

Tugas Prak.Pemrograman Berbasis Web A
untuk memenuhi tugas Prak.Pemrograman Berbasis Web A



Oleh :

Nama : Jati Rahmatulloh

NPM : 4522210039

Dosen :

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

S1-Teknik Informatika

Fakultas Teknik Universitas Pancasila

2023/2024

Tujuan Praktikum:

- Implementasi prinsip OOP dalam PHP.
- Penggunaan PDO untuk koneksi dan manipulasi data di MySQL.
- Menerapkan konsep MVC dalam membangun aplikasi.
- Membuat fitur login admin untuk mengelola (mengedit dan menghapus) komentar dari pengguna.

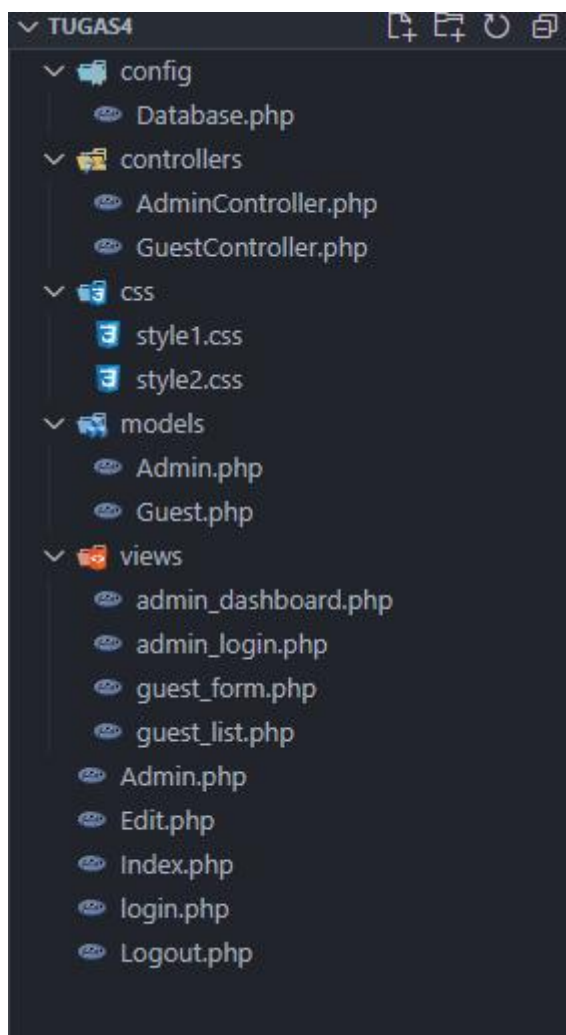
Struktur

Dibangun dengan menggunakan konsep MVC, di mana:

- Model menangani logika bisnis terkait data.
- View bertanggung jawab menampilkan data ke pengguna.
- Controller menjadi penghubung antara model dan view.

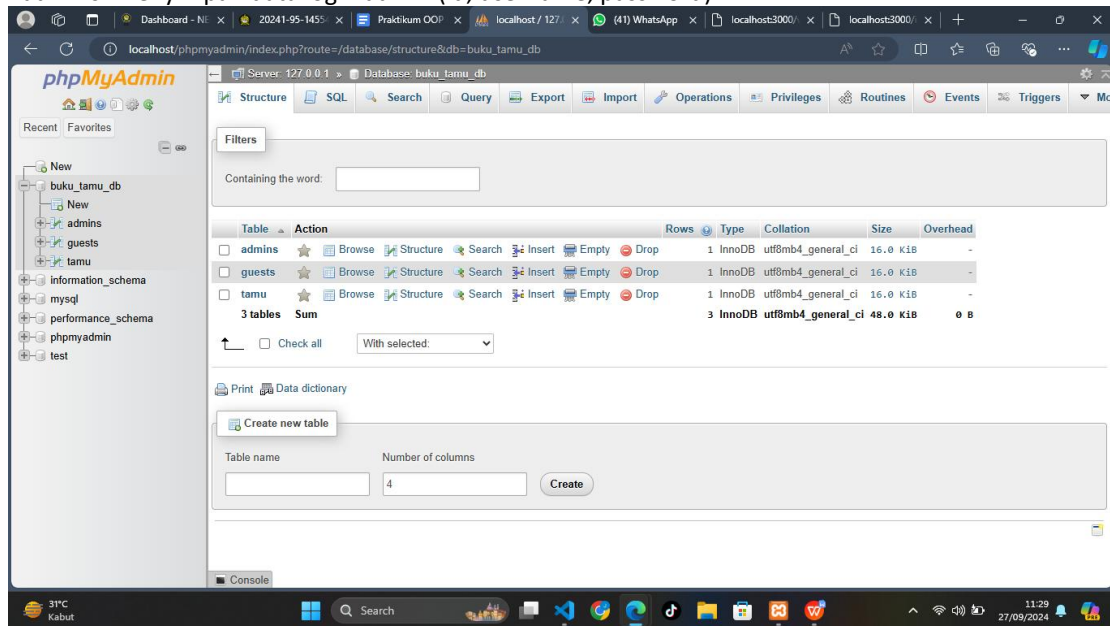
Struktur direktori proyek:

- /buku-tamu/
 - index.php: Halaman utama buku tamu.
 - login.php: Halaman login admin.
 - admin.php: Dashboard admin.
- config/: Folder berisi konfigurasi database.
- controllers/: Folder berisi logika kontrol (GuestController, AdminController).
- models/: Folder berisi model data (Guest, Admin).
- views/: Folder berisi tampilan (guest_form, guest_list, admin_login, admin_dashboard).



Database

- Database yang digunakan bernama buku_tamu, dengan dua tabel utama:
- guests: Menyimpan data buku tamu (id, name, comment, created_at).
- admins: Menyimpan data login admin (id, username, password).



Model (Model PHP)

- Guest.php: Mengelola data buku tamu (membuat, membaca, menghapus, dan memperbarui data tamu).

```
<?php
class Guest {
    private $conn;
    private $table_name = "guests";

    public $id;
    public $name;
    public $comment;
    public $created_at;
    public function __construct($db) {
        $this->conn = $db;
    }
    // Menambahkan data tamu baru
    public function create() {
        $query = "INSERT INTO " . $this->table_name . " SET name=:name,
comment=:comment";
        $stmt = $this->conn->prepare($query);
        // Bersihkan data
        $this->name = htmlspecialchars(strip_tags($this->name));
        $this->comment = htmlspecialchars(strip_tags($this->comment));
        // Bind data
        $stmt->bindParam(":name", $this->name);
        $stmt->bindParam(":comment", $this->comment);
        if($stmt->execute()) {
            return true;
        }
    }
}
```

```

    }
    return false;
}
// Mengambil semua data tamu
public function readAll() {
    $query = "SELECT * FROM " . $this->table_name. " ORDER BY
created_at DESC";
    $stmt = $this->conn->prepare($query);
    $stmt->execute();
    return $stmt;
}
// Menghapus data tamu
public function delete() {
    $query = "DELETE FROM " . $this->table_name. " WHERE id = ?";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $this->id);
    if($stmt->execute()) {
        return true;
    }
    return false;
}
// Mengupdate data tamu
public function update() {
    $query = "UPDATE " . $this->table_name. " SET name=:name,
comment=:comment WHERE id=:id";
    $stmt = $this->conn->prepare($query);
    // Bersihkan data
    $this->name = htmlspecialchars(strip_tags($this->name));
    $this->comment = htmlspecialchars(strip_tags($this->comment));
    $this->id = htmlspecialchars(strip_tags($this->id));
    // Bind data
    $stmt->bindParam(":name", $this->name);
    $stmt->bindParam(":comment", $this->comment);
    $stmt->bindParam(":id", $this->id);
    if($stmt->execute()) {
        return true;
    }
    return false;
}
}
?>

```

- Admin.php: Menangani login admin dengan menggunakan password hash

```

<?php
class Admin {
    private $conn;
    private $table_name = "admins";

    public $id;

```

```

    public $username;
    public $password;
    public function __construct($db) {
        $this->conn = $db;
    }
    // Fungsi Login admin
    public function login() {
        $query = "SELECT * FROM " . $this->table_name . " WHERE username
= :username LIMIT 0,1";
        $stmt = $this->conn->prepare($query);
        // Bersihkan data
        $this->username = htmlspecialchars(strip_tags($this->username));
        // Bind data
        $stmt->bindParam(":username", $this->username);
        $stmt->execute();
        $row = $stmt->fetch(PDO::FETCH_ASSOC)
        if($row) {
            if(password_verify($this->password, $row['password'])) {
                $this->id = $row['id'];
                return true;
            }
        }
        return false;
    }
}
?>

```

- Admin :

```

<?php
class Admin {
    private $conn;
    private $table_name = "admins";

    public $id;
    public $username;
    public $password;
    public function __construct($db) {
        $this->conn = $db;
    }
    // Fungsi Login admin
    public function login() {
        $query = "SELECT * FROM " . $this->table_name . " WHERE username
= :username LIMIT 0,1";
        $stmt = $this->conn->prepare($query);
        // Bersihkan data
        $this->username = htmlspecialchars(strip_tags($this->username));
        // Bind data
        $stmt->bindParam(":username", $this->username);
        $stmt->execute();
    }
}

```

```

        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        if($row) {
            if(password_verify($this->password, $row['password'])) {
                $this->id = $row['id'];
                return true;
            }
        }
        return false;
    }
}
?>

```

- Guest :

```

<?php
class Guest {
    private $conn;
    private $table_name = "guests";

    public $id;
    public $name;
    public $comment;
    public $created_at;
    public function __construct($db) {
        $this->conn = $db;
    }
    // Menambahkan data tamu baru
    public function create() {
        $query = "INSERT INTO " . $this->table_name . " SET name=:name,
comment=:comment";
        $stmt = $this->conn->prepare($query);
        // Bersihkan data
        $this->name = htmlspecialchars(strip_tags($this->name));
        $this->comment = htmlspecialchars(strip_tags($this->comment));
        // Bind data
        $stmt->bindParam(":name", $this->name);
        $stmt->bindParam(":comment", $this->comment);
        if($stmt->execute()) {
            return true;
        }
        return false;
    }
    // Mengambil semua data tamu
    public function readAll() {
        $query = "SELECT * FROM " . $this->table_name . " ORDER BY
created_at DESC";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
    }
}

```

```

        return $stmt;
    }
    // Menghapus data tamu
    public function delete() {
        $query = "DELETE FROM " . $this->table_name. " WHERE id = ?";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(1, $this->id);
        if($stmt->execute()) {
            return true;
        }
        return false;
    }
    // Mengupdate data tamu
    public function update() {
        $query = "UPDATE " . $this->table_name. " SET name=:name,
comment=:comment WHERE id=:id";
        $stmt = $this->conn->prepare($query);
        // Bersihkan data
        $this->name = htmlspecialchars(strip_tags($this->name));
        $this->comment = htmlspecialchars(strip_tags($this->comment));
        $this->id = htmlspecialchars(strip_tags($this->id));
        // Bind data
        $stmt->bindParam(":name", $this->name);
        $stmt->bindParam(":comment", $this->comment);
        $stmt->bindParam(":id", $this->id);
        if($stmt->execute()) {
            return true;
        }
        return false;
    }
}
?>

```

Controller (Controller PHP)

- GuestController.php: Mengelola interaksi antara data buku tamu dan tampilan (create, readAll, update, delete).

```

<?php
include_once './config/Database.php';
include_once './models/Guest.php';

class GuestController {
    private $guest;
    public function __construct() {
        $database = new Database();
        $db = $database->getConnection();
        $this->guest = new Guest($db);
    }
}

```

```

// Fungsi untuk menambahkan tamu baru
public function create($name, $comment) {
    $this->guest->name = $name;
    $this->guest->comment = $comment;
    return $this->guest->create();
}
// Fungsi untuk mendapatkan semua data tamu
public function readAll() {
    return $this->guest->readAll();
}
// Fungsi untuk menghapus tamu
public function delete($id) {
    $this->guest->id = $id;
    return $this->guest->delete();
}
// Fungsi untuk mengupdate tamu
public function update($id, $name, $comment) {
    $this->guest->id = $id;
    $this->guest->name = $name;
    $this->guest->comment = $comment;
    return $this->guest->update();
}
}
?>

```

- AdminController.php: Mengelola login admin.

```

<?php
include_once './config/Database.php';
include_once './models/Admin.php';

class AdminController {
    private $admin;
    public function __construct() {
        $database = new Database();
        $db = $database->getConnection();
        $this->admin = new Admin($db);
    }
    // Fungsi untuk login
    public function login($username, $password) {
        $this->admin->username = $username;
        $this->admin->password = $password;
        return $this->admin->login();
    }
}
?>

```


View (Tampilan HTML dan PHP)

- guest_form.php: Formulir untuk pengunjung buku tamu.

```
<h2>Selamat Datang di Buku Tamu</h2>
<link rel="stylesheet" href="css/style1.css">
<form action="index.php" method="post">
  <label for="name">Nama:</label><br>
  <input type="text" name="name" required><br><br>
  <label for="comment">Komentar:</label><br>
  <textarea name="comment" rows="5" required></textarea><br><br>
  <input type="submit" value="Kirim">
</form>
```

- guest_list.php: Daftar komentar buku tamu.

```
<h3>Daftar Komentar</h3>
<table border="1" cellpadding="5" cellspacing="0">
  <tr>
    <th>Nama</th>
    <th>Komentar</th>
    <th>Waktu</th>
  </tr>
  <?php while ($row = $guests->fetch(PDO::FETCH_ASSOC)) { ?>
    <tr>
      <td><?php echo htmlspecialchars($row['name']); ?></td>
      <td><?php echo htmlspecialchars($row['comment']); ?></td>
      <td><?php echo $row['created_at']; ?></td>
    </tr>
  <?php } ?>
</table>
```

- admin_login.php: Formulir login untuk admin.

```
<h2>Login Admin</h2>
<link rel="stylesheet" href="/css/style2.css">
<form action="login.php" method="post">
  <label for="username">Username:</label><br>
  <input type="text" name="username" required><br><br>
  <label for="password">Password:</label><br>
  <input type="password" name="password" required><br><br>
  <input type="submit" value="Login">
</form>
```

- admin_dashboard.php: Dashboard admin untuk mengelola data tamu

```
<h2>Dashboard Admin</h2>
<p>Selamat datang, Admin!</p>
<a href="logout.php">Logout</a>
<table border="1" cellpadding="5" cellspacing="0">
  <tr>
```

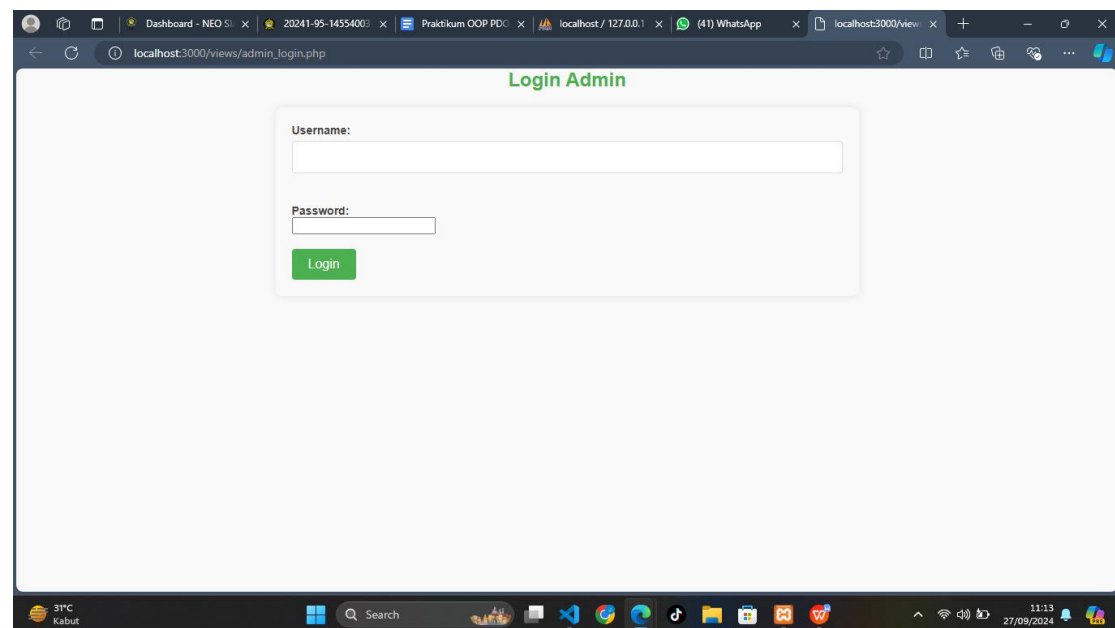
```

        <th>Nama</th>
        <th>Komentar</th>
        <th>Waktu</th>
        <th>Aksi</th>
    </tr>
    <?php while ($row = $guests->fetch(PDO::FETCH_ASSOC)) { ?>
    <tr>
        <td><?php echo htmlspecialchars($row['name']); ?></td>
        <td><?php echo htmlspecialchars($row['comment']); ?></td>
        <td><?php echo $row['created_at']; ?></td>
        <td>
            <a href="edit.php?id=<?php echo $row['id']; ?>">Edit</a> |
            <a href="admin.php?delete_id=<?php echo $row['id']; ?>"
onclick="return confirm('Yakin ingin menghapus?')">Hapus</a>
        </td>
    </tr>
    <?php } ?>
</table>

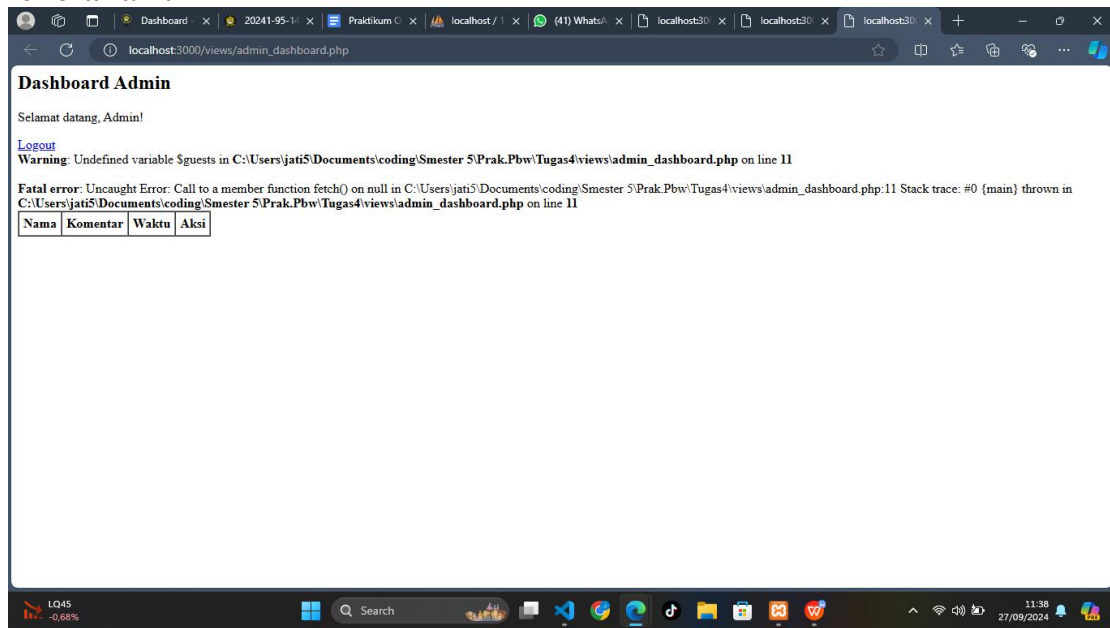
```

Login dan Dashboard Admin

- login.php: Admin dapat login dengan memasukkan username dan password.



- admin.php: Menampilkan dashboard admin yang memungkinkan admin menghapus atau mengedit komentar tamu.



- logout.php: Fungsi untuk logout admin.

Kesimpulan

Buku Tamu yang dibangun menggunakan OOP, PDO, dan MVC memungkinkan:

- Pengelolaan komentar tamu melalui antarmuka admin.
- Login admin dengan verifikasi password yang aman.
- Struktur proyek yang modular dan mudah dikelola sesuai konsep MVC.