

# **TUGAS PRAKTIKUM**

## **PEMROGRAMAN BERBASIS WEB (A)**

Untuk memenuhi tugas Praktikum Pemrograman Berbasis Web



Oleh :

Nama : Dzaki Yushiibanaa

NPM : 4522210041

Kelas : A

Dosen :

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

S1-Teknik Informatika

Fakultas Teknik Universitas Pancasila

2024/2025

# Aplikasi Buku Tamu

## Pendahuluan

Aplikasi Buku Tamu ini dibuat dengan tujuan untuk mengimplementasikan beberapa konsep penting dalam pengembangan aplikasi web berbasis PHP, yaitu:

1. **Prinsip OOP (Object-Oriented Programming):** Digunakan untuk memisahkan logika bisnis dalam class dan objek.
2. **PDO (PHP Data Objects):** Untuk mengelola koneksi dan manipulasi data di basis data MySQL dengan cara yang aman dan efisien.
3. **MVC (Model-View-Controller):** Digunakan untuk memisahkan logika aplikasi, manipulasi data, dan tampilan menjadi tiga bagian terpisah.

Tujuan utama dari aplikasi ini adalah:

- Memungkinkan pengguna untuk memasukkan nama dan komentar ke buku tamu.
- Mengizinkan admin untuk mengelola komentar (menambah, mengedit, atau menghapus komentar).
- Menerapkan fitur login admin dengan enkripsi kata sandi untuk keamanan.

## Langkah-Langkah

### 1. Membuat Basis Data dan Tabel

Pertama, dibuat basis data dengan nama buku\_tamu yang berisi dua tabel:

- **guests** untuk menyimpan nama, komentar, dan waktu pengiriman komentar.
- **admins** untuk menyimpan data admin, termasuk username dan password yang sudah di-hash.

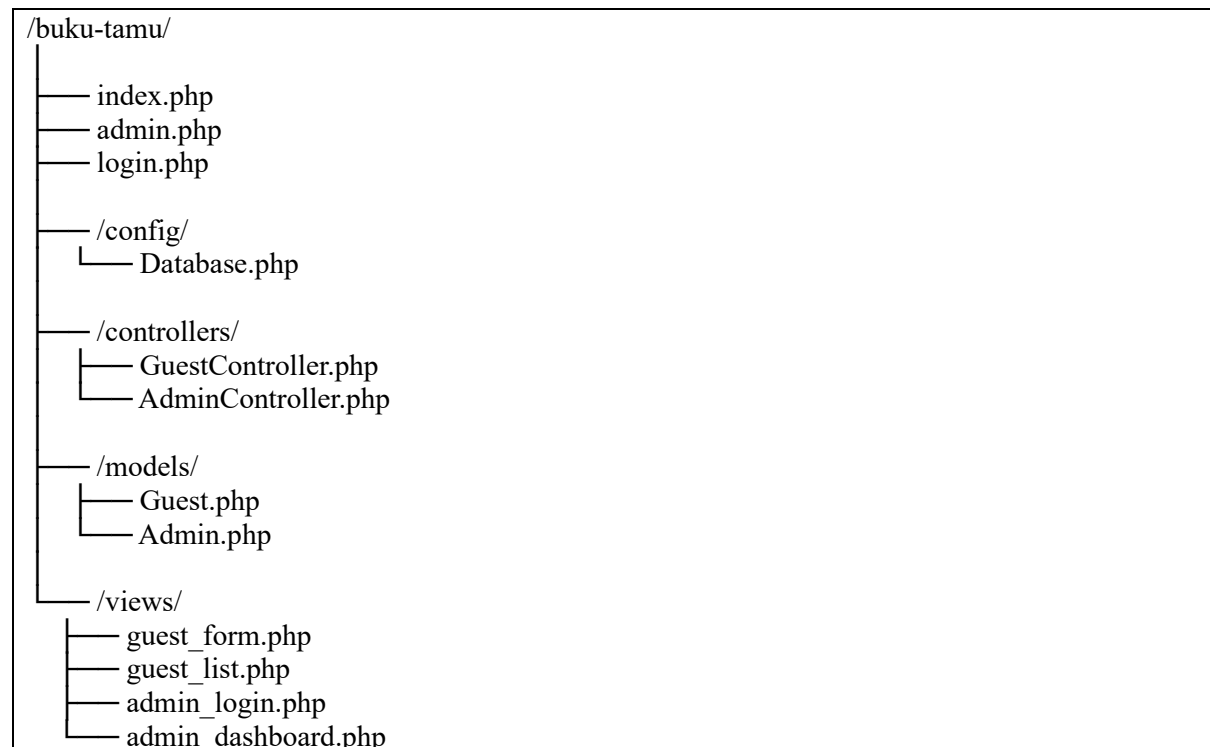
```
CREATE DATABASE buku_tamu;
USE buku_tamu;

CREATE TABLE guests (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    comment TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE admins (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL
);
```

### 2. Struktur Proyek

Struktur proyek aplikasi ini mengikuti arsitektur MVC:



## Membuat Model

Terdapat dua model: Guest untuk menangani data tamu dan Admin untuk menangani data admin serta proses login.

Admin.php

```
<?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

```

<?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;
}
}
?>

```

#### 4. Membuat Kontroller

- **GuestController** bertanggung jawab untuk menangani semua operasi terkait tamu (membaca, menambah, mengedit, menghapus).

```

<?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** menangani 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);
}

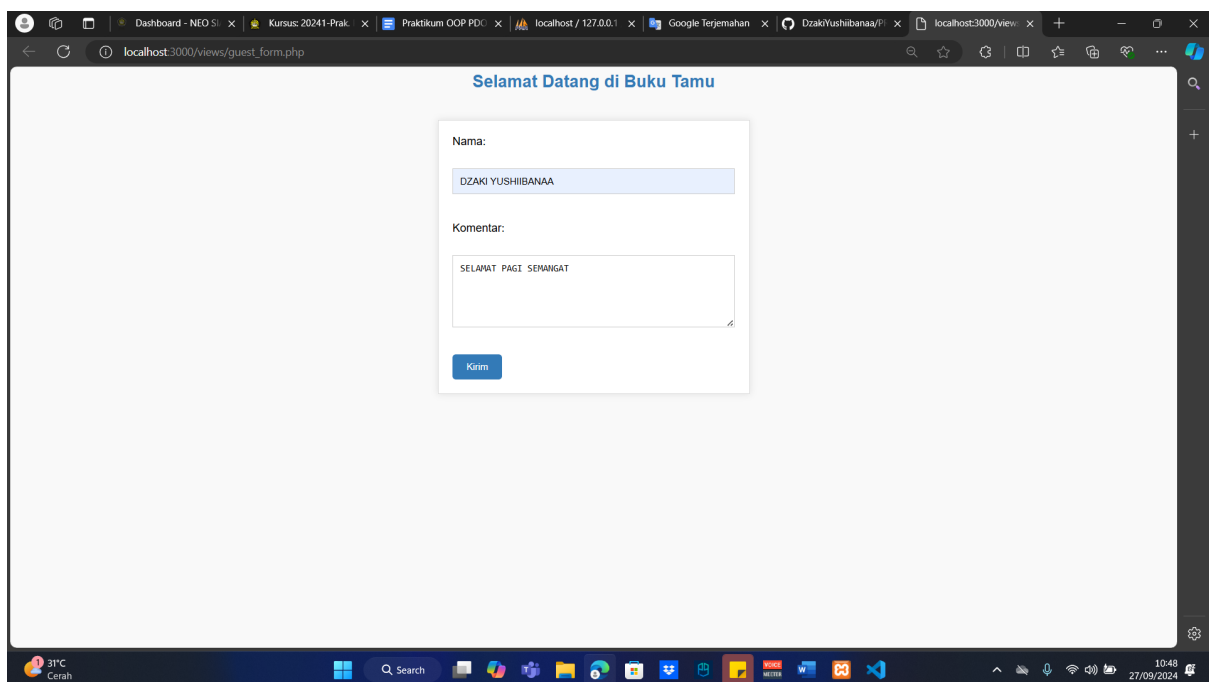
// Fungsi untuk login
public function login($username, $password) {
    $this->admin->username = $username;
    $this->admin->password = $password;
    return $this->admin->login();
}
}
?>

```

## 5. Membuat Tampilan

Tampilan terdiri dari form buku tamu, daftar komentar, form login admin, dan dashboard admin.

guest\_form.php



Code

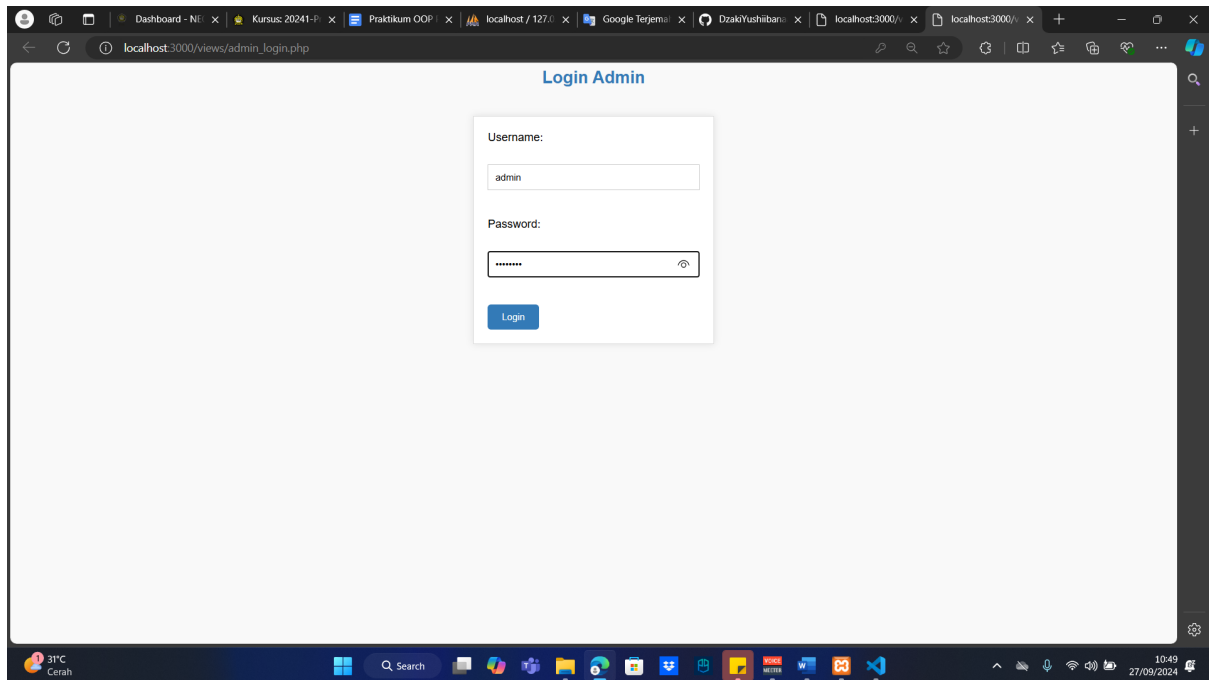
```

<link rel="stylesheet" type="text/css" href="/css/style1.css">
<h2>Selamat Datang di Buku Tamu</h2>
<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>
```

## Admin login

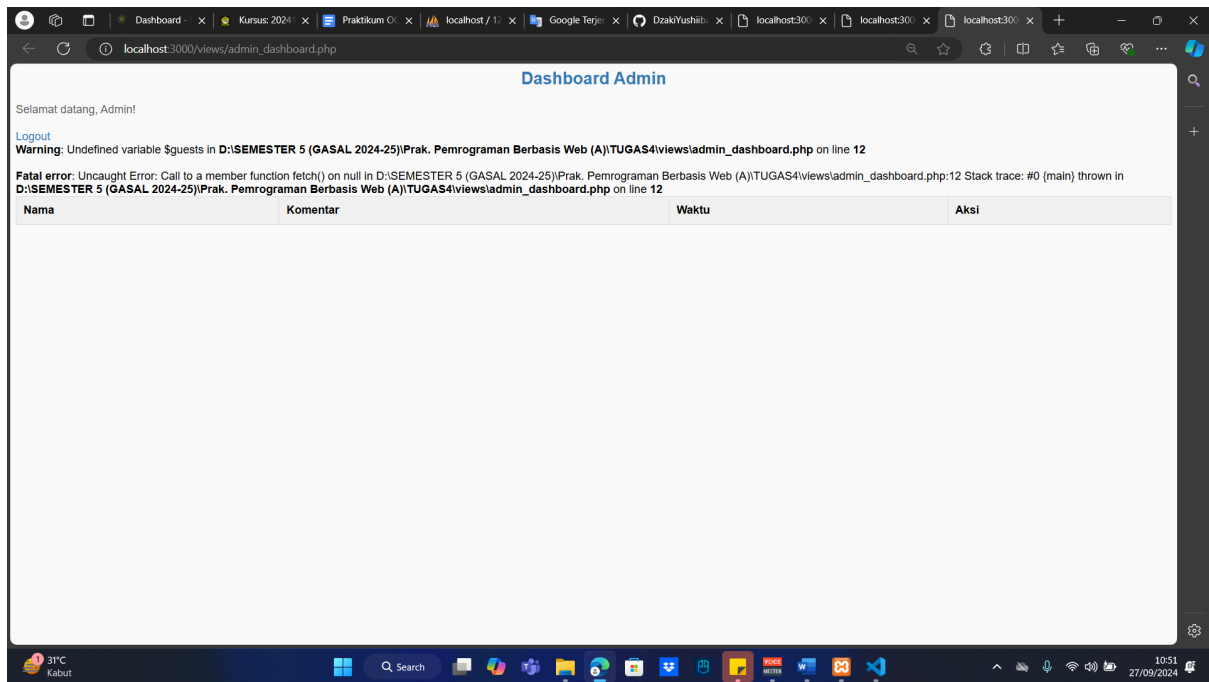


## Code admin\_login.php

```
<link rel="stylesheet" type="text/css" href="/css/style3.css">
<h2>Login Admin</h2>
<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

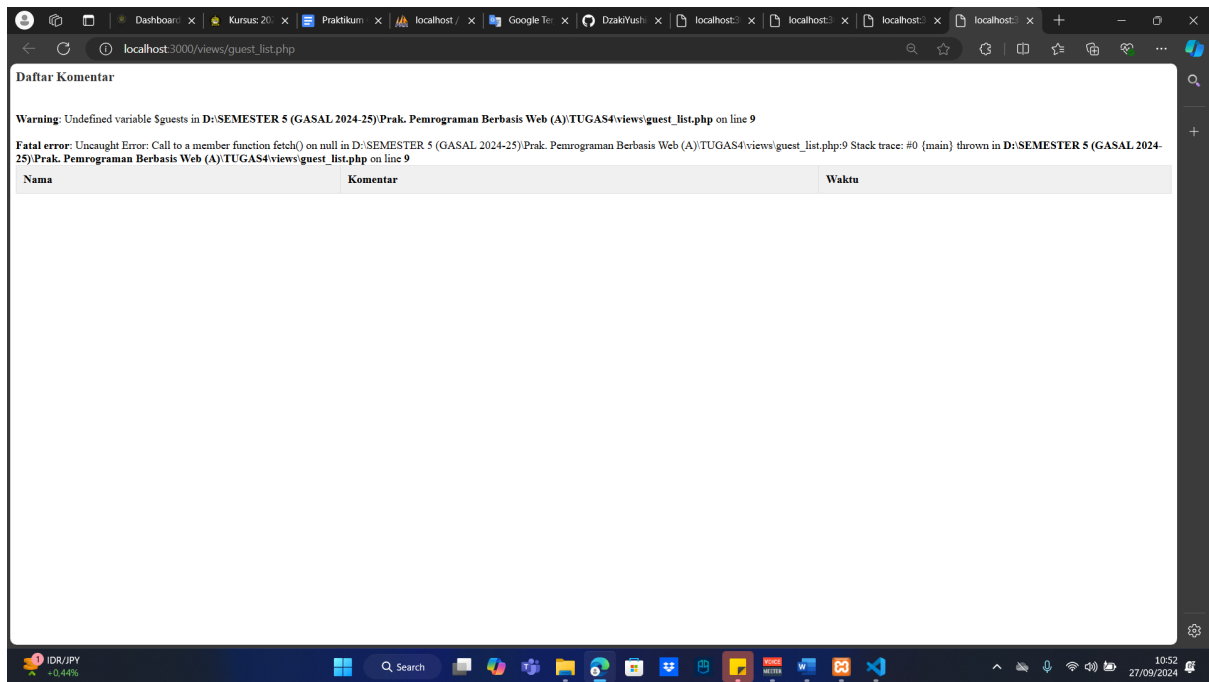




Code admin\_dashboard.php

```
<h2>Dashboard Admin</h2>
<link rel="stylesheet" type="text/css" href="/css/style.css">
<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>
```

Guest list



### Code Guest\_list.php

```
<link rel="stylesheet" type="text/css" href="/css/style2.css">
<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>
```

## 6. Fitur Tambahan

- **Login Admin:** Menggunakan hashing pada password dan session untuk otentikasi.

### Code login.php

```
<?php
session_start();
require_once 'controllers/AdminController.php';

$adminController = new AdminController();
```

```

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    if ($adminController->login($username, $password)) {
        $_SESSION['admin'] = $username;
        header("Location: admin.php");
    } else {
        $error = "Username atau password salah!";
    }
}

include 'views/admin_login.php';

if (isset($error)) {
    echo "<p style='color:red;'>$error</p>";
}
?>

```

- **CRUD pada Buku Tamu:** Admin dapat menambah, mengedit, dan menghapus komentar.

**Code edit.php**

```

<?php
session_start();
if (!isset($_SESSION['admin'])) {
    header("Location: login.php");
    exit();
}

require_once 'controllers/GuestController.php';

$guestController = new GuestController();

if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $stmt = $guestController->readAll();
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
        if ($row['id'] == $id) {
            $name = $row['name'];
            $comment = $row['comment'];
            break;
        }
    }
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $comment = $_POST['comment'];
    $guestController->update($id, $name, $comment);
    header("Location: admin.php");
}

```

```

?>

<h2>Edit Komentar</h2>
<form action="edit.php" method="post">
  <input type="hidden" name="id" value="<?php echo $id; ?>">
  <label for="name">Nama:</label><br>
  <input type="text" name="name" value="<?php echo htmlspecialchars($name); ?>"
required><br><br>
  <label for="comment">Komentar:</label><br>
  <textarea name="comment" rows="5" required><?php echo
htmlspecialchars($comment); ?></textarea><br><br>
  <input type="submit" value="Update">
</form>

```

### Index.php

```

<?php
require_once 'controllers/GuestController.php';

$guestController = new GuestController();

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $name = $_POST['name'];
  $comment = $_POST['comment'];
  $guestController->create($name, $comment);
  header("Location: index.php");
}

$guests = $guestController->readAll();

include 'views/guest_form.php';
include 'views/guest_list.php';
?>

```

### Login.php

```

<?php
session_start();
require_once 'controllers/AdminController.php';

$adminController = new AdminController();

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $username = $_POST['username'];
  $password = $_POST['password'];

  if ($adminController->login($username, $password)) {
    $_SESSION['admin'] = $username;
    header("Location: admin.php");
  } else {
    $error = "Username atau password salah!";
  }
}
}

```

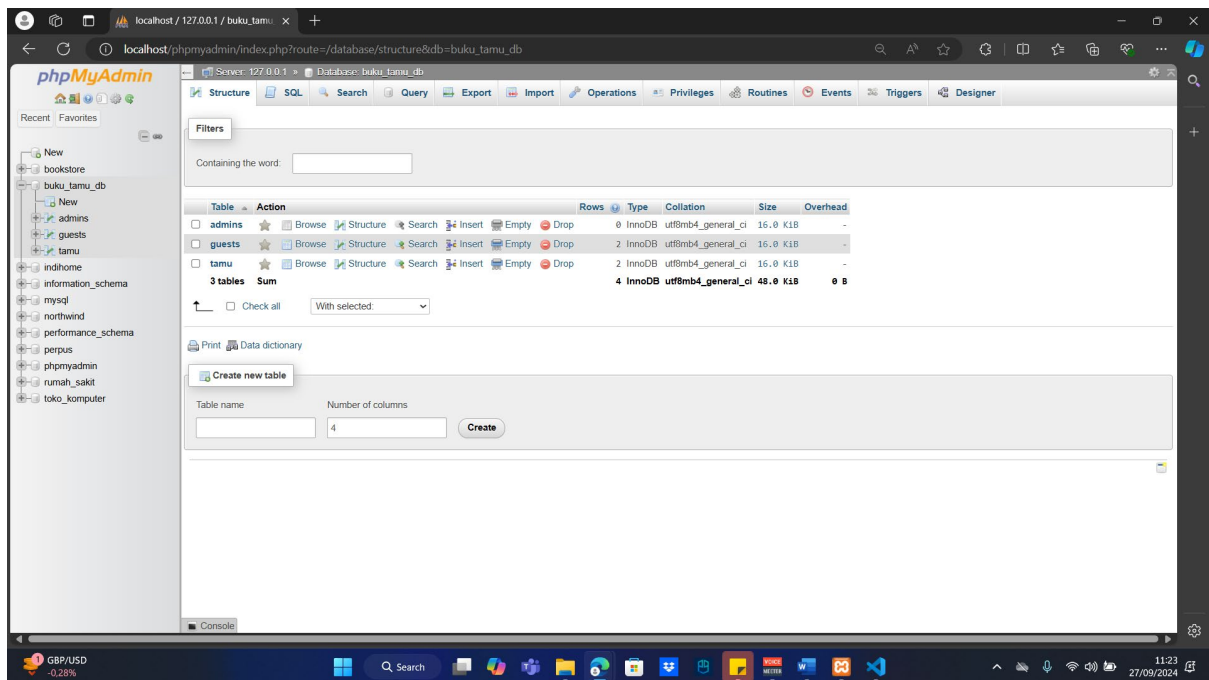
```
include 'views/admin_login.php';

if (isset($error)) {
    echo "<p style='color:red;'>$error</p>";
}
?>
```

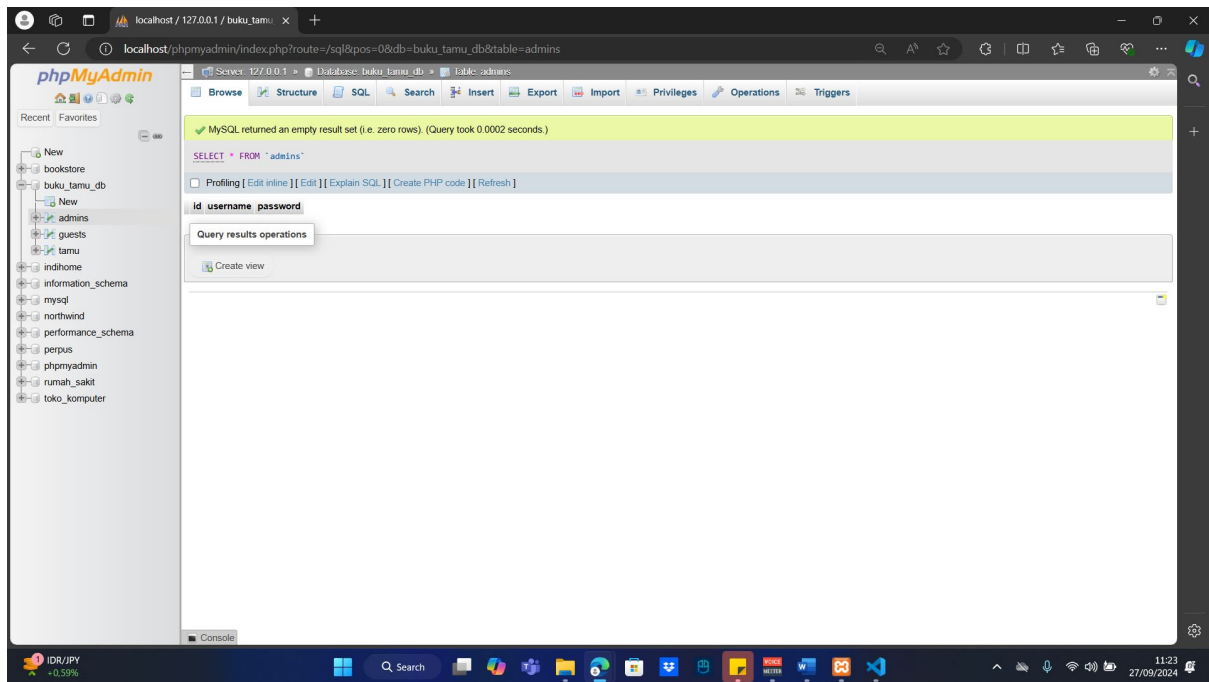
## Logout.php

```
<?php
session_start();
session_unset();
session_destroy();
header("Location: login.php");
?>
```

## Database



## Tabel admins



## Tabel guests

