mengembalikan False jika data tidak ada.

****Baris 93: Blok finally untuk menutup koneksi (jika ada).**

Baris 95-113: Fungsi `getById($koneksi, $table, $keyColumn, $keyValue)`

Mendapatkan data dari tabel berdasarkan kolom kunci dan nilai kunci.

Baris 95-96: Blok try untuk mencoba eksekusi query.

Baris 97-101: Membangun query SELECT dengan prepared statement (menyesuaikan untuk tabel 'peminjaman' jika diperlukan).

Baris 102: Memasang parameter `keyValue` ke statement.

Baris 103: Eksekusi query menggunakan `$stmt->execute()`.

Baris 104: Mengambil hasil query sebagai objek.

Baris 105: Mengembalikan objek data jika ada.

Baris 106-108: Blok catch untuk menangani kesalahan query.

Baris 109: Mengembalikan null jika data tidak ada.

Baris 110: Blok finally untuk menutup koneksi (jika ada).

Baris 112: `?>` - Menutup kode PHP

Member.php

Baris 1-18:

Baris 1: `require_once 'model.php'`: Memuat file model.php yang diasumsikan berisi fungsi untuk mengakses data (seperti `getData`).

Baris 2: `$members = getData($koneksi, 'member', 'id_member')`: Mengambil data member dari tabel 'member' dengan order by 'id_member' dan menyimpannya di variable `$members`.

Baris 3-9: Blok if: Memeriksa apakah request method adalah POST dan apakah ada submit button bernama 'hapus'.

Jika kondisi terpenuhi:

Baris 4: `$id = $_POST['id']`: Mengambil nilai dari input hidden 'id' yang dikirimkan dari form.

Baris 5: `deleteData($koneksi, 'member', 'id_member', $id)`: Menghapus data member dengan id tersebut menggunakan fungsi `deleteData`.

Baris 6-8: `header('Location: Member.php')`: Merespons dengan redirect ke halaman 'Member.php' untuk menghindari refresh dan resubmit form.

Baris 19-47:

Baris 19-27: Bagian `<!DOCTYPE html>` hingga `<title>Daftar Member</title>`: Struktur dasar dan judul halaman HTML.

Baris 28: `Tambah Member`: Link untuk membuka halaman form tambah member.

Baris 29-42: Tabel dengan header: ID, Nama, Nomor, Alamat, Tanggal Daftar, Transaksi Terakhir, Aksi.

Baris 43-46: Blok if (count(\$members) > 0): Memeriksa apakah ada data member.

Baris 44-45: Looping foreach: Menampilkan setiap baris data member.

htmlspecialchars: Mencegah script injection pada data yang ditampilkan.

Baris 47: Form edit member dan form hapus member (sudah dijelaskan sebelumnya).

Penjelasan Baris Kode yang Belum Dijelaskan:

Baris 48: <link rel="stylesheet" href="styleslists.css">: Menambahkan file CSS eksternal bernama 'styleslists.css' untuk menata tampilan tabel dan elemen lain di halaman.

Baris 49: Bagian ini merupakan struktur penutup dari halaman HTML

FormMember.php

require 'Model.php';: Memuat file Model.php yang mungkin berisi fungsi-fungsi untuk database.

\$id_member = isset(\$_POST['id']) ? \$_POST['id'] : null;: Mengambil ID member dari data POST jika tersedia, jika tidak, menetapkan null.

\$data = null;: Menginisialisasi variabel \$data dengan null.

if (\$id_member) { ... }: Jika ID member tersedia, panggil fungsi getById untuk mendapatkan data member dari database.

if (\$_SERVER['REQUEST_METHOD'] == 'POST' && isset(\$_POST['submit'])) { ... }:
Jika permintaan adalah POST dan tombol submit ditekan, lakukan operasi berikut:

Mengumpulkan data dari form dan menyimpannya dalam array \$arr_data.

Mengatur kolom dan nilai untuk kondisi WHERE pada query SQL.

Memeriksa apakah ID member sudah ada di database dengan isExist.

Jika ada, panggil fungsi updateData untuk memperbarui data.

Jika tidak ada, panggil fungsi insertData untuk menambahkan data baru.

Redirect ke halaman Member.php setelah operasi selesai.

<!DOCTYPE html>: Mendefinisikan dokumen ini sebagai HTML5.

<html>: Tag pembuka untuk dokumen HTML.

<head>: Bagian kepala dari dokumen HTML yang berisi metadata.

<title>Form Member</title>: Menetapkan judul dokumen yang akan ditampilkan di tab browser.

<meta charset="UTF-8">: Menetapkan karakter encoding untuk dokumen ini.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Mengatur viewport agar tampilan lebih responsif di berbagai perangkat.

<link rel="preconnect" href="https://fonts.googleapis.com">: Preconnect ke sumber Google Fonts untuk meningkatkan kinerja.

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>: Preconnect ke Google Fonts dengan CORS untuk meningkatkan kinerja.

<link href="https://fonts.googleapis.com/css2?family=Spectral:...&display=swap" rel="stylesheet">: Memuat font Spectral dari Google Fonts.

<link rel="stylesheet" href="styles.css">: Menautkan stylesheet eksternal styles.css.

<body>: Tag pembuka untuk bagian tubuh dokumen.

<h1>Form Member</h1>: Menampilkan judul halaman "Form Member".

<form method="post">: Membuka form HTML yang mengirim data dengan metode POST.

<input type="hidden" name="id_member" value="<?= \$data ? htmlspecialchars(\$data->id_member) : " ?">
: Input tersembunyi untuk menyimpan ID member.

<label for="nama">Nama</label>: Label untuk input nama.

<input type="text" name="nama" id="nama" max=250 required value="<?= \$data ? htmlspecialchars(\$data->nama_member) : " ?">
: Input teks untuk nama member.

<label for="nomor">Nomor Telepon</label>: Label untuk input nomor telepon.

<input type="num" name="nomor" id="nomor" max=15 required value="<?= \$data ? htmlspecialchars(\$data->nomor_member) : " ?">
: Input teks untuk nomor telepon member.

<label for="alamat">Alamat</label>: Label untuk input alamat.

<input type="text" name="alamat" id="alamat" max=500 required value="<?= \$data ? htmlspecialchars(\$data->alamat) : " ?">
: Input teks untuk alamat member.

<label for="alamat">Tanggal Terakhir Transaksi</label>: Label untuk input tanggal terakhir transaksi.

<input type="date" name="tgl_terakhir" id="tgl_terakhir" required value="<?= \$data ? htmlspecialchars(\$data->tgl_terakhir_bayar) : " ?">
: Input tanggal untuk tanggal terakhir transaksi.

<button type="submit" name="submit">Simpan</button>: Tombol submit untuk mengirim form.

</form>: Menutup form.

`</body>`: Menutup bagian tubuh dokumen.

`</html>`: Menutup dokumen HTML.

Buku.php

`require_once 'model.php';`: Memuat file model.php yang mungkin berisi fungsi-fungsi untuk berinteraksi dengan database.

`$books = getData($koneksi, 'buku', 'id_buku');`: Mengambil semua data buku dari tabel buku menggunakan fungsi `getData`.

`if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['hapus'])) { ... }`: Memeriksa apakah permintaan adalah POST dan apakah tombol hapus ditekan.

`$id = $_POST['id'];`: Mengambil ID buku dari data POST.

`deleteData($koneksi, 'buku', 'id_buku', $id);`: Memanggil fungsi `deleteData` untuk menghapus buku berdasarkan ID.

`header('Location: Buku.php');`: Melakukan redirect ke halaman Buku.php.

`exit();`: Menghentikan eksekusi skrip setelah redirect.

`<!DOCTYPE html>`: Mendefinisikan dokumen ini sebagai HTML5.

`<html>`: Tag pembuka untuk dokumen HTML.

`<head>`: Bagian kepala dari dokumen HTML yang berisi metadata.

`<title>Daftar Buku</title>`: Menetapkan judul dokumen yang akan ditampilkan di tab browser.

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Mengatur viewport agar tampilan lebih responsif di berbagai perangkat.

`<link rel="preconnect" href="https://fonts.googleapis.com">`: Preconnect ke sumber Google Fonts untuk meningkatkan kinerja.

`<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>`: Preconnect ke Google Fonts dengan CORS untuk meningkatkan kinerja.

`<link href="https://fonts.googleapis.com/css2?family=Spectral:...&display=swap" rel="stylesheet">`: Memuat font Spectral dari Google Fonts.

`<link rel="stylesheet" href="styleslists.css">`: Menautkan stylesheet eksternal styleslists.css.

`<body>`: Tag pembuka untuk bagian tubuh dokumen.

`<h1>Daftar Buku</h1>`: Menampilkan judul halaman "Daftar Buku".

`Tambah Buku`: Link untuk menuju halaman form penambahan buku.

`<table border="1">`: Membuka tabel HTML dengan border.

`<tr><th>...</th></tr>`: Baris pertama tabel yang berisi header kolom.

`<?php if (count($books) > 0): ?> ... <?php else: ?> ... <?php endif; ?>`: Struktur kontrol untuk memeriksa apakah ada buku yang terdaftar.

`<?php foreach ($books as $row): ?> ... <?php endforeach; ?>`: Loop melalui setiap buku dan menampilkan data dalam baris tabel.

`<tr> ... </tr>`: Menampilkan data setiap buku dalam baris tabel.

`<td><?= htmlspecialchars($row->...)?></td>`: Menampilkan data buku dan menggunakan `htmlspecialchars` untuk menghindari XSS.

`<form ...>`: Form untuk tindakan edit dan hapus buku.

`<input type="hidden" name="id" value="<?= htmlspecialchars($row->id_buku) ?>">`: Input tersembunyi untuk menyimpan ID buku.

`<button type="submit" name="Edit">Edit</button>`: Tombol untuk mengedit buku.

`<button type="submit" name="hapus" class="button edit-button">Hapus</button>`: Tombol untuk menghapus buku.

`<?php else: ?>`: Jika tidak ada buku yang terdaftar, menampilkan pesan "Tidak ada buku yang terdaftar".

`</table>`: Menutup tabel.

`</body>`: Menutup bagian tubuh dokumen.

`</html>`: Menutup dokumen HTML.

FormBuku.php

`require 'Model.php';`: Memuat file `Model.php` yang mungkin berisi fungsi-fungsi untuk berinteraksi dengan database.

`$id_buku = isset($_POST['id']) ? $_POST['id'] : null;`: Memeriksa apakah ada id yang dikirim melalui POST dan menetapkan nilainya ke `$id_buku`, atau null jika tidak ada.

`$data = null;`: Menginisialisasi variabel `$data` dengan nilai null.

`if ($id_buku) { ... }`: Jika `$id_buku` tidak null, mengeksekusi blok kode di dalamnya.

`echo $id_buku;`: Menampilkan nilai `$id_buku` (mungkin untuk tujuan debugging).

`$data = getById($koneksi, 'buku', 'id_buku', $id_buku);`: Mengambil data buku dari database berdasarkan id_buku.

`if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['submit'])) { ... }`: Memeriksa apakah metode permintaan adalah POST dan apakah tombol submit ditekan.

`$arr_data = [...];`: Membuat array `$arr_data` dengan data yang diambil dari POST.

`$whereColumn = 'id_buku';`: Menetapkan nama kolom ID untuk kondisi WHERE.

`$whereValue = $_POST['id_buku'];`: Menetapkan nilai ID untuk kondisi WHERE.

`if (isExist($koneksi, 'id_buku', $_POST['id_buku'], 'buku')) { ... } else { ... }`: Memeriksa apakah buku dengan ID tersebut sudah ada.

`updateData($koneksi, 'buku', $arr_data, $whereColumn, $whereValue);`: Memperbarui data buku jika sudah ada.

`insertData($koneksi, 'buku', $arr_data);`: Menambahkan data buku baru jika belum ada.

`header('Location: Buku.php');`: Redirect ke halaman Buku.php setelah operasi berhasil.

`exit();`: Menghentikan eksekusi skrip lebih lanjut setelah header.

`<!DOCTYPE html>`: Mendefinisikan dokumen ini sebagai HTML5.

`<html>`: Tag pembuka untuk dokumen HTML.

`<head>`: Bagian kepala dari dokumen HTML yang berisi metadata.

`<title>Form Buku</title>`: Menetapkan judul dokumen yang akan ditampilkan di tab browser.

`<link rel="preconnect" href="https://fonts.googleapis.com">`: Preconnect ke sumber Google Fonts untuk meningkatkan kinerja.

`<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>`: Preconnect ke Google Fonts dengan CORS untuk meningkatkan kinerja.

`<link href="https://fonts.googleapis.com/css2?family=Spectral:ital,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;1,300;1,400;1,500;1,600;1,700;1,800&display=swap" rel="stylesheet">`: Memuat font Spectral dari Google Fonts.

`<link rel="stylesheet" href="styles.css">`: Menautkan stylesheet eksternal styles.css.

`<body>`: Tag pembuka untuk bagian tubuh dokumen.

`<h1>Form Buku</h1>`: Menampilkan judul halaman "Form Buku".

`<form method="post">`: Membuka form HTML dengan metode POST.

<input type="hidden" name="id_buku" value="<?= \$data ? htmlspecialchars(\$data->id_buku) : " ?>">: Input tersembunyi untuk menyimpan ID buku jika ada (untuk pengeditan).

<Label for="judul">Judul</Label>: Label untuk input judul buku.

<input type="text" name="judul" id="judul" max=500 required value="<?= \$data ? htmlspecialchars(\$data->judul_buku) : " ?>">
: Input untuk judul buku dengan nilai dari data jika ada.

<Label for="penulis">Penulis</Label>: Label untuk input penulis buku.

<input type="text" name="penulis" id="penulis" max=500 required value="<?= \$data ? htmlspecialchars(\$data->penulis) : " ?>">
: Input untuk penulis buku dengan nilai dari data jika ada.

<Label for="penerbit">Penerbit</Label>: Label untuk input penerbit buku.

**`<input type="text" name="penerbit"

Peminjaman.php

require_once 'model.php';: Memuat file model.php yang mungkin berisi fungsi-fungsi untuk berinteraksi dengan database.

\$lends = getData(\$koneksi, 'peminjaman', 'id_peminjaman');: Mengambil data peminjaman dari database menggunakan fungsi getData dan menyimpannya dalam variabel \$lends.

if (\$_SERVER['REQUEST_METHOD'] == 'POST' && isset(\$_POST['hapus'])) { ... } : Memeriksa apakah metode permintaan adalah POST dan apakah tombol hapus ditekan.

\$id = \$_POST['id'];: Mengambil ID peminjaman dari POST.

deleteData(\$koneksi, 'peminjaman', 'id_peminjaman', \$id);: Menghapus data peminjaman dari database berdasarkan ID yang diberikan.

header('Location: Peminjaman.php');: Redirect ke halaman Peminjaman.php setelah operasi berhasil.

exit();: Menghentikan eksekusi skrip lebih lanjut setelah header.

<!DOCTYPE html>: Mendefinisikan dokumen ini sebagai HTML5.

<html>: Tag pembuka untuk dokumen HTML.

<head>: Bagian kepala dari dokumen HTML yang berisi metadata.

<title>Daftar Peminjaman</title>: Menetapkan judul dokumen yang akan ditampilkan di tab browser.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Mengatur viewport untuk desain responsif.

<link rel="preconnect" href="https://fonts.googleapis.com">: Preconnect ke sumber Google Fonts untuk meningkatkan kinerja.

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>: Preconnect ke Google Fonts dengan CORS untuk meningkatkan kinerja.

<link href="https://fonts.googleapis.com/css2?family=Spectral:ital,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;1,300;1,400;1,500;1,600;1,700;1,800&display=swap" rel="stylesheet">: Memuat font Spectral dari Google Fonts.

<link rel="stylesheet" href="styleslists.css">: Menautkan stylesheet eksternal styleslists.css.

<body>: Tag pembuka untuk bagian tubuh dokumen.

<h1>Daftar Peminjaman</h1>: Menampilkan judul halaman "Daftar Peminjaman".

Tambah Peminjaman: Link ke halaman untuk menambahkan peminjaman baru.

<table border="1">: Membuka tabel HTML dengan border.

<tr><th>ID</th>...</th><th>Aksi</th></tr>: Baris header tabel dengan kolom ID, Nama, Judul Buku, Tanggal Pinjam, Tanggal Kembali, dan Aksi.

<?php if (count(\$lends) > 0): ?>: Memeriksa apakah ada data peminjaman.

<?php foreach (\$lends as \$row): ?>: Iterasi melalui setiap baris data peminjaman.

<tr>...</tr>: Membuat baris tabel untuk setiap peminjaman.

<td><?= htmlspecialchars(\$row->id_peminjaman) ?></td>: Menampilkan ID peminjaman dengan aman.

<td><?= htmlspecialchars(\$row->nama_member) ?></td>: Menampilkan nama member dengan aman.

<td><?= htmlspecialchars(\$row->judul_buku) ?></td>: Menampilkan judul buku dengan aman.

<td><?= htmlspecialchars(\$row->tgl_pinjam) ?></td>: Menampilkan tanggal pinjam dengan aman.

<td><?= htmlspecialchars(\$row->tgl_kembali) ?></td>: Menampilkan tanggal kembali dengan aman.

<td>: Kolom aksi untuk edit dan hapus.

<form action="FormPeminjaman.php" method="post">: Form untuk mengedit peminjaman.

<input type="hidden" name="id" value="<?= htmlspecialchars(\$row->id_peminjaman) ?>">: Input tersembunyi untuk menyimpan ID peminjaman.

<button type="submit" name="Edit">Edit</button>: Tombol untuk mengedit peminjaman.

<form method="post">: Form untuk menghapus peminjaman.

<input type="hidden" name="id" value="<?= htmlspecialchars(\$row->id_peminjaman) ?>">: Input tersembunyi untuk menyimpan ID peminjaman.

<button type="submit" name="hapus" class="button edit-button">Hapus</button>: Tombol untuk menghapus peminjaman.

<?php else: ?>: Jika tidak ada data peminjaman.

<tr><td colspan="6">Tidak ada peminjaman yang terdaftar</td></tr>: Menampilkan pesan "Tidak ada peminjaman yang terdaftar".

FormPeminjaman.php

require 'model.php';: Memuat file model.php yang berisi fungsi-fungsi untuk berinteraksi dengan database.

\$id_peminjaman = isset(\$_POST['id']) ? \$_POST['id'] : null;: Mengambil ID peminjaman dari data POST jika tersedia, jika tidak, set ke null.

\$data = null;: Inisialisasi variabel \$data ke null.

if (\$id_peminjaman) { ... }: Jika ID peminjaman tersedia, ambil data peminjaman dari database menggunakan fungsi getById.

if (\$_SERVER['REQUEST_METHOD'] == 'POST' && isset(\$_POST['submit'])) { ... }: Jika metode permintaan adalah POST dan tombol submit ditekan.

\$arr_data = [...];: Membuat array data yang akan dimasukkan atau diperbarui di database.

\$whereColumn = 'id_peminjaman';: Kolom yang digunakan untuk kondisi WHERE dalam query.

\$whereValue = \$_POST['id_peminjaman'];: Nilai kolom untuk kondisi WHERE.

if (isExist(\$koneksi, 'id_member', \$_POST['id_member'], 'member') && isExist(\$koneksi, 'id_buku', \$_POST['id_buku'], 'buku')) { ... }: Memeriksa apakah ID member dan ID buku ada dalam database.

if (isExist(\$koneksi, 'id_peminjaman', \$_POST['id_peminjaman'], 'peminjaman')) { ... }: Memeriksa apakah ID peminjaman ada dalam database.

updateData(\$koneksi, 'peminjaman', \$arr_data, \$whereColumn, \$whereValue);: Jika ID peminjaman ada, lakukan pembaruan data menggunakan fungsi updateData.

insertData(\$koneksi, 'peminjaman', \$arr_data);: Jika ID peminjaman tidak ada, tambahkan data baru menggunakan fungsi insertData.

header('Location: Peminjaman.php');: Redirect ke halaman Peminjaman.php setelah operasi berhasil.

else { echo "Id Member atau Id Buku tidak terdaftar"; }: Menampilkan pesan jika ID member atau ID buku tidak terdaftar.

<!DOCTYPE html>: Mendeklarasikan dokumen HTML5.

<html lang="en">: Mendefinisikan elemen html dengan atribut lang untuk bahasa Inggris.

<head>: Bagian header dari dokumen HTML yang berisi metadata.

<meta charset="UTF-8">: Menentukan set karakter dokumen sebagai UTF-8.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Mengatur viewport agar halaman responsif.

<title>Form Peminjaman</title>: Menentukan judul halaman.

<link rel="preconnect" href="https://fonts.googleapis.com">: Mempercepat koneksi ke Google Fonts.

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>: Mempercepat koneksi ke Google Fonts dengan crossorigin.

<link href="https://fonts.googleapis.com/css2?family=Spectral:ital,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,200;1,300;1,400;1,500;1,600;1,700;1,800&display=swap" rel="stylesheet">: Menyertakan font Google.

<link rel="stylesheet" href="styles.css">: Menyertakan file CSS eksternal untuk styling.

<body>: Bagian utama dari dokumen HTML yang berisi konten halaman.

<h1>Form Peminjaman</h1>: Menampilkan judul halaman.

<form method="post">: Membuat form dengan metode POST.

<input type="hidden" name="id_peminjaman" value="<?= \$data ? htmlspecialchars(\$data->id_peminjaman) : " ?">: Input tersembunyi untuk ID peminjaman.

<label for="id_member">ID Member</label>: Label untuk input ID member.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/kemerekehe/Web-II.vsc/tree/main/Modul%205>