

LAPORAN PEMROGRAMAN BERBASIS WEB



Dosen :

Adi Wahyu Pribadi, S.Si., M.Kom.

Nama :

Sulis Afryanti Lubis – 4522210024

Nadiyah Qasamah – 4522210046

Shafira Desika A – 4522210084

Universitas Pancasila

Fakultas Teknik

Teknik Informatika

2024

OUTPUT HALAMAN UTAMA

Sign in

Group 1

Kampus Merdeka Batch 7

Post Attendee - Zoom

Daftar Buku

localhost / 127.0.0.1 | php

localhost:3000/buku.php

Daftar Buku

[Tambah Buku](#)

ID Buku	Judul	Penulis	Penerbit	Tahun Terbit	Status	Aksi
1	Sayang	Nadia	gramedia.com	2004	tersedia	Hapus
2	Sayang	Nadia	gramedia.com	2004	tersedia	Hapus
3	Bulan	tere liye	kompas.com	2015	dipinjam	Hapus
4	matahari	tere liye	kompas.com	2017	tersedia	Hapus

4

09:58
02/10/2024

Sign in

Group 1

Kampus Merdeka Batch 7

Post Attendee - Zoom

Tambah Buku

localhost / 127.0.0.1 | php

localhost:3000/buku.php?action=create

Tambah Buku

Judul:

Penulis:

Penerbit:

Tahun Terbit:

Status:

Tersedia

Tambah Buku

4

09:59
02/10/2024

Sign in

Group 1

Kampus Merdeka Batch 7

Post Attendee - Zoom

Daftar Pengguna

localhost / 127.0.0.1 / perip

localhost:3000/index.php

Daftar Pengguna

Tambah Pengguna Baru

Nama:

Email:

No Telepon:

Simpan

List Pengguna

ID Pengguna	Nama	Email	No Telepon	Aksi
1	nadiyah qasamah	nadiyahqass@gmail.com	080682693372	Hapus
2	Sulis	sulisulubiss@gmail.com	12345	Hapus
3	Shafira	shafira@gmail.com	6387272	Hapus
4	irma	irma@gmail.com	9386	Hapus
5	cici	titin@gmail.com	36378	Hapus

4

10:00
02/10/2024

Sign in

Group 1

Kampus Merdeka Batch 7

Post Attendee - Zoom

localhost:3000/peminjaman

localhost / 127.0.0.1 / perip

localhost:3000/peminjaman.php

Peminjaman Buku

Pilih Pengguna:

nadiyah qasamah

Pilih Buku:

Sayang

Pinjam Buku

[Kembali](#)

4

10:00
02/10/2024

OUTPUT TAMPILAN DATABASE

The screenshot shows the phpMyAdmin interface with the 'peminjaman' table selected. The table structure is displayed with columns: id_peminjaman, id_user, id_buku, tgl_pinjam, and tgl_kembali. A query is executed, returning an empty result set.

Server: 127.0.0.1 > Database: perpustakaan > Table: peminjaman

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds)

```
SELECT * FROM `peminjaman`
```

Query results operations

Create view

The screenshot shows the phpMyAdmin interface with the 'perpustakaan' database selected. The database structure is displayed, showing three tables: buku, peminjaman, and pengguna. The 'Create new table' form is also visible.

Server: 127.0.0.1 > Database: perpustakaan

Filters

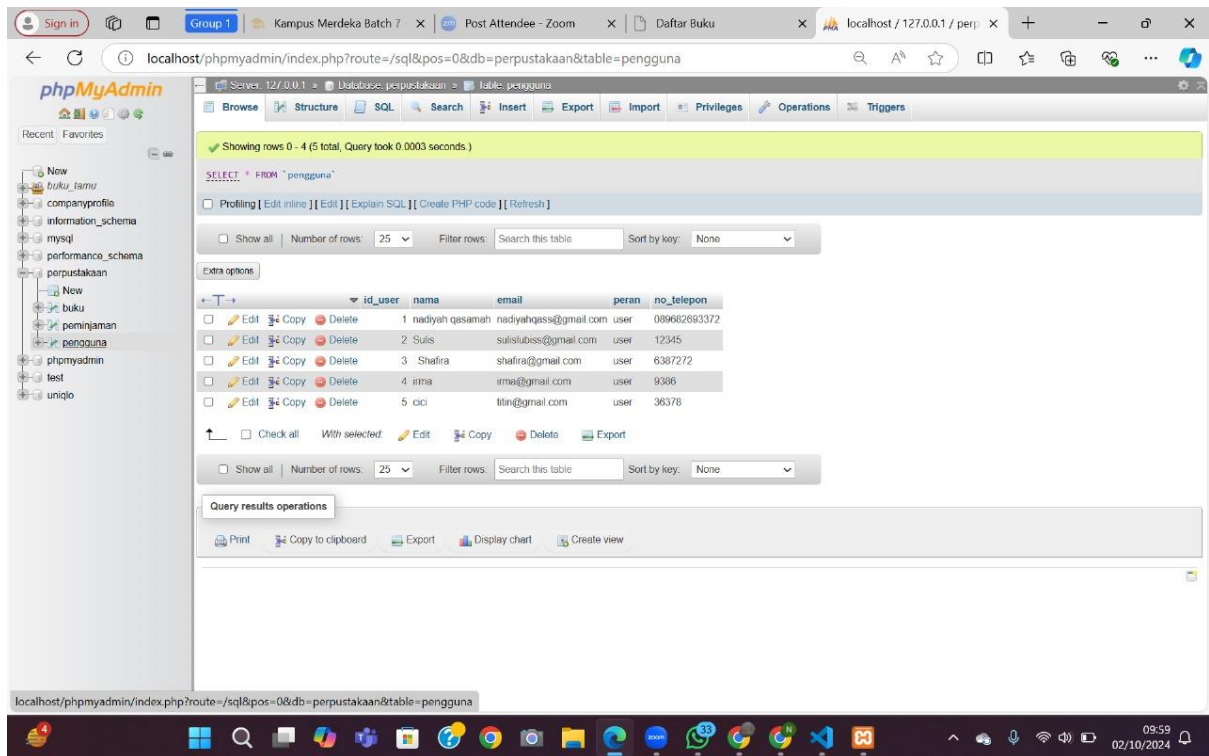
Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
buku	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 Kib	-
peminjaman	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
pengguna	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 Kib	-
3 tables	Sum	9	InnoDB	utf8mb4_general_ci	64.0 Kib	0 B

Create new table

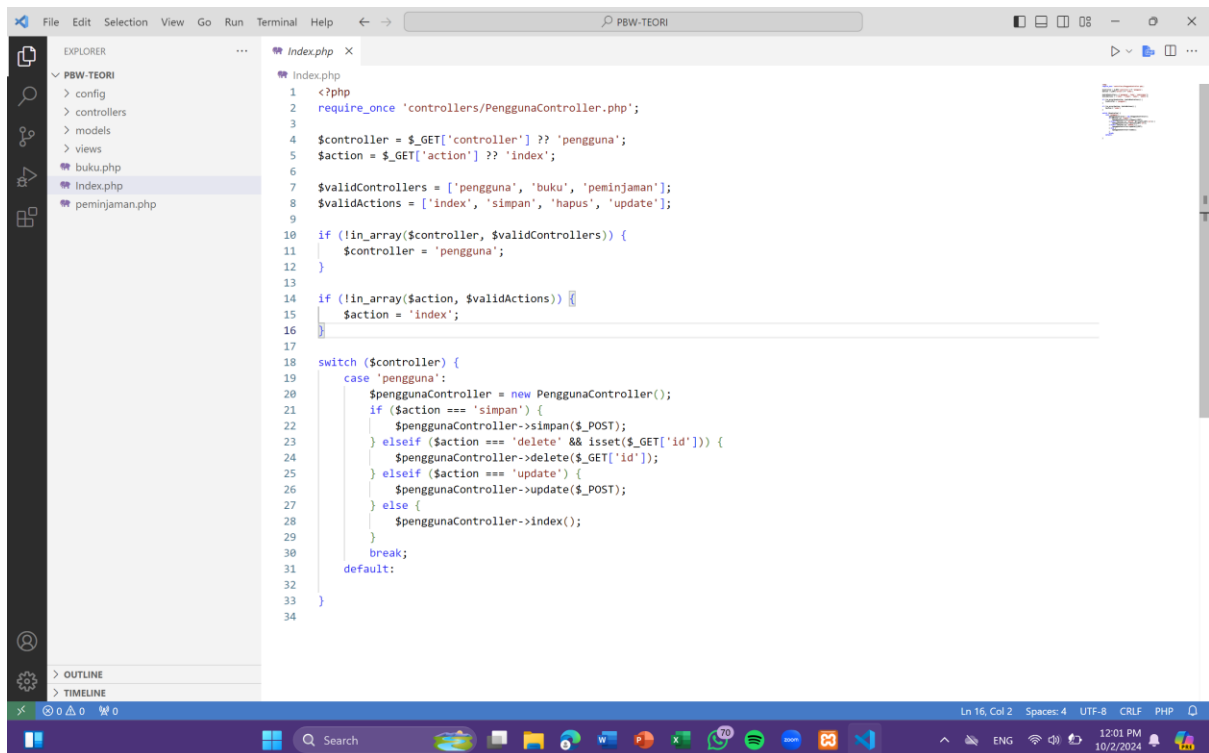
Table name: Number of columns: 4

Create



CODE

Index.php



Buku.php

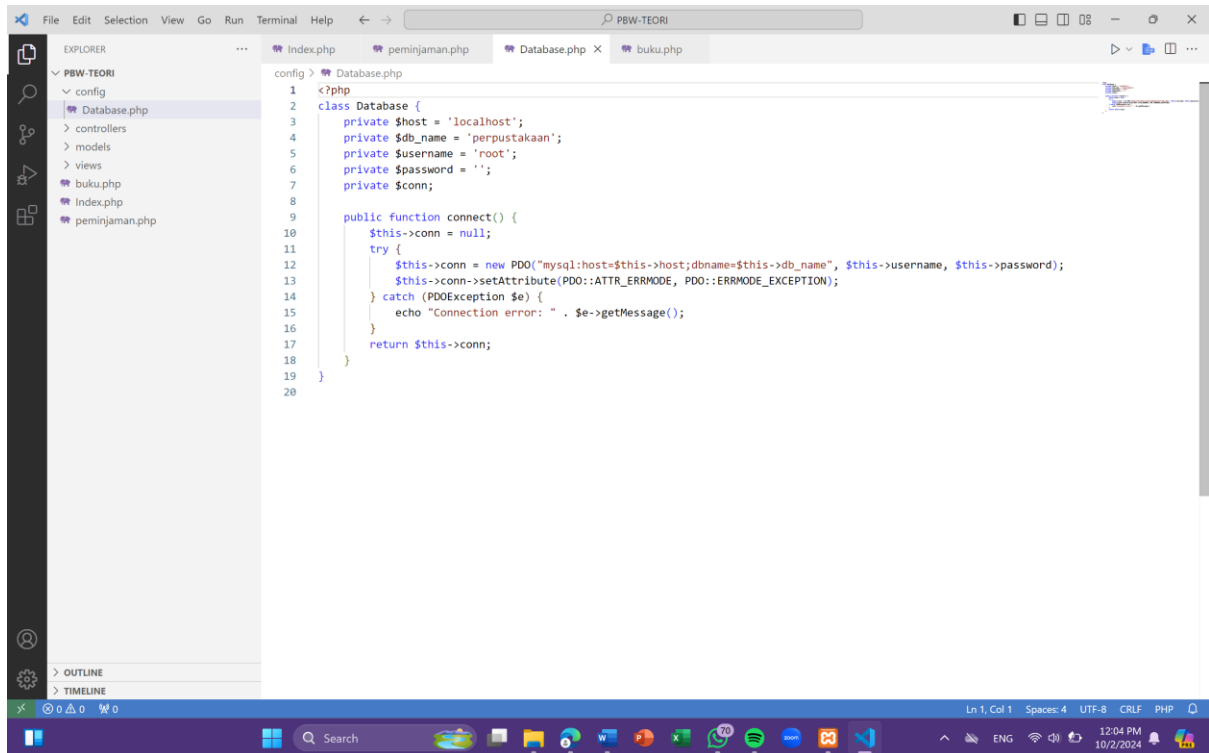
```
1 <?php
2 require_once 'config/Database.php';
3 require_once 'controllers/BukuController.php';
4
5 $database = new Database();
6 $db = $database->connect();
7
8 $controller = $_GET['controller'] ?? 'buku';
9 $action = $_GET['action'] ?? 'index';
10
11 $validControllers = ['buku'];
12 $validActions = ['index', 'simpan', 'hapus', 'update', 'create'];
13
14 if (in_array($controller, $validControllers)) {
15     $controller = 'buku';
16 }
17
18 $bukuController = new BukuController($db);
19 switch ($controller) {
20     case 'buku':
21         if ($action === 'simpan') {
22             if ($_SERVER['REQUEST_METHOD'] === 'POST') {
23                 $bukuController->create();
24             }
25             elseif ($action === 'hapus' && isset($_GET['id_buku'])) {
26                 $bukuController->delete($_GET['id_buku']);
27             }
28             elseif ($action === 'update') {
29                 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
30                     $bukuController->update($_POST);
31                 }
32             }
33             elseif ($action === 'create') {
34                 include 'views/buku_form.php';
35             }
36             else {
37                 $result = $bukuController->index();
38                 include 'views/buku_list.php';
39             }
40             break;
41         default:
42             echo "404 - Page not found.";
43 }
```

Peminjaman.php

```
1 <?php
2 session_start();
3 require_once 'config/Database.php';
4 require_once 'controllers/PeminjamanController.php';
5
6 $db = (new Database())->connect();
7 $action = isset($_GET['action']) ? $_GET['action'] : '';
8
9 switch ($action) {
10     case 'peminjam':
11         $peminjamanController = new PeminjamanController($db);
12         $peminjamanList = $peminjamanController->listPeminjaman();
13         include 'views/peminjam_list.php';
14         break;
15
16     case 'store':
17         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
18             $id_user = $_POST['id_user'];
19             $id_buku = $_POST['id_buku'];
20             $peminjamanController = new PeminjamanController($db);
21             $peminjamanController->storePeminjaman($id_user, $id_buku);
22
23             $pengguna = $peminjamanController->getPenggunaById($id_user);
24             $buku = $peminjamanController->getBukuById($id_buku);
25             $_SESSION['message'] = "Berhasil meminjam buku '{$buku['judul']}' oleh '{$pengguna['nama']}' pada " . date('Y-m-d H:i:s');
26             header("Location: index.php?action=peminjam");
27             exit();
28         }
29         break;
30
31     default:
32         $peminjamanController = new PeminjamanController($db);
33         $penggunaList = $peminjamanController->listPeminjam();
34         $bukuList = $peminjamanController->listBuku();
35         include 'views/peminjaman_form.php';
36         break;
37 }
```

Folder config

Database.php

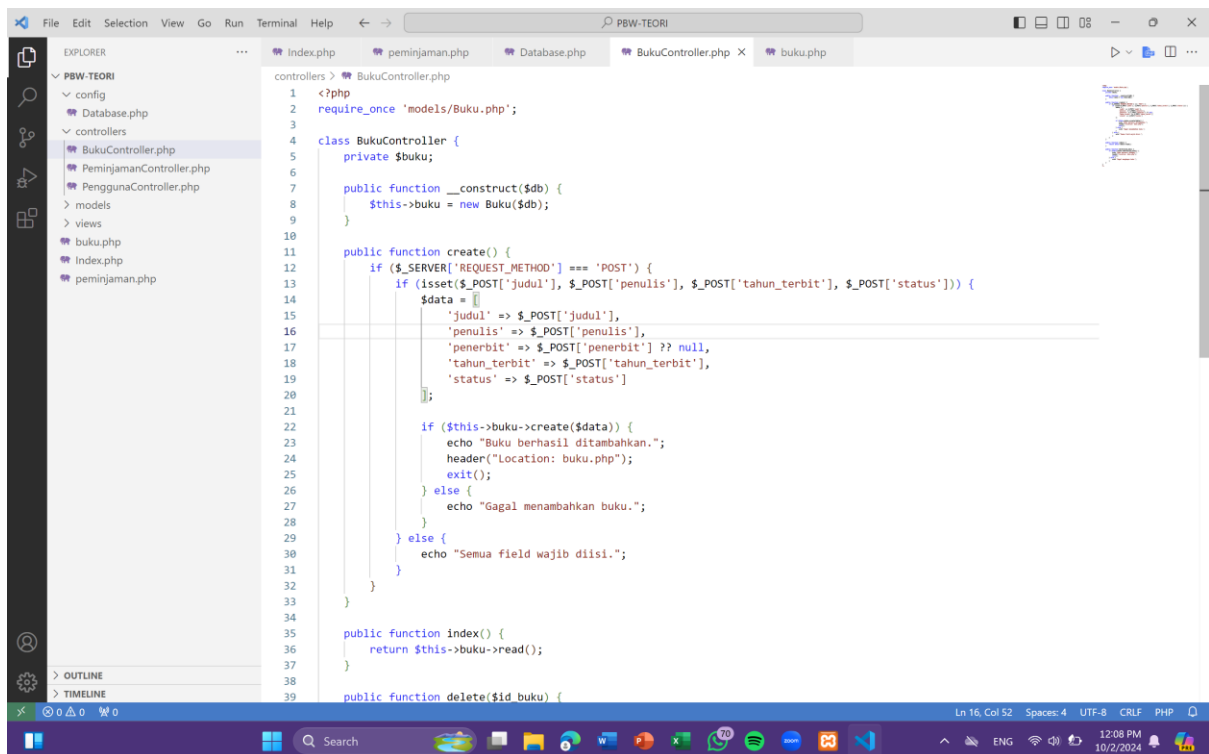


The screenshot shows a Visual Studio Code editor window with the file 'Database.php' open. The Explorer sidebar on the left shows a project structure with folders 'config', 'controllers', 'models', and 'views', and files 'buku.php', 'Index.php', and 'peminjaman.php'. The 'Database.php' file is selected in the Explorer. The code in the editor is as follows:

```
1 <?php
2 class Database {
3     private $host = 'localhost';
4     private $db_name = 'perpustakaan';
5     private $username = 'root';
6     private $password = '';
7     private $conn;
8
9     public function connect() {
10         $this->conn = null;
11         try {
12             $this->conn = new PDO("mysql:host=$this->host;dbname=$this->db_name", $this->username, $this->password);
13             $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
14         } catch (PDOException $e) {
15             echo "Connection error: " . $e->getMessage();
16         }
17         return $this->conn;
18     }
19 }
20
```

Folder controller

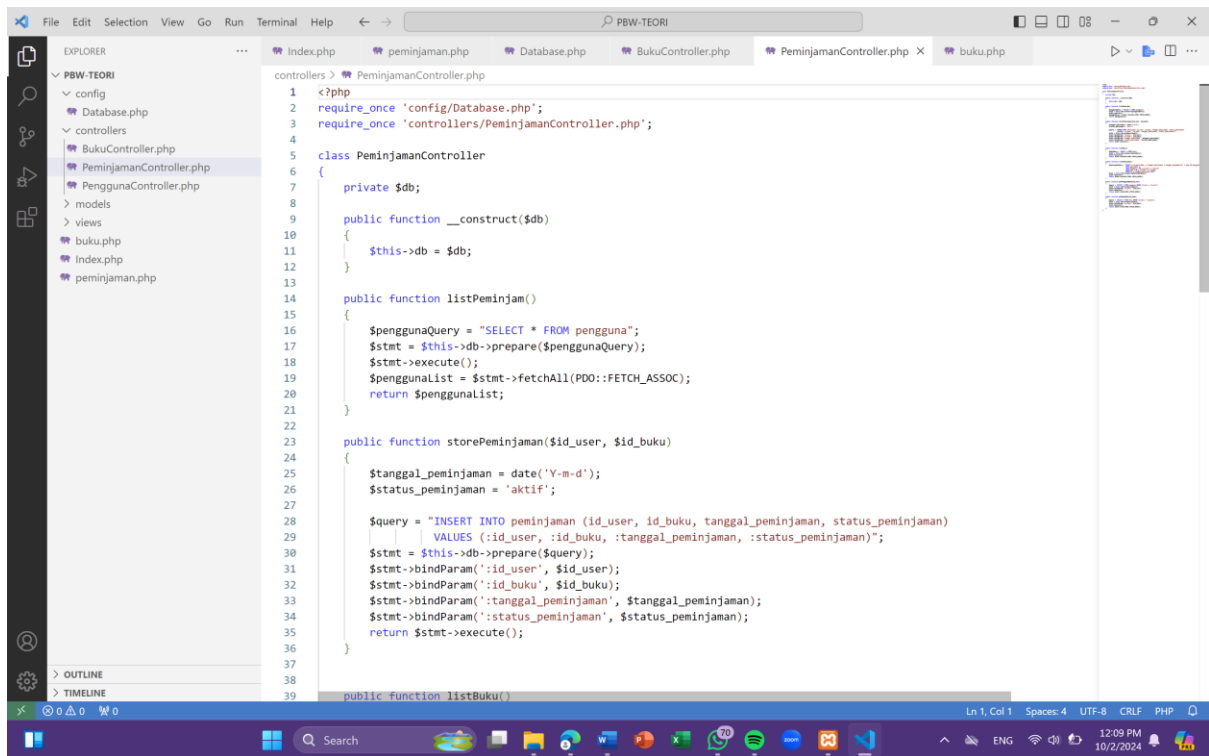
BukuController.php



The screenshot shows a Visual Studio Code editor window with the file 'BukuController.php' open. The Explorer sidebar on the left shows a project structure with folders 'config', 'controllers', 'models', and 'views', and files 'buku.php', 'Index.php', and 'peminjaman.php'. The 'BukuController.php' file is selected in the Explorer. The code in the editor is as follows:

```
1 <?php
2 require_once 'models/Buku.php';
3
4 class BukuController {
5     private $buku;
6
7     public function __construct($db) {
8         $this->buku = new Buku($db);
9     }
10
11     public function create() {
12         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
13             if (isset($_POST['judul'], $_POST['penulis'], $_POST['tahun_terbit'], $_POST['status'])) {
14                 $data = [
15                     'judul' => $_POST['judul'],
16                     'penulis' => $_POST['penulis'],
17                     'penerbit' => $_POST['penerbit'] ?? null,
18                     'tahun_terbit' => $_POST['tahun_terbit'],
19                     'status' => $_POST['status']
20                 ];
21
22                 if ($this->buku->create($data)) {
23                     echo "Buku berhasil ditambahkan.";
24                     header("Location: buku.php");
25                     exit();
26                 } else {
27                     echo "Gagal menambahkan buku.";
28                 }
29             } else {
30                 echo "Semua field wajib diisi.";
31             }
32         }
33     }
34
35     public function index() {
36         return $this->buku->read();
37     }
38
39     public function delete($id_buku) {
40
41     }
42 }
```

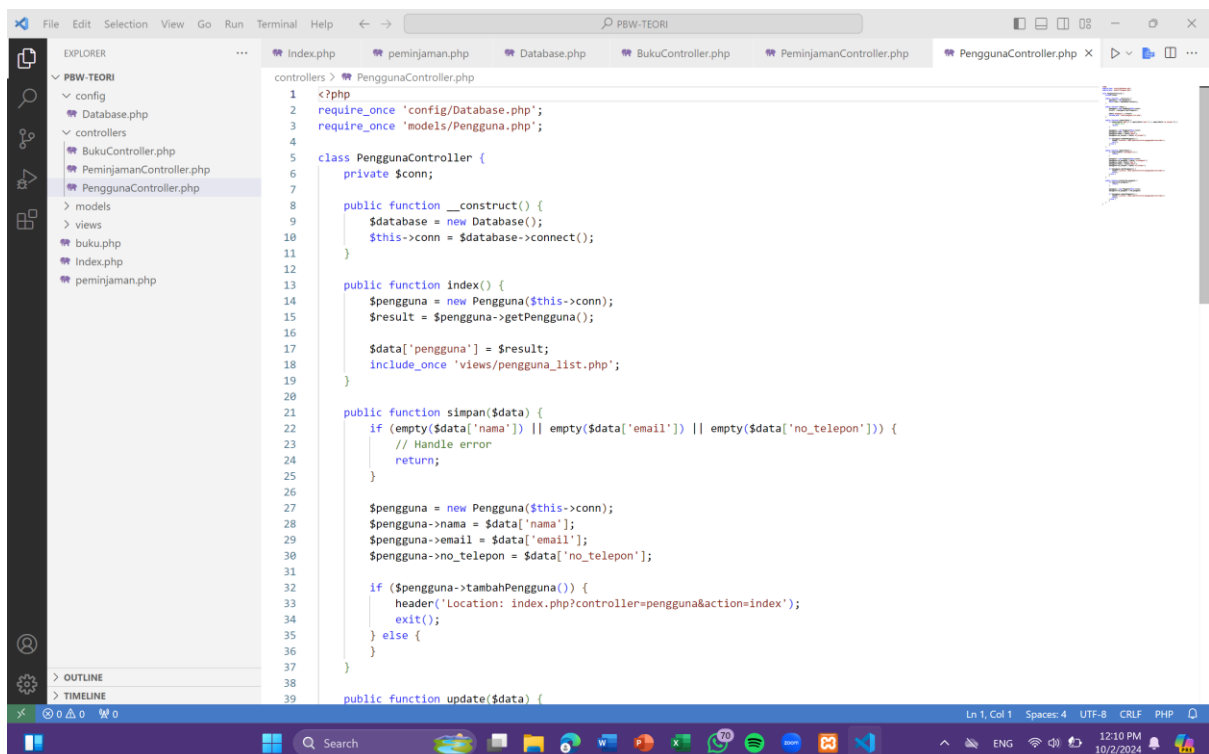
PeminjamanController.php



The screenshot shows the Visual Studio Code editor with the file `PeminjamanController.php` open. The Explorer sidebar on the left shows the project structure for `PBW-TEORI`, including `config`, `Database.php`, `controllers` (with `BukuController.php`, `PeminjamanController.php`, and `PenggunaController.php`), `models`, `views`, `buku.php`, `Index.php`, and `peminjaman.php`. The main editor displays the code for `PeminjamanController` class, which includes methods for listing and storing loans.

```
1 <?php
2 require_once 'config/Database.php';
3 require_once 'controllers/PeminjamanController.php';
4
5 class PeminjamanController
6 {
7     private $db;
8
9     public function __construct($db)
10     {
11         $this->db = $db;
12     }
13
14     public function listPeminjam()
15     {
16         $penggunaQuery = "SELECT * FROM pengguna";
17         $stmt = $this->db->prepare($penggunaQuery);
18         $stmt->execute();
19         $penggunaList = $stmt->fetchAll(PDO::FETCH_ASSOC);
20         return $penggunaList;
21     }
22
23     public function storePeminjaman($id_user, $id_buku)
24     {
25         $tanggal_peminjaman = date('Y-m-d');
26         $status_peminjaman = 'aktif';
27
28         $query = "INSERT INTO peminjaman (id_user, id_buku, tanggal_peminjaman, status_peminjaman)
29             VALUES (:id_user, :id_buku, :tanggal_peminjaman, :status_peminjaman)";
30         $stmt = $this->db->prepare($query);
31         $stmt->bindParam(':id_user', $id_user);
32         $stmt->bindParam(':id_buku', $id_buku);
33         $stmt->bindParam(':tanggal_peminjaman', $tanggal_peminjaman);
34         $stmt->bindParam(':status_peminjaman', $status_peminjaman);
35         return $stmt->execute();
36     }
37
38     public function listBuku()
39 }
```

PenggunaController.php

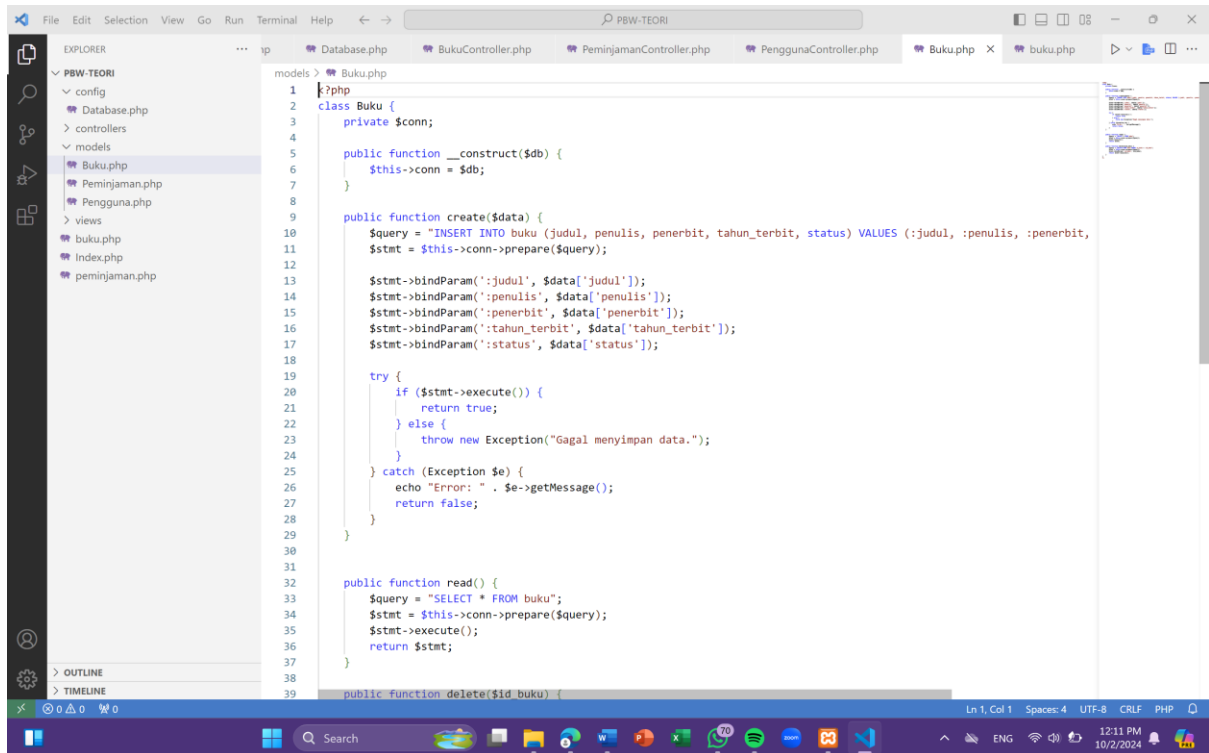


The screenshot shows the Visual Studio Code editor with the file `PenggunaController.php` open. The Explorer sidebar on the left shows the project structure for `PBW-TEORI`, including `config`, `Database.php`, `controllers` (with `BukuController.php`, `PeminjamanController.php`, and `PenggunaController.php`), `models`, `views`, `buku.php`, `Index.php`, and `peminjaman.php`. The main editor displays the code for `PenggunaController` class, which includes methods for listing, saving, and updating users.

```
1 <?php
2 require_once 'config/Database.php';
3 require_once 'models/Pengguna.php';
4
5 class PenggunaController {
6     private $conn;
7
8     public function __construct() {
9         $database = new Database();
10         $this->conn = $database->connect();
11     }
12
13     public function index() {
14         $pengguna = new Pengguna($this->conn);
15         $result = $pengguna->getPengguna();
16
17         $data['pengguna'] = $result;
18         include_once 'views/pengguna_list.php';
19     }
20
21     public function simpan($data) {
22         if (empty($data['nama']) || empty($data['email']) || empty($data['no_telepon'])) {
23             // Handle error
24             return;
25         }
26
27         $pengguna = new Pengguna($this->conn);
28         $pengguna->nama = $data['nama'];
29         $pengguna->email = $data['email'];
30         $pengguna->no_telepon = $data['no_telepon'];
31
32         if ($pengguna->TambahPengguna()) {
33             header('Location: index.php?controller=pengguna&action=index');
34             exit();
35         } else {
36             // Handle error
37         }
38     }
39
40     public function update($data) {
41 }
```

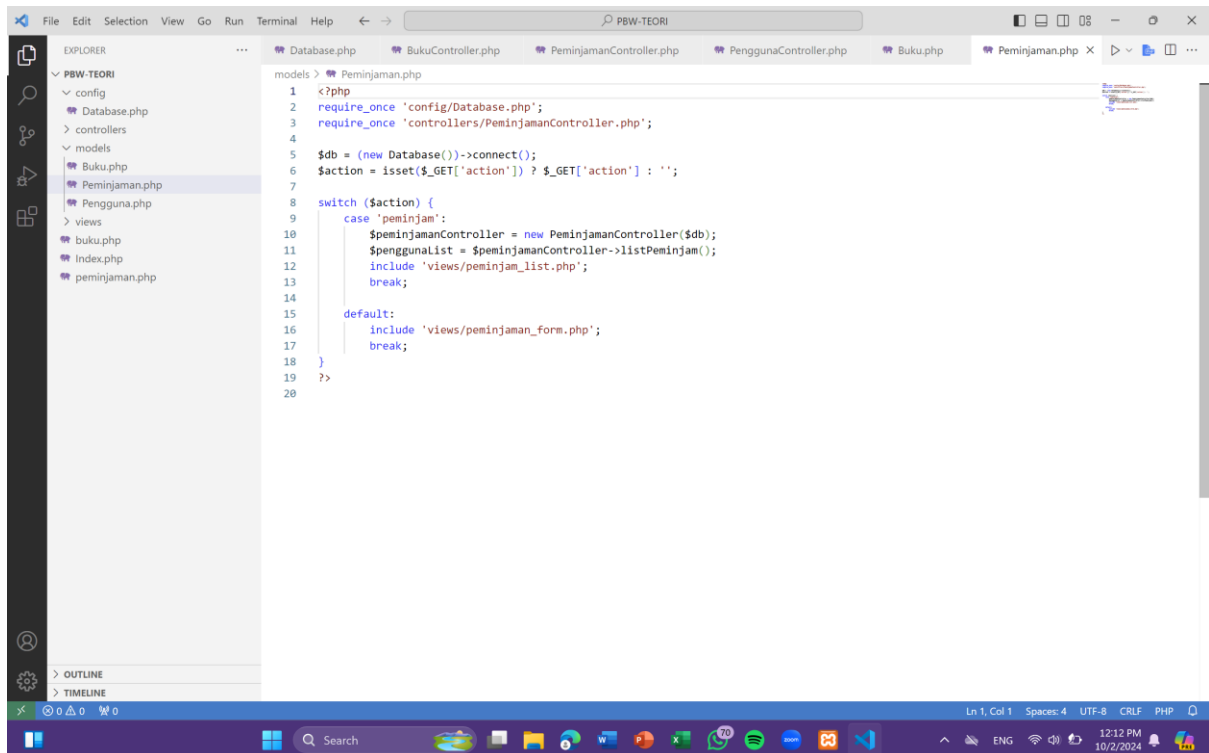

Folder models

Buku.php



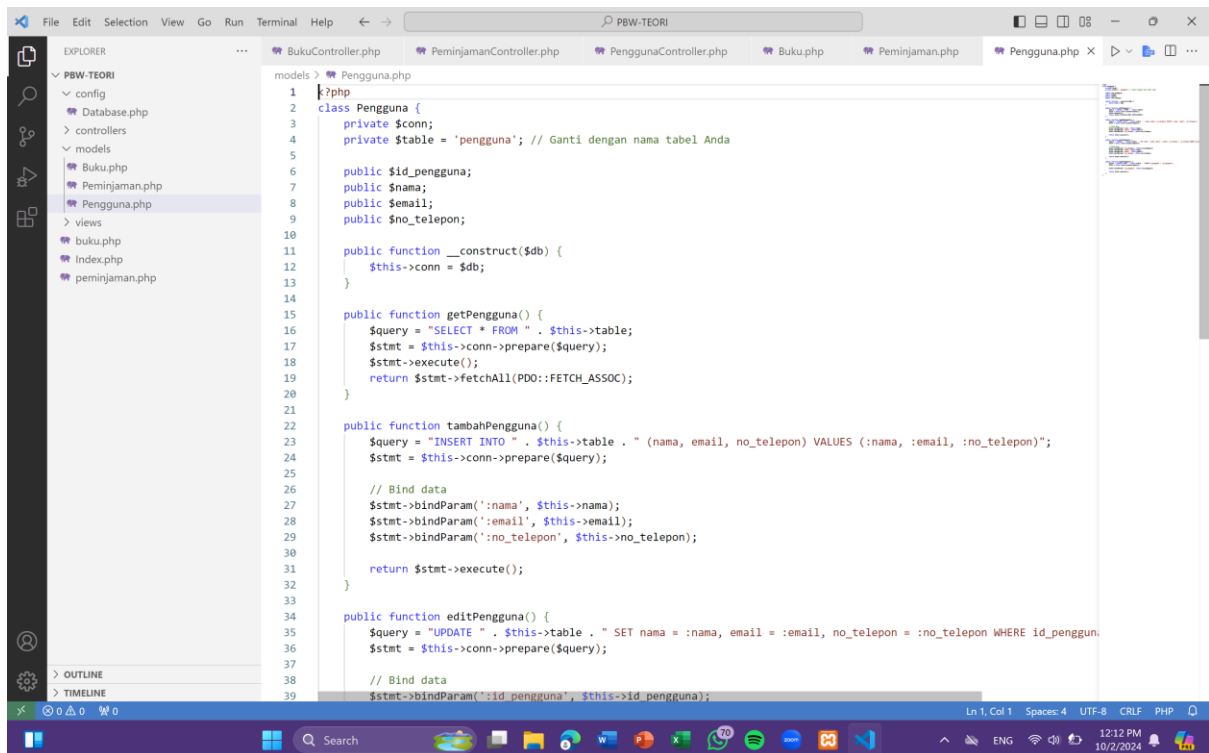
```
1 <?php
2 class Buku {
3     private $conn;
4
5     public function __construct($db) {
6         $this->conn = $db;
7     }
8
9     public function create($data) {
10         $query = "INSERT INTO buku (judul, penulis, penerbit, tahun_terbit, status) VALUES (:judul, :penulis, :penerbit, :tahun_terbit, :status)";
11         $stmt = $this->conn->prepare($query);
12
13         $stmt->bindParam(':judul', $data['judul']);
14         $stmt->bindParam(':penulis', $data['penulis']);
15         $stmt->bindParam(':penerbit', $data['penerbit']);
16         $stmt->bindParam(':tahun_terbit', $data['tahun_terbit']);
17         $stmt->bindParam(':status', $data['status']);
18
19         try {
20             if ($stmt->execute()) {
21                 return true;
22             } else {
23                 throw new Exception("Gagal menyimpan data.");
24             }
25         } catch (Exception $e) {
26             echo "Error: " . $e->getMessage();
27             return false;
28         }
29     }
30
31     public function read() {
32         $query = "SELECT * FROM buku";
33         $stmt = $this->conn->prepare($query);
34         $stmt->execute();
35         return $stmt;
36     }
37
38     public function delete($id buku) {
```

Peminjaman.php



```
1 <?php
2 require_once 'config/Database.php';
3 require_once 'controllers/PeminjamanController.php';
4
5 $db = (new Database())->connect();
6 $action = isset($_GET['action']) ? $_GET['action'] : '';
7
8 switch ($action) {
9     case 'peminjam':
10         $peminjamanController = new PeminjamanController($db);
11         $penggunaList = $peminjamanController->listPeminjam();
12         include 'views/peminjam_list.php';
13         break;
14
15     default:
16         include 'views/peminjaman_form.php';
17         break;
18 }
19 ?>
```

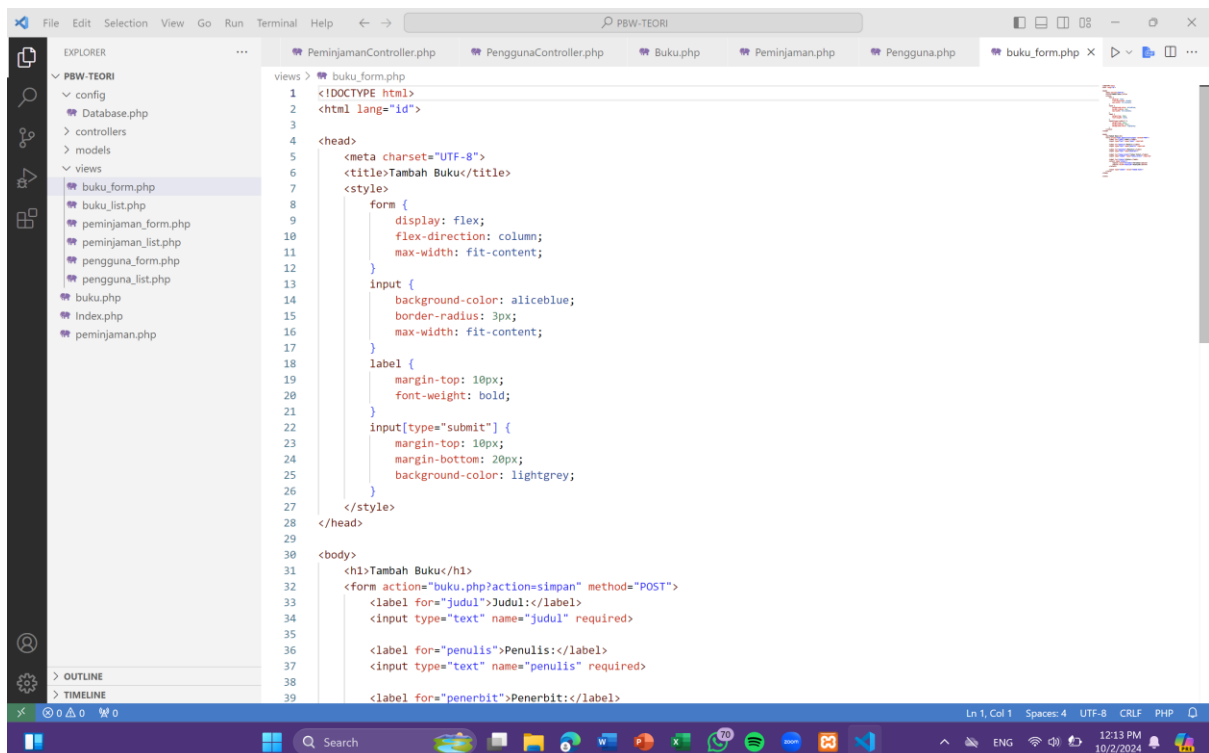
Pengguna.php



```
1 1?php
2 class Pengguna {
3     private $conn;
4     private $table = 'pengguna'; // Ganti dengan nama tabel Anda
5
6     public $id_pengguna;
7     public $nama;
8     public $email;
9     public $no_telepon;
10
11     public function __construct($db) {
12         $this->conn = $db;
13     }
14
15     public function getPengguna() {
16         $query = "SELECT * FROM " . $this->table;
17         $stmt = $this->conn->prepare($query);
18         $stmt->execute();
19         return $stmt->fetchAll(PDO::FETCH_ASSOC);
20     }
21
22     public function tambahPengguna() {
23         $query = "INSERT INTO " . $this->table . " (nama, email, no_telepon) VALUES (:nama, :email, :no_telepon)";
24         $stmt = $this->conn->prepare($query);
25
26         // Bind data
27         $stmt->bindParam(':nama', $this->nama);
28         $stmt->bindParam(':email', $this->email);
29         $stmt->bindParam(':no_telepon', $this->no_telepon);
30
31         return $stmt->execute();
32     }
33
34     public function editPengguna() {
35         $query = "UPDATE " . $this->table . " SET nama = :nama, email = :email, no_telepon = :no_telepon WHERE id_pengguna = :id_pengguna";
36         $stmt = $this->conn->prepare($query);
37
38         // Bind data
39         $stmt->bindParam(':id_pengguna', $this->id_pengguna);
```

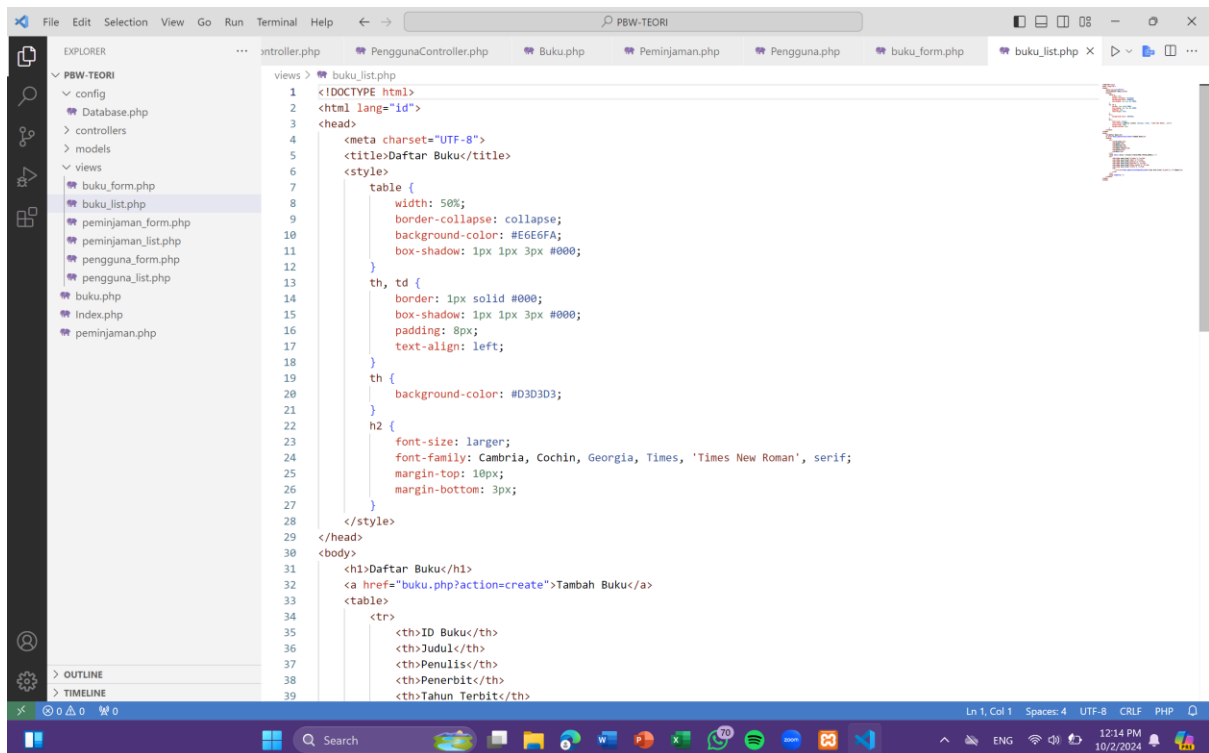
Folder Views

Buku_form.php



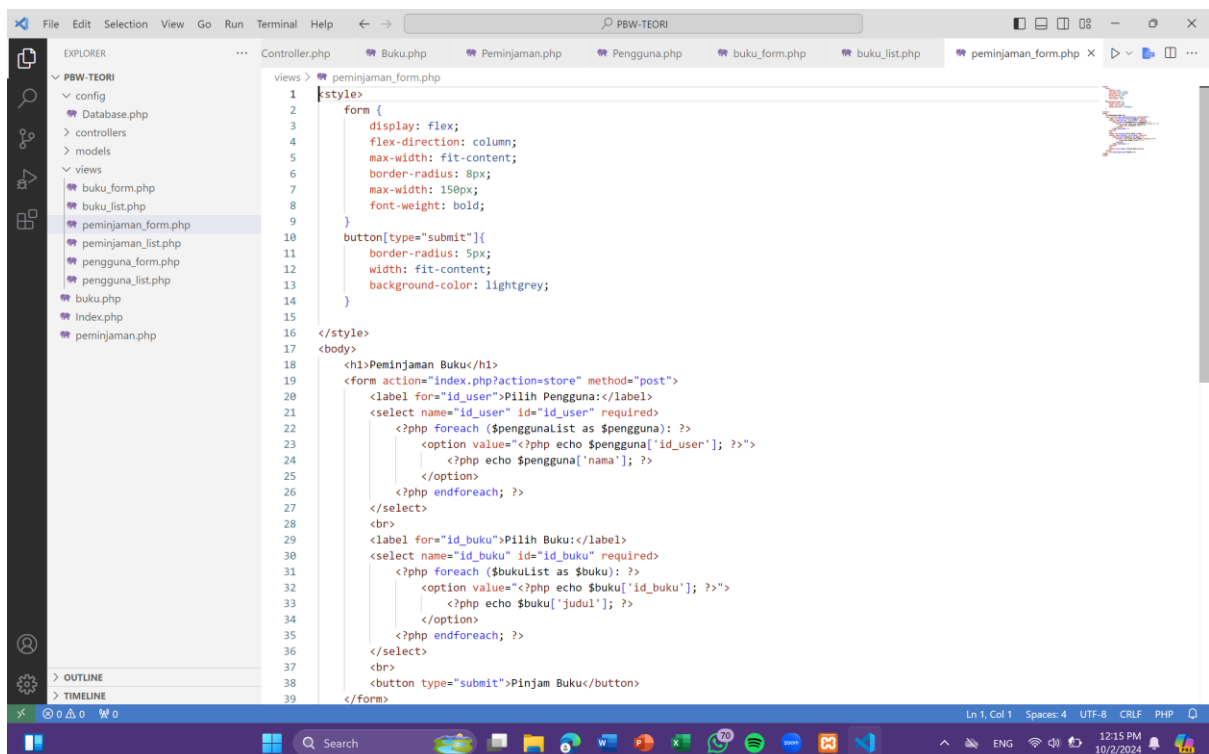
```
1 <!DOCTYPE html>
2 <html lang="id">
3
4 <head>
5     <meta charset="UTF-8">
6     <title>Tambah Buku</title>
7     <style>
8         form {
9             display: flex;
10            flex-direction: column;
11            max-width: fit-content;
12        }
13        input {
14            background-color: aliceblue;
15            border-radius: 3px;
16            max-width: fit-content;
17        }
18        label {
19            margin-top: 10px;
20            font-weight: bold;
21        }
22        input[type="submit"] {
23            margin-top: 10px;
24            margin-bottom: 20px;
25            background-color: lightgrey;
26        }
27    </style>
28 </head>
29
30 <body>
31     <h1>Tambah Buku</h1>
32     <form action="buku.php?action=simpan" method="POST">
33         <label for="judul">Judul:</label>
34         <input type="text" name="judul" required>
35
36         <label for="penulis">Penulis:</label>
37         <input type="text" name="penulis" required>
38
39         <label for="penerbit">Penerbit:</label>
```

Buku_list.php



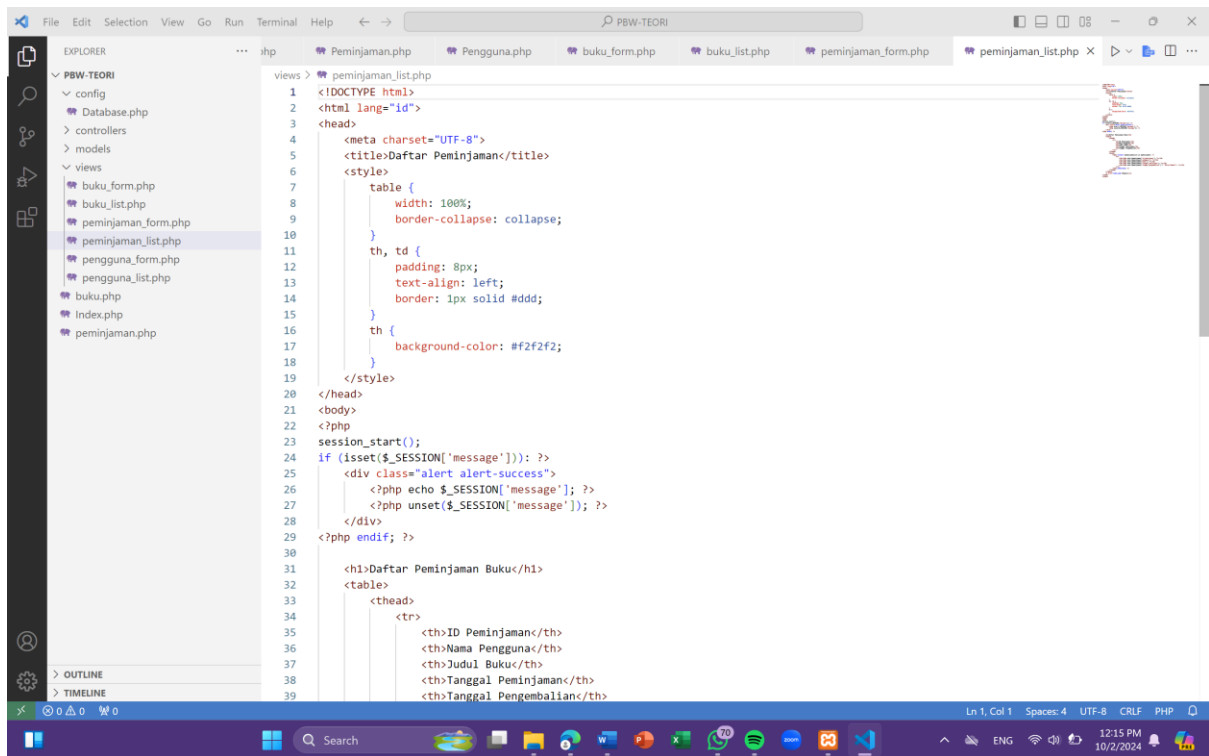
```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4 <meta charset="UTF-8">
5 <title>Daftar Buku</title>
6 <style>
7     table {
8         width: 50%;
9         border-collapse: collapse;
10        background-color: #E6E6FA;
11        box-shadow: 1px 1px 3px #000;
12    }
13    th, td {
14        border: 1px solid #000;
15        box-shadow: 1px 1px 3px #000;
16        padding: 8px;
17        text-align: left;
18    }
19    th {
20        background-color: #D3D3D3;
21    }
22    h2 {
23        font-size: larger;
24        font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
25        margin-top: 10px;
26        margin-bottom: 3px;
27    }
28 </style>
29 </head>
30 <body>
31 <h1>Daftar Buku</h1>
32 <a href="buku.php?action=create">Tambah Buku</a>
33 <table>
34     <tr>
35         <th>ID Buku</th>
36         <th>Judul</th>
37         <th>Penulis</th>
38         <th>Penerbit</th>
39         <th>Tahun Terbit</th>
```

Peminjaman_form



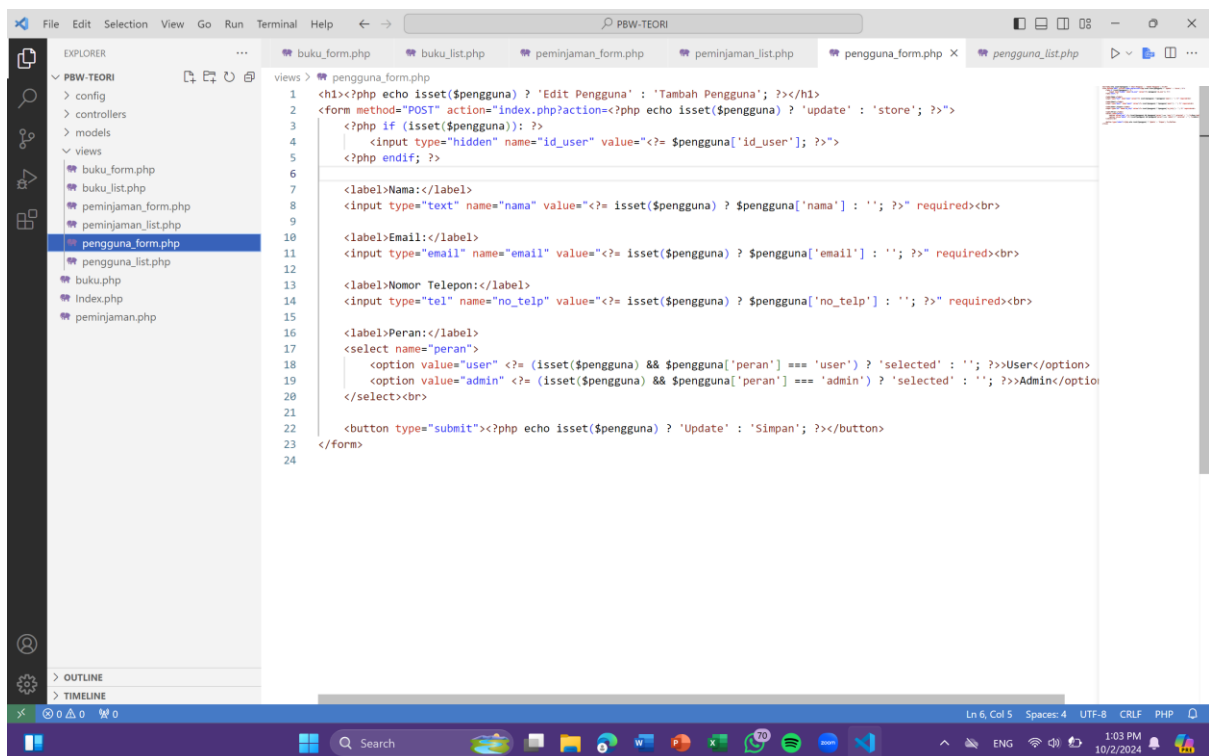
```
1 <style>
2     form {
3         display: flex;
4         flex-direction: column;
5         max-width: fit-content;
6         border-radius: 8px;
7         max-width: 150px;
8         font-weight: bold;
9     }
10    button[type="submit"] {
11        border-radius: 5px;
12        width: fit-content;
13        background-color: lightgrey;
14    }
15 </style>
16 <body>
17 <h1>Peminjaman Buku</h1>
18 <form action="index.php?action=store" method="post">
19     <label for="id_user">Pilih Pengguna:</label>
20     <select name="id_user" id="id_user" required>
21         <?php foreach ($penggunaList as $pengguna): ?>
22             <option value="<?php echo $pengguna['id_user']; ?>">
23                 <?php echo $pengguna['nama']; ?>
24             </option>
25         </select>
26     <br>
27     <label for="id_buku">Pilih Buku:</label>
28     <select name="id_buku" id="id_buku" required>
29         <?php foreach ($bukuList as $buku): ?>
30             <option value="<?php echo $buku['id_buku']; ?>">
31                 <?php echo $buku['judul']; ?>
32             </option>
33         </select>
34     <br>
35     <button type="submit">Pinjam Buku</button>
36 </form>
```

Peminjaman_list.php



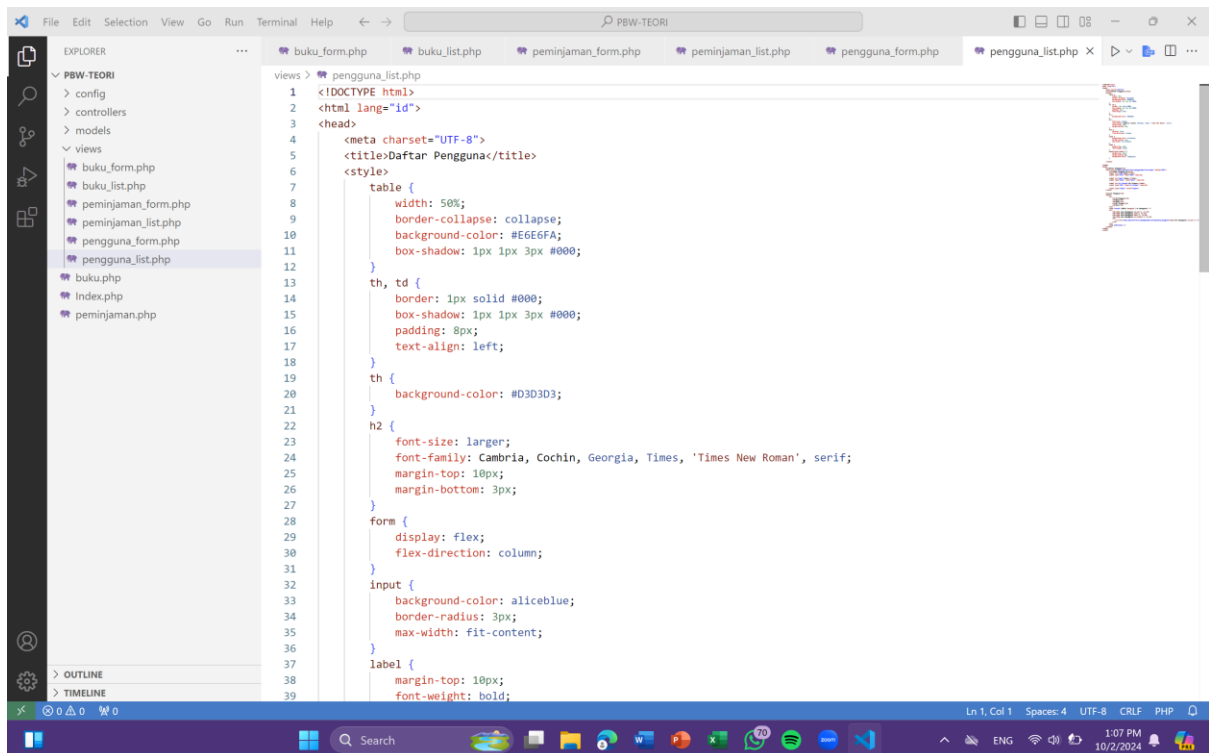
```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4 <meta charset="UTF-8">
5 <title>Daftar Peminjaman</title>
6 <style>
7     table {
8         width: 100%;
9         border-collapse: collapse;
10    }
11    th, td {
12        padding: 8px;
13        text-align: left;
14        border: 1px solid #ddd;
15    }
16    th {
17        background-color: #f2f2f2;
18    }
19 </style>
20 </head>
21 <body>
22 <?php
23 session_start();
24 if (isset($_SESSION['message'])): ?>
25 <div class="alert alert-success">
26 <?php echo $_SESSION['message']; ?>
27 <?php unset($_SESSION['message']); ?>
28 </div>
29 <?php endif; ?>
30
31 <h1>Daftar Peminjaman Buku</h1>
32 <table>
33 <thead>
34 <tr>
35 <th>ID Peminjaman</th>
36 <th>Nama Pengguna</th>
37 <th>Judul Buku</th>
38 <th>Tanggal Peminjaman</th>
39 <th>Tanggal Pengembalian</th>
```

Pengguna_form.php



```
1 <h1><?php echo isset($pengguna) ? 'Edit Pengguna' : 'Tambah Pengguna'; ?></h1>
2 <form method="POST" action="index.php?action=<?php echo isset($pengguna) ? 'update' : 'store'; ?>">
3 <?php if (isset($pengguna)): ?>
4 <input type="hidden" name="id_user" value="<?php echo $pengguna['id_user']; ?>">
5 <?php endif; ?>
6
7 <label>Nama:</label>
8 <input type="text" name="nama" value="<?php echo isset($pengguna) ? $pengguna['nama'] : ''; ?>" required><br>
9
10 <label>Email:</label>
11 <input type="email" name="email" value="<?php echo isset($pengguna) ? $pengguna['email'] : ''; ?>" required><br>
12
13 <label>Nomor Telepon:</label>
14 <input type="tel" name="no_telp" value="<?php echo isset($pengguna) ? $pengguna['no_telp'] : ''; ?>" required><br>
15
16 <label>Peran:</label>
17 <select name="peran">
18 <option value="user" <?php echo (isset($pengguna) && $pengguna['peran'] === 'user') ? 'selected' : ''; ?>>User</option>
19 <option value="admin" <?php echo (isset($pengguna) && $pengguna['peran'] === 'admin') ? 'selected' : ''; ?>>Admin</option>
20 </select><br>
21
22 <button type="submit"><?php echo isset($pengguna) ? 'Update' : 'Simpan'; ?></button>
23 </form>
24
```

Pengguna_list.php



Contoh PDO pada kode Buku.php

```
class Buku {
    private $conn;

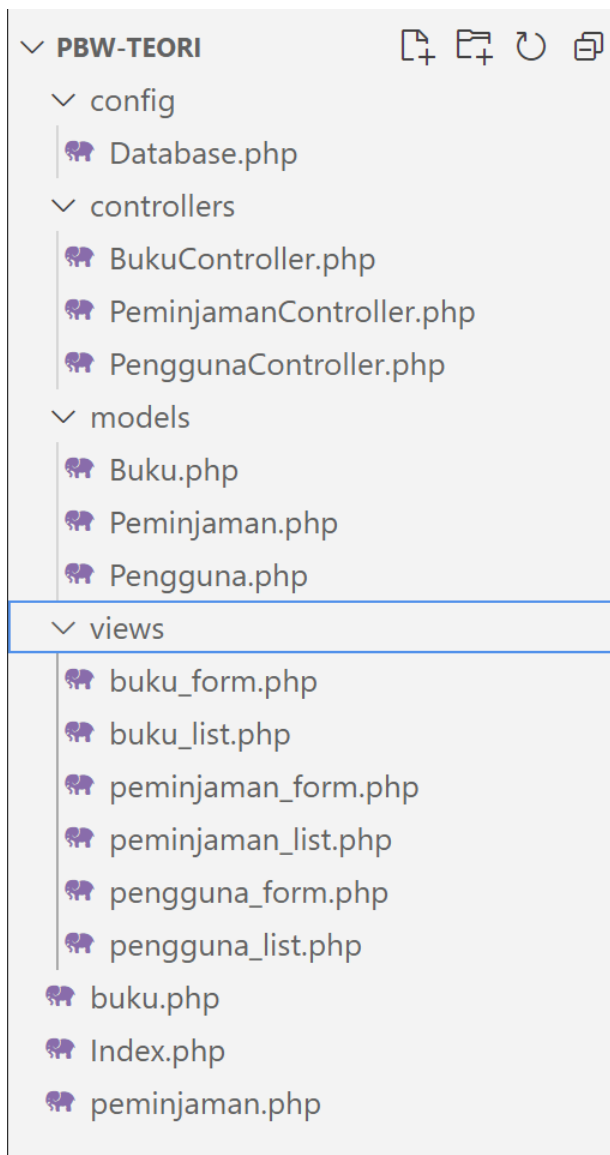
    public function __construct($db) {
        $this->conn = $db;
    }

    public function insert($data) {
        $query = "INSERT INTO buku (judul, penulis, penerbit, tahun_terbit, status) VALUES (:judul, :penulis, :penerbit, :tahun_terbit, :status)";
        $stmt = $this->conn->prepare($query);

        $stmt->bindParam(':judul', $data['judul']);
        $stmt->bindParam(':penulis', $data['penulis']);
        $stmt->bindParam(':penerbit', $data['penerbit']);
        $stmt->bindParam(':tahun_terbit', $data['tahun_terbit']);
        $stmt->bindParam(':status', $data['status']);

        try {
            if ($stmt->execute()) {
                return true;
            } else {
                throw new Exception("Gagal menyimpan data.");
            }
        } catch (Exception $e) {
            echo "Error: " . $e->getMessage();
            return false;
        }
    }
}
```

MVC pada sistem perpustakaan



Config digunakan untuk mengisi program untuk database

Controllers digunakan untuk memanfaatkan objek model Buku, Peminjaman, Pengguna mendapatkan data dan menentukan view yang akan ditampilkan.

Views digunakan untuk menampilkan list yang sudah diinput dalam database.

Models digunakan untuk berinteraksi dengan database yang sudah dibuat.

Link Github : [SHAFIRADESIKAAZ-ZAHRA/PBW_Perpustakaan \(github.com\)](https://github.com/SHAFIRADESIKAAZ-ZAHRA/PBW_Perpustakaan)