

Praktikum Pemrograman Berbasis Web
Untuk Memenuhi Tugas Ke-4



Dosen

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

Disusun Oleh

Dio Prasetyo (4522210148)

Fakultas Teknik

Program Studi Teknik Informatika

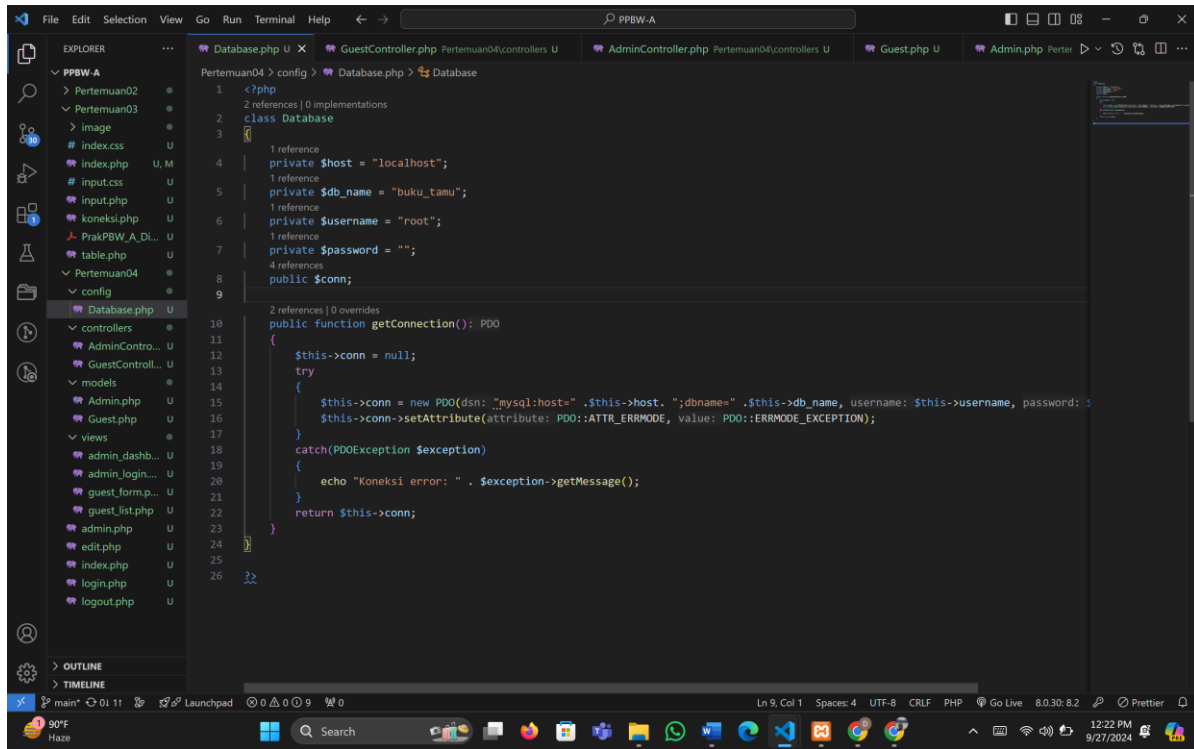
Universitas Pancasila

Tahun Akademik 2024 / 2025

Jakarta

A. Source Code

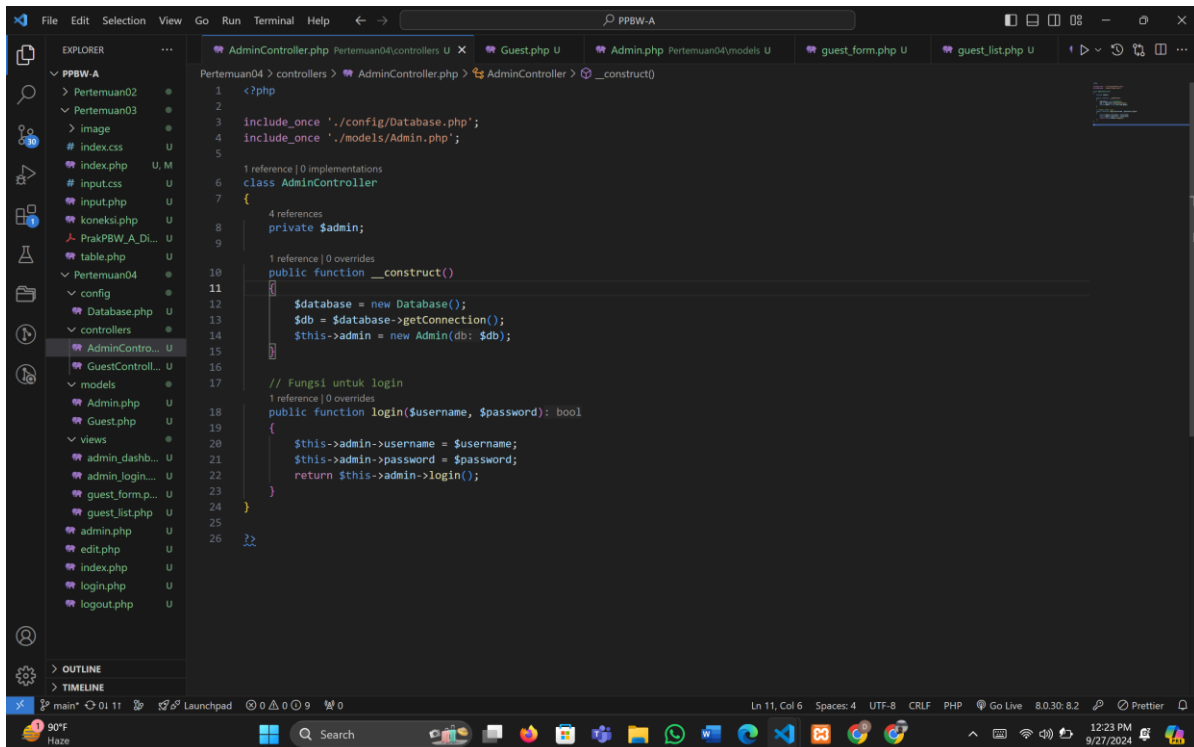
I. Config / Database.php



The screenshot shows the Visual Studio Code editor with the 'Database.php' file open. The Explorer sidebar on the left shows the project structure for 'PPBW-A', including folders for 'Pertemuan02', 'Pertemuan03', 'image', 'index.php', 'input.php', 'koneksi.php', 'PrakPBW_A_Di...', 'table.php', 'Pertemuan04', 'config', 'Database.php', 'controllers', 'models', 'views', and 'admin.php'. The main editor area displays the code for 'Database.php' with line numbers 1 through 26. The code defines a 'Database' class with private properties for host, db_name, username, and password, and a public static property '\$conn'. It includes a 'getConnection()' method that attempts to establish a PDO connection and returns it, or throws an exception if it fails. The status bar at the bottom indicates the file is at line 9, column 1, with 4 spaces, UTF-8 encoding, and CRLF line endings.

```
1 <?php
2 2 references | 0 implementations
3 class Database
4
5 1 reference
6 private $host = "localhost";
7 1 reference
8 private $db_name = "buku_tamu";
9 1 reference
10 private $username = "root";
11 1 reference
12 private $password = "";
13 4 references
14 public static $conn;
15
16 2 references | 0 overrides
17 public function getConnection(): PDO
18 {
19     $this->conn = null;
20     try
21     {
22         $this->conn = new PDO(dsn: "mysql:host=". $this->host. ";dbname=". $this->db_name, username: $this->username, password: $this->password);
23         $this->conn->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
24     }
25     catch(PDOException $exception)
26     {
27         echo "Koneksi error: " . $exception->getMessage();
28     }
29     return $this->conn;
30 }
```

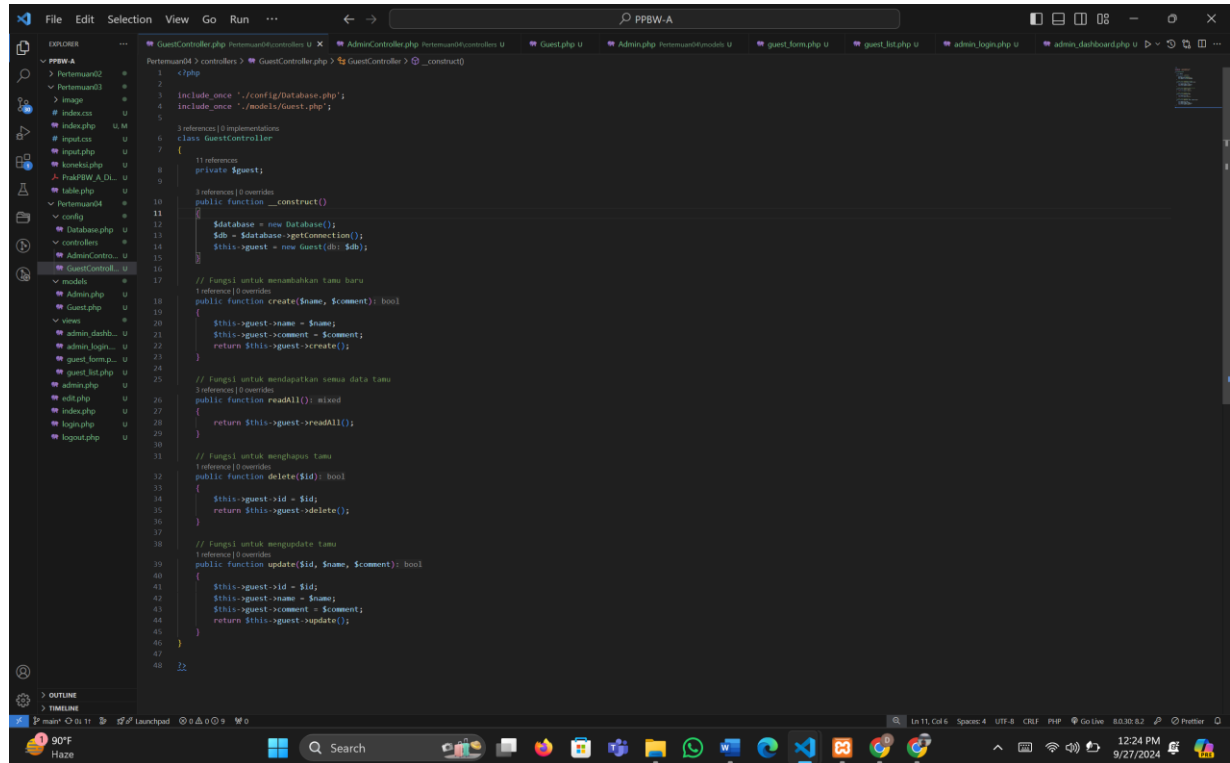
II. Controllers / AdminController.php



The screenshot shows the Visual Studio Code editor with the 'AdminController.php' file open. The Explorer sidebar on the left shows the project structure for 'PPBW-A', including folders for 'Pertemuan02', 'Pertemuan03', 'image', 'index.php', 'input.php', 'koneksi.php', 'PrakPBW_A_Di...', 'table.php', 'Pertemuan04', 'config', 'Database.php', 'controllers', 'models', 'views', and 'admin.php'. The main editor area displays the code for 'AdminController.php' with line numbers 1 through 26. The code defines an 'AdminController' class with a private property '\$admin'. It includes a '__construct()' method that initializes the database connection and the admin object. It also includes a 'login()' method that takes a username and password as input and returns a boolean value indicating the success of the login attempt. The status bar at the bottom indicates the file is at line 11, column 6, with 4 spaces, UTF-8 encoding, and CRLF line endings.

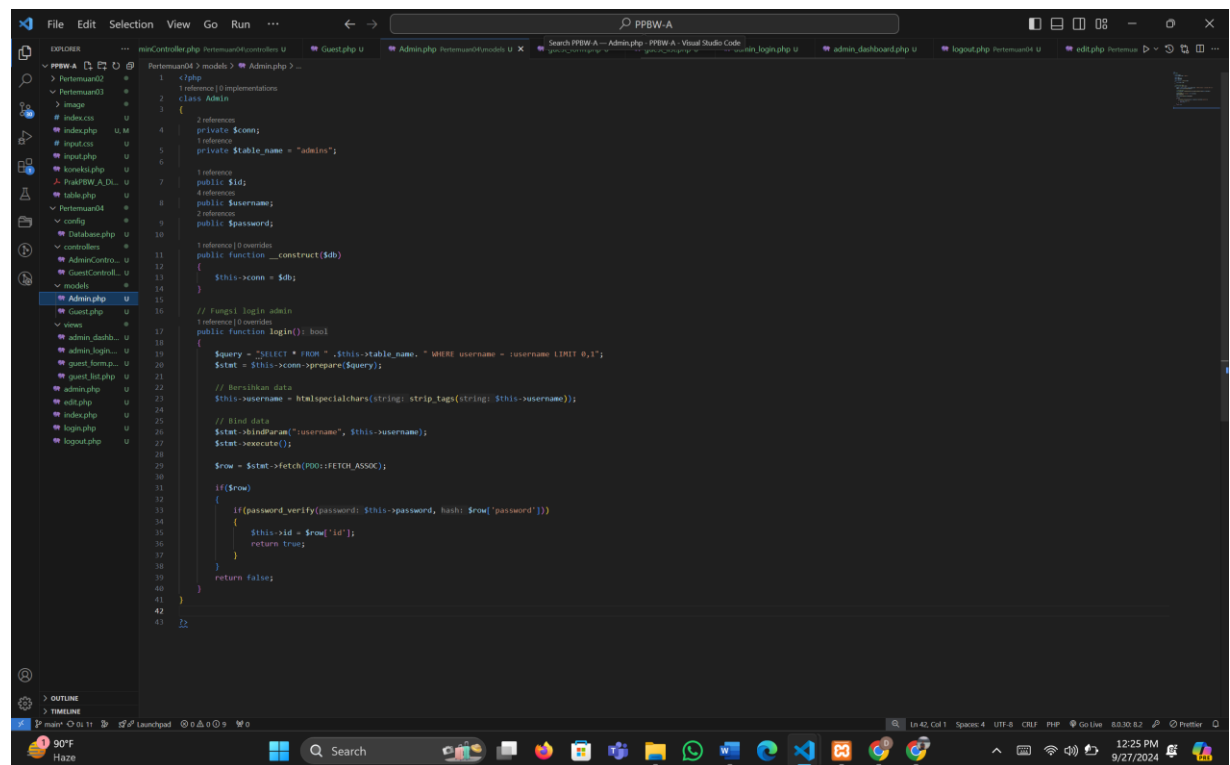
```
1 <?php
2
3 include_once './config/Database.php';
4 include_once './models/Admin.php';
5
6 1 reference | 0 implementations
7 class AdminController
8 {
9
10 4 references
11 private $admin;
12
13 1 reference | 0 overrides
14 public function __construct()
15 {
16     $database = new Database();
17     $db = $database->getConnection();
18     $this->admin = new Admin($db);
19 }
20
21 // Fungsi untuk login
22 1 reference | 0 overrides
23 public function login($username, $password): bool
24 {
25     $this->admin->username = $username;
26     $this->admin->password = $password;
27     return $this->admin->login();
28 }
29 }
```

III. Controlers / GuestController.php



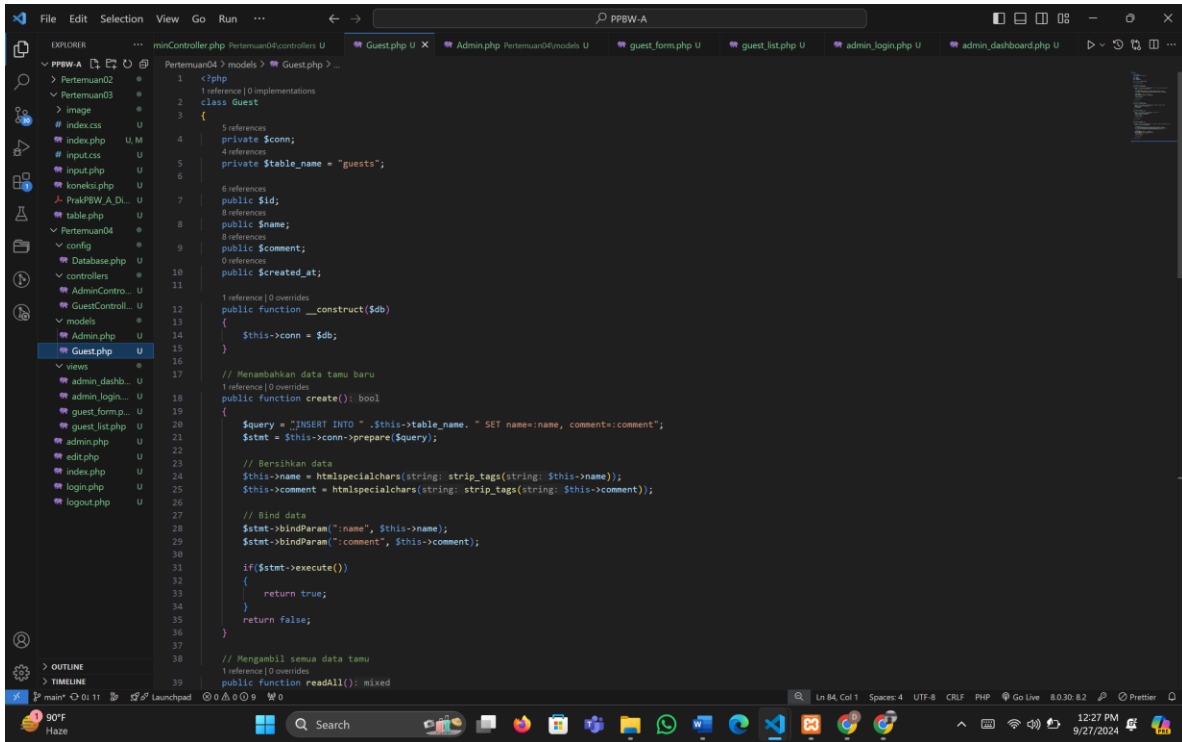
```
1 <?php
2
3 include_once './config/database.php';
4 include_once './models/Guest.php';
5
6 //reference: 0 implementations
7 class GuestController
8 {
9     //reference
10     private $guest;
11
12     //reference: 0 overrides
13     public function __construct()
14     {
15         $database = new Database();
16         $db = $database->getConnection();
17         $this->guest = new Guest($db);
18     }
19
20     // Fungsi untuk menambahkan tamu baru
21     //reference: 0 overrides
22     public function create($name, $comment): bool
23     {
24         $this->guest->name = $name;
25         $this->guest->comment = $comment;
26         return $this->guest->create();
27     }
28
29     // Fungsi untuk mendapatkan semua data tamu
30     //reference: 0 overrides
31     public function readAll(): mixed
32     {
33         return $this->guest->readAll();
34     }
35
36     // Fungsi untuk menghapus tamu
37     //reference: 0 overrides
38     public function delete($id): bool
39     {
40         $this->guest->id = $id;
41         return $this->guest->delete();
42     }
43
44     // Fungsi untuk mengupdate tamu
45     //reference: 0 overrides
46     public function update($id, $name, $comment): bool
47     {
48         $this->guest->id = $id;
49         $this->guest->name = $name;
50         $this->guest->comment = $comment;
51         return $this->guest->update();
52     }
53 }
```

IV. models / Admin.php

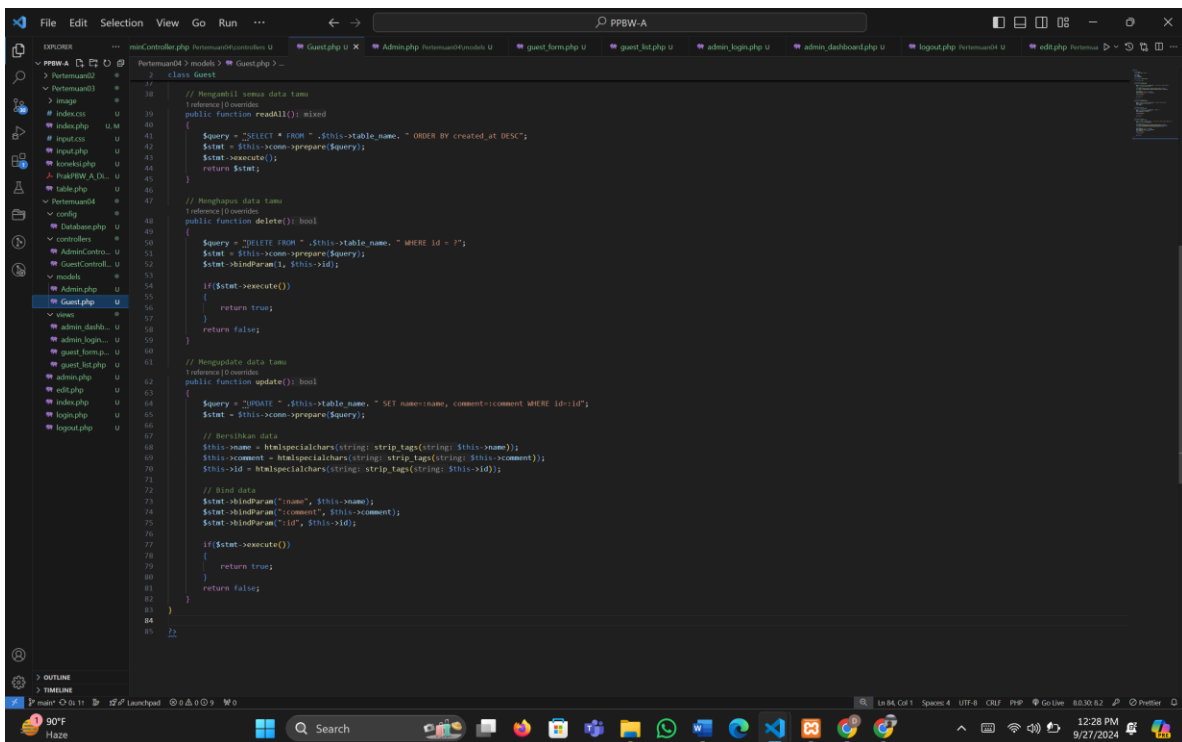


```
1 <?php
2
3 //reference: 0 implementations
4 class Admin
5 {
6     //reference
7     private $conn;
8     //reference
9     private $table_name = "admins";
10
11     //reference
12     public $id;
13     //reference
14     public $username;
15     //reference
16     public $password;
17
18     //reference: 0 overrides
19     public function __construct($db)
20     {
21         $this->conn = $db;
22     }
23
24     // Fungsi login admin
25     //reference: 0 overrides
26     public function login(): bool
27     {
28         $query = "SELECT * FROM " . $this->table_name . " WHERE username = ?";
29         $stmt = $this->conn->prepare($query);
30
31         // Bersihkan data
32         $this->username = htmlspecialchars(strip_tags($this->username));
33
34         // Bind data
35         $stmt->bind_param("s", $this->username);
36         $stmt->execute();
37
38         $row = $stmt->fetch(PDO::FETCH_ASSOC);
39
40         if($row)
41         {
42             if(password_verify($this->password, hash($row['password'])))
43             {
44                 $this->id = $row['id'];
45                 return true;
46             }
47             return false;
48         }
49     }
50 }
```

V. model / Guest.php

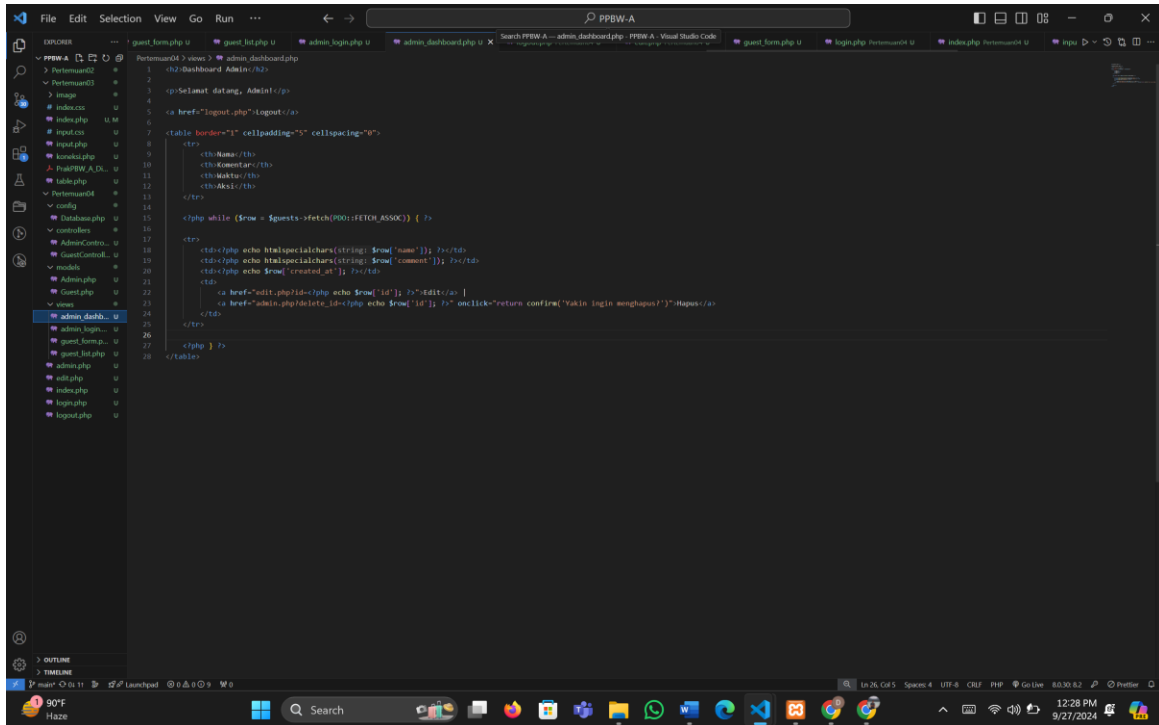


```
1 <?php
2 class Guest
3 {
4     private $conn;
5     private $table_name = "guests";
6
7     public $id;
8     public $name;
9     public $comment;
10    public $created_at;
11
12    1 reference [0 overrides]
13    public function __construct($db)
14    {
15        $this->conn = $db;
16    }
17
18    // Menambahkan data tamu baru
19    1 reference [0 overrides]
20    public function create(): bool
21    {
22        $query = "INSERT INTO " . $this->table_name . " SET name=:name, comment=:comment";
23        $stmt = $this->conn->prepare($query);
24
25        // Bersihkan data
26        $this->name = htmlspecialchars(strip_tags($this->name));
27        $this->comment = htmlspecialchars(strip_tags($this->comment));
28
29        // Bind data
30        $stmt->bindParam(":name", $this->name);
31        $stmt->bindParam(":comment", $this->comment);
32
33        if($stmt->execute())
34        {
35            return true;
36        }
37        return false;
38    }
39
40    // Mengambil semua data tamu
41    1 reference [0 overrides]
42    public function readAll(): mixed
43    {
```



```
44        $query = "SELECT * FROM " . $this->table_name . " ORDER BY created_at DESC";
45        $stmt = $this->conn->prepare($query);
46        $stmt->execute();
47        return $stmt;
48    }
49
50    // Menghapus data tamu
51    1 reference [0 overrides]
52    public function delete(): bool
53    {
54        $query = "DELETE FROM " . $this->table_name . " WHERE id = ?";
55        $stmt = $this->conn->prepare($query);
56        $stmt->bindParam(1, $this->id);
57
58        if($stmt->execute())
59        {
60            return true;
61        }
62        return false;
63    }
64
65    // Mengupdate data tamu
66    1 reference [0 overrides]
67    public function update(): bool
68    {
69        $query = "UPDATE " . $this->table_name . " SET name=:name, comment=:comment WHERE id=:id";
70        $stmt = $this->conn->prepare($query);
71
72        // Bersihkan data
73        $this->name = htmlspecialchars(strip_tags($this->name));
74        $this->comment = htmlspecialchars(strip_tags($this->comment));
75        $this->id = htmlspecialchars(strip_tags($this->id));
76
77        // Bind data
78        $stmt->bindParam(":name", $this->name);
79        $stmt->bindParam(":comment", $this->comment);
80        $stmt->bindParam(":id", $this->id);
81
82        if($stmt->execute())
83        {
84            return true;
85        }
86        return false;
87    }
88 }
```

V. views / admin_dashboard.php



```
1 <?php
2 <h2>Dashboard Admin</h2>
3 <p>Selamat datang, Admin!</p>
4 <a href="login.php">Logout</a>
5
6 <table border="1" cellpadding="5" cellspacing="0">
7
8     <tr>
9         <th>Name</th>
10        <th>Comment</th>
11        <th>Action</th>
12    </tr>
13
14    <tr>
15        <td colspan="3">
16            <?php while ($row = $query->fetch(PDO::FETCH_ASSOC)) { ?>
17
18                <tr>
19                    <td><?php echo htmlspecialchars($row['name']); ?></td>
20                    <td><?php echo htmlspecialchars($row['comment']); ?></td>
21                    <td><?php echo $row['comment_id']; ?></td>
22                </tr>
23
24                <tr>
25                    <td colspan="3">
26                        <a href="edit.php?id=<?php echo $row['id']; ?>">Edit</a> |
27                        <a href="admin.php/delete_id=<?php echo $row['id']; ?>">Delete</a>
28                    </td>
29                </tr>
30            </tr>
31        </td>
32    </tr>
33    </table>
```

VI. views / admin_login.php

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

margin: 0;

```
padding: 0;
justify-content: center;
align-items: center;
height: 100vh;
}
```

```
.container {
  background-color: orange;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  padding: 30px;
  width: 100%;
  max-width: 400px;
  margin: 100px 500px;
}
```

```
.kata {
  text-align: center;
  margin-bottom: 20px;
}
```

```
h2 {
  color: green;
  margin: 0;
}
```

```
.form {
  margin: 10px 0;
```

```
}
```

```
.form label {  
    display: block;  
    margin-bottom: 5px;  
    color: #555;  
}
```

```
.form input[type="text"],  
.form textarea {  
    width: 100%;  
    padding: 8px;  
    margin-bottom: 15px;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    box-sizing: border-box;  
}
```

```
.form input[type="password"],  
.form textarea {  
    width: 100%;  
    padding: 8px;  
    margin-bottom: 15px;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    box-sizing: border-box;  
}
```

```
.form textarea {  
    resize: vertical;  
}
```

```
.form input[type="submit"] {  
    background-color: #4CAF50;  
    color: white;  
    border: none;  
    padding: 10px 20px;  
    border-radius: 4px;  
    cursor: pointer;  
    font-size: 16px;  
    transition: background-color 0.3s;  
    width: 100%;  
}
```

```
.form input[type="submit"]:hover {  
    background-color: #45a049;  
}
```

```
</style>
```

```
<body>
```

```
<div class="container">
```

```
<div class="kata">
```

```
<h2>Login Admin</h2>
```

```
</div>
```

```
<div class="form">
```



```
<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>
</div>
</div>

</body>
</html>
```

VII. views / guest_form.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form</title>
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    margin: 0;
    padding: 0;
    justify-content: center;
```

```
    align-items: center;
    min-height: 100vh;
}
```

```
.container {
    background-color: orange;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    padding: 30px;
    width: 100%;
    max-width: 400px;
    margin: 100px 500px;
}
```

```
.kata {
    text-align: center;
    margin-bottom: 20px;
}
```

```
h2 {
    color: green;
    margin: 0;
}
```

```
.form {
    margin: 10px 0;
}
```

```
.form label {  
    display: block;  
    margin-bottom: 5px;  
    color: #555;  
}
```

```
.form input[type="text"],  
.form textarea {  
    width: 100%;  
    padding: 8px;  
    margin-bottom: 15px;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    box-sizing: border-box;  
}
```

```
.form textarea {  
    resize: vertical;  
}
```

```
.form input[type="submit"] {  
    background-color: #4CAF50;  
    color: white;  
    border: none;  
    padding: 10px 20px;  
    border-radius: 4px;  
    cursor: pointer;  
    font-size: 16px;
```

```
        transition: background-color 0.3s;
        width: 100%;
    }

    .form input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>

<body>
    <div class="container">
        <div class="kata">
            <h2>Selamat Datang di Buku Tamu</h2>
        </div>

        <div class="form">
            <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>
        </div>
    </div>
</body>
</html>
```

VIII. models / guest_list.php

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

</head>

<style>

  body {

    font-family: Arial, sans-serif;

    line-height: 1.6;

    margin: 0;

    padding: 20px;

    background-color: #f4f4f4;

  }

  h3 {

    color: #333;

  }

  table {

    width: 100%;

    border-collapse: collapse;

    margin-top: 20px;

    background-color: #fff;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    text-align: center;
```

```

    }
    th, td {
        padding: 12px;
        text-align: left;
        border-bottom: 1px solid #ddd;
    }
    th {
        background-color: #f2f2f2;
        font-weight: bold;
        color: #333;
    }
    tr:hover {
        background-color: #f5f5f5;
    }
</style>

<body>
<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 } ?>

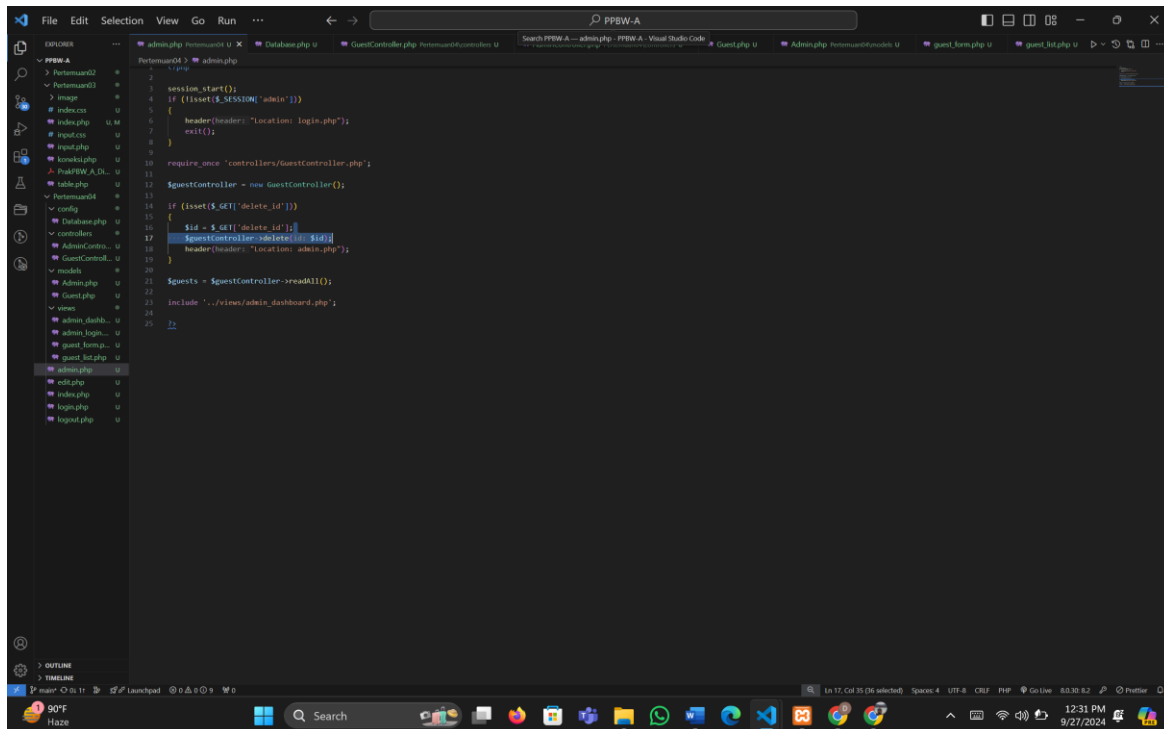
</table>

</body>

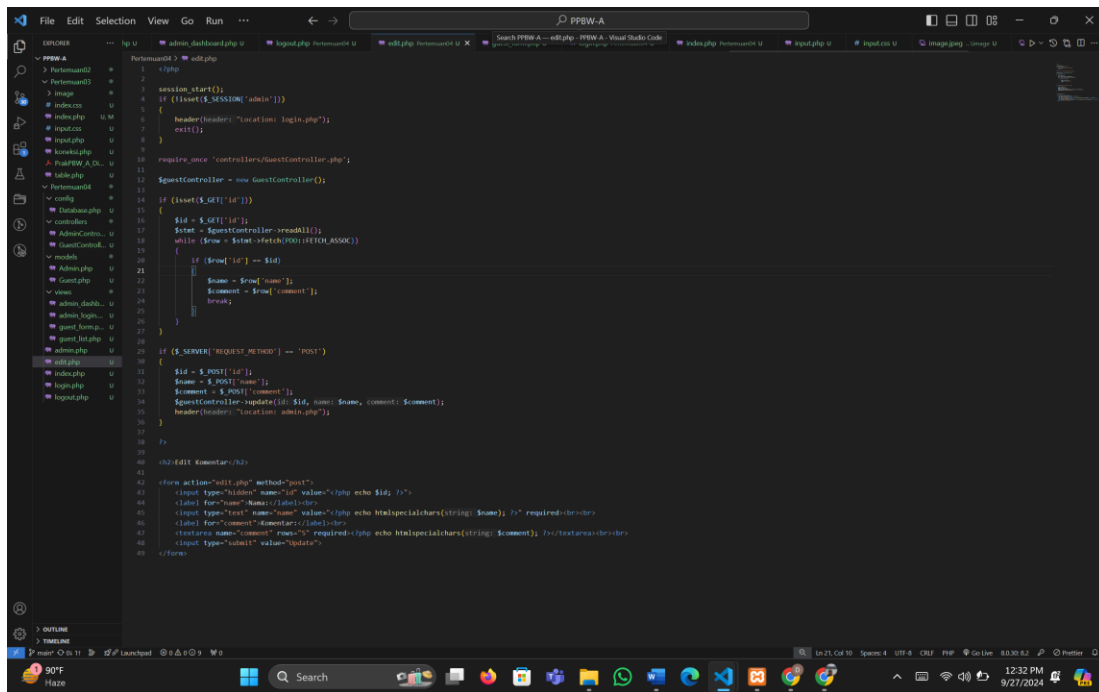
</html>

```

IX. admin.php

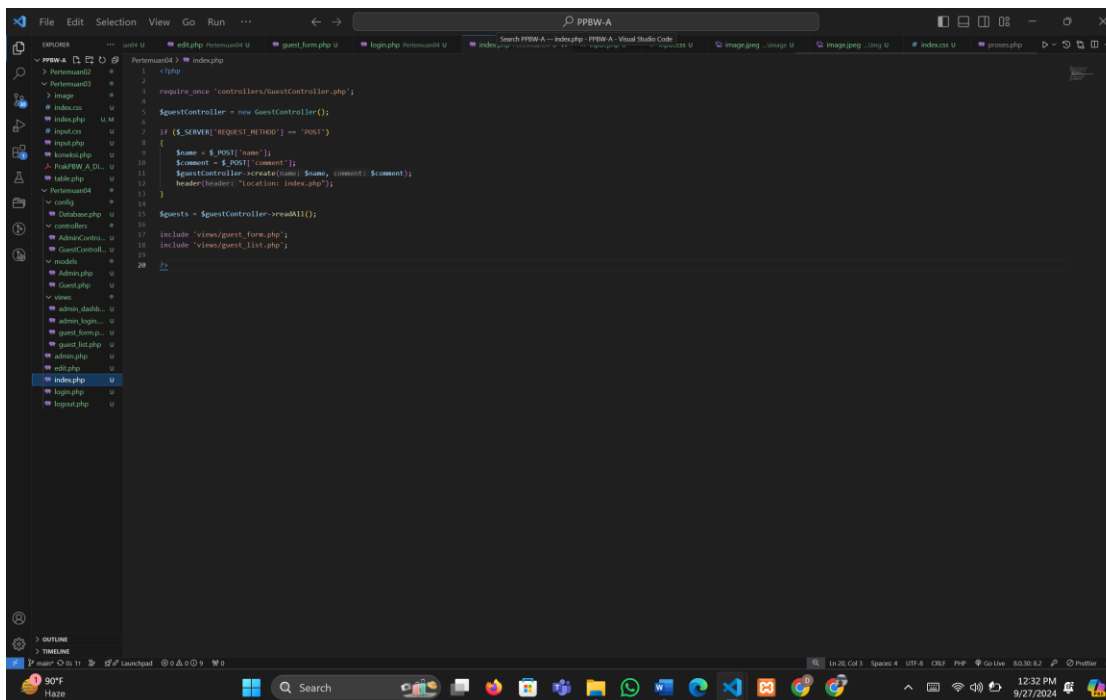


X. edit.php



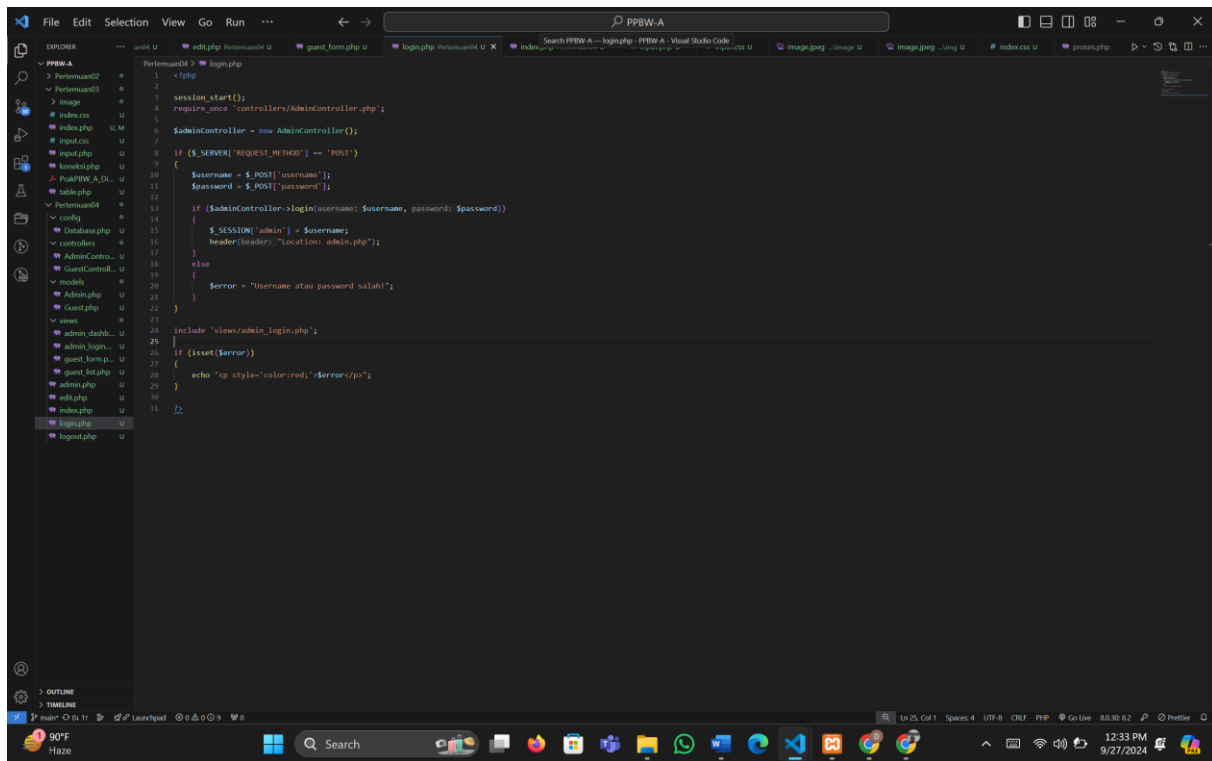
```
1 <?php
2
3 session_start();
4 if (!isset($_SESSION['admin']))
5 {
6     header(header: "location: login.php");
7     exit();
8 }
9
10 require_once 'controllers/guestController.php';
11
12 $guestController = new GuestController();
13
14 if (isset($_GET['id']))
15 {
16     $id = $_GET['id'];
17     $stat = $guestController->readAll();
18     while ($row = $stat->fetch(PDO::FETCH_ASSOC))
19     {
20         if ($row['id'] == $id)
21         {
22             $name = $row['name'];
23             $comment = $row['comment'];
24             break;
25         }
26     }
27
28     if ($_SERVER['REQUEST_METHOD'] == "POST")
29     {
30         $id = $_POST['id'];
31         $name = $_POST['name'];
32         $comment = $_POST['comment'];
33         $guestController->update($id, $name, $comment);
34         header(header: "location: admin.php");
35     }
36 }
37
38 //
39
40 //edit Edit Controller/2/
41
42 <form action="edit.php" method="post">
43 <input type="hidden" name="id" value="{php echo $id; }"/>
44 <label for="name">Name:</label><br>
45 <input type="text" name="name" value="{php echo htmlspecialchars(string: $name); }" required/><br>
46 <label for="comment">Comment:</label><br>
47 <textarea name="comment" rows="5" required/>{php echo htmlspecialchars(string: $comment); }</textarea><br>
48 <input type="submit" value="Update">
49 </form>
```

XI. index.php



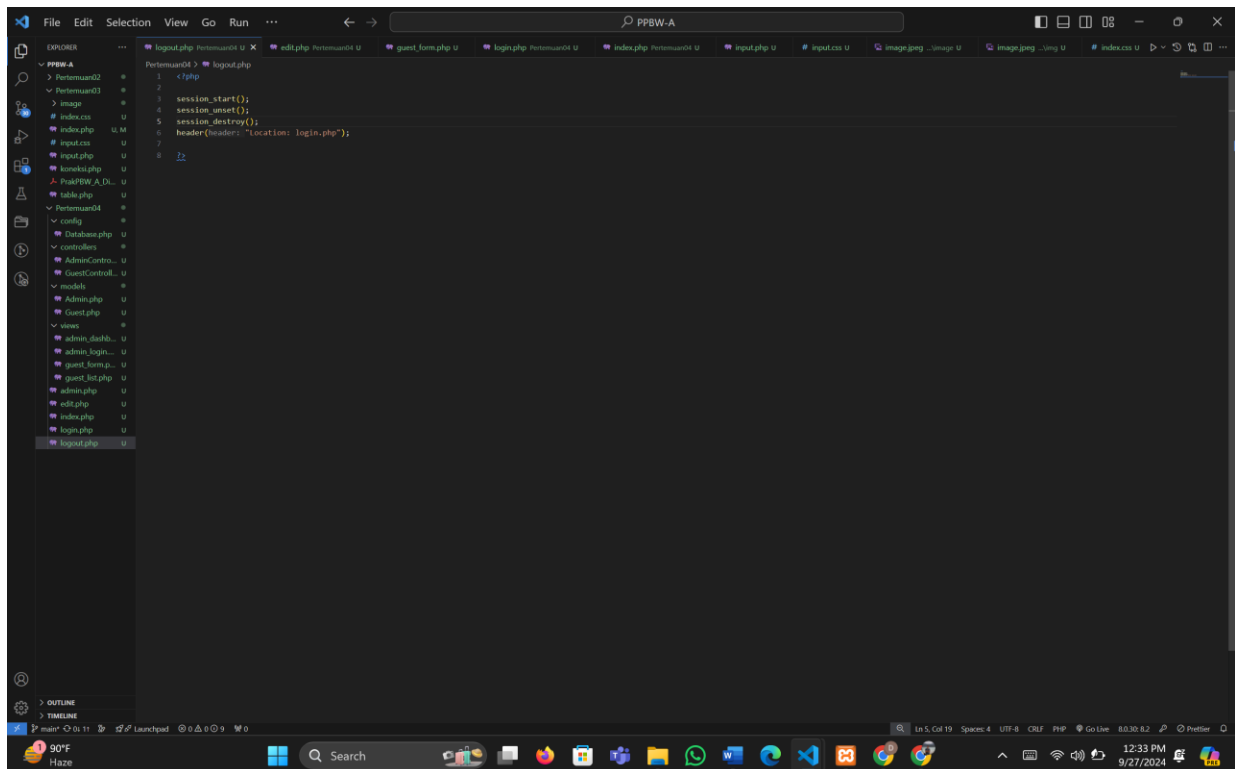
```
1 <?php
2
3 require_once 'controllers/guestController.php';
4
5 $guestController = new GuestController();
6
7 if ($_SERVER['REQUEST_METHOD'] == "POST")
8 {
9     $name = $_POST['name'];
10     $comment = $_POST['comment'];
11     $guestController->create($name, $comment);
12     header(header: "location: index.php");
13 }
14
15 $guests = $guestController->readAll();
16
17 include 'views/guest_form.php';
18 include 'views/guest_list.php';
19
20 //
```


XII. Login.php



```
1 <?php
2
3 session_start();
4 require_once 'controllers/AdminController.php';
5
6 $adminController = new AdminController();
7
8 if ($_SERVER['REQUEST_METHOD'] == 'POST')
9 {
10     $username = $_POST['username'];
11     $password = $_POST['password'];
12
13     if ($adminController->login($username, $password))
14     {
15         $_SESSION['admin'] = $username;
16         header(header: "location: admin.php");
17     }
18     else
19     {
20         $error = "Username atau password salah!";
21     }
22 }
23
24 include 'views/admin_login.php';
25
26 if (isset($error))
27 {
28     echo "<p style='color:red;'>$error</p>";
29 }
30
31 ?>
```

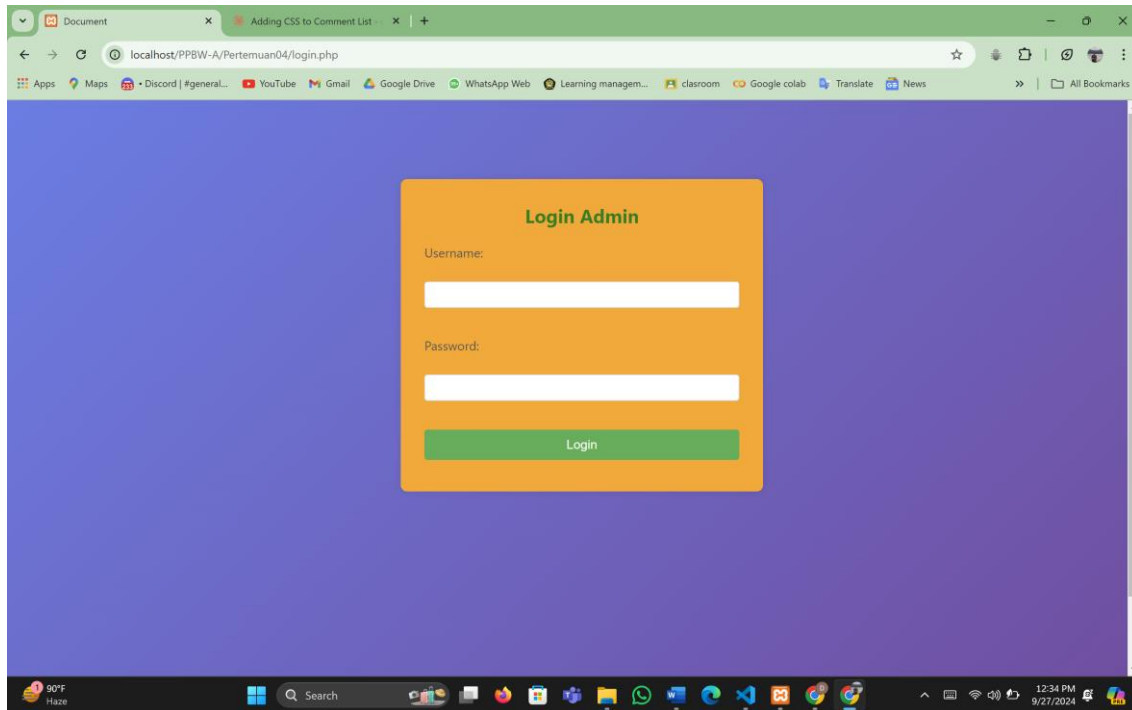
XIII. Logout.php



```
1 <?php
2
3 session_start();
4 session_unset();
5 session_destroy();
6 header(header: "location: login.php");
7
8 ?>
```

B. Screen Shoot Hasil Running Program

A. Login Admin



Document x Adding CSS to Comment List x +

localhost/PPBW-A/Pertemuan04/login.php

Apps Maps Discord | #general... YouTube Gmail Google Drive WhatsApp Web Learning managem... classroom Google colab Translate News All Bookmarks

Login Admin

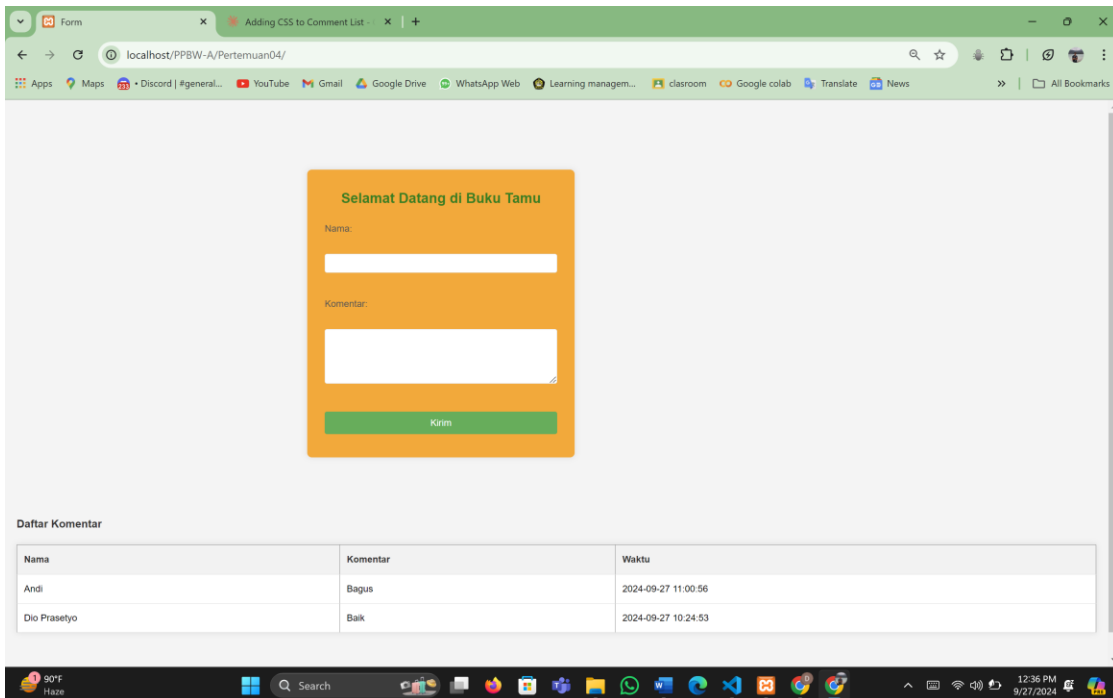
Username:

Password:

Login

90°F Haze Search 12:34 PM 9/27/2024

B. Login User



Form x Adding CSS to Comment List x +

localhost/PPBW-A/Pertemuan04/

Apps Maps Discord | #general... YouTube Gmail Google Drive WhatsApp Web Learning managem... classroom Google colab Translate News All Bookmarks

Selamat Datang di Buku Tamu

Nama:

Komentar:

Kirim

Daftar Komentar

Nama	Komentar	Waktu
Andi	Bagus	2024-09-27 11:00:56
Dio Prasetyo	Baik	2024-09-27 10:24:53

90°F Haze Search 12:36 PM 9/27/2024

C. PHP MYAdmin (database)

The screenshot displays the phpMyAdmin web interface in a browser window. The address bar shows the URL: `localhost/phpmyadmin/index.php?route=/database/structure&db=buku_tamu`. The interface is titled "Database: buku_tamu" and features a top navigation bar with tabs for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers, and Designer. The "Structure" tab is active, showing a list of tables in the "buku_tamu" database. The left sidebar contains a tree view of the database structure, including "buku_tamu", "admins", "guests", "buku_tamu_db", "information_schema", "ktm", "ktm2", "mysql", "performance_schema", "phpmyadmin", and "test". The main content area shows a table with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The table lists two tables: "admins" and "guests". Below the table, there is a "Create new table" section with fields for "Table name" and "Number of columns" (set to 4), and a "Create" button. The bottom of the screen shows a Windows taskbar with the date and time: 12:37 PM, 9/27/2024.

Table	Action	Rows	Type	Collation	Size	Overhead
admins	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KIB	-
guests	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KIB	-
2 tables	Sum	3	InnoDB	utf8mb4_general_ci	32.0 KIB	0 B

localhost / 127.0.0.1 / buku_tamu x +

localhost/phpmyadmin/index.php?route=/sql&db=buku_tamu&table=admins&pos=0

Server: 127.0.0.1 x Database: buku_tamu x Table: admins

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds)

```
SELECT * FROM `admins`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	id	username	PASSWORD
<input type="checkbox"/>	1	admin	admin123

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

90°F Haze 12:38 PM 9/27/2024

localhost / 127.0.0.1 / buku_tamu x +

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=buku_tamu&table=guests

Server: 127.0.0.1 x Database: buku_tamu x Table: guests

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds)

```
SELECT * FROM `guests`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	name	comment	created_at
<input type="checkbox"/>	1	Dio Prasetyo	Baik	2024-09-27 10:24:53
<input type="checkbox"/>	2	Andi	Bagus	2024-09-27 11:00:56

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

90°F Haze 12:38 PM 9/27/2024