

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



Oleh :

Muhammad Rachel Fathan Idzhany

4522210071

**Dosen:**

**Adi Wahyu Pribadi, M.Kom.**

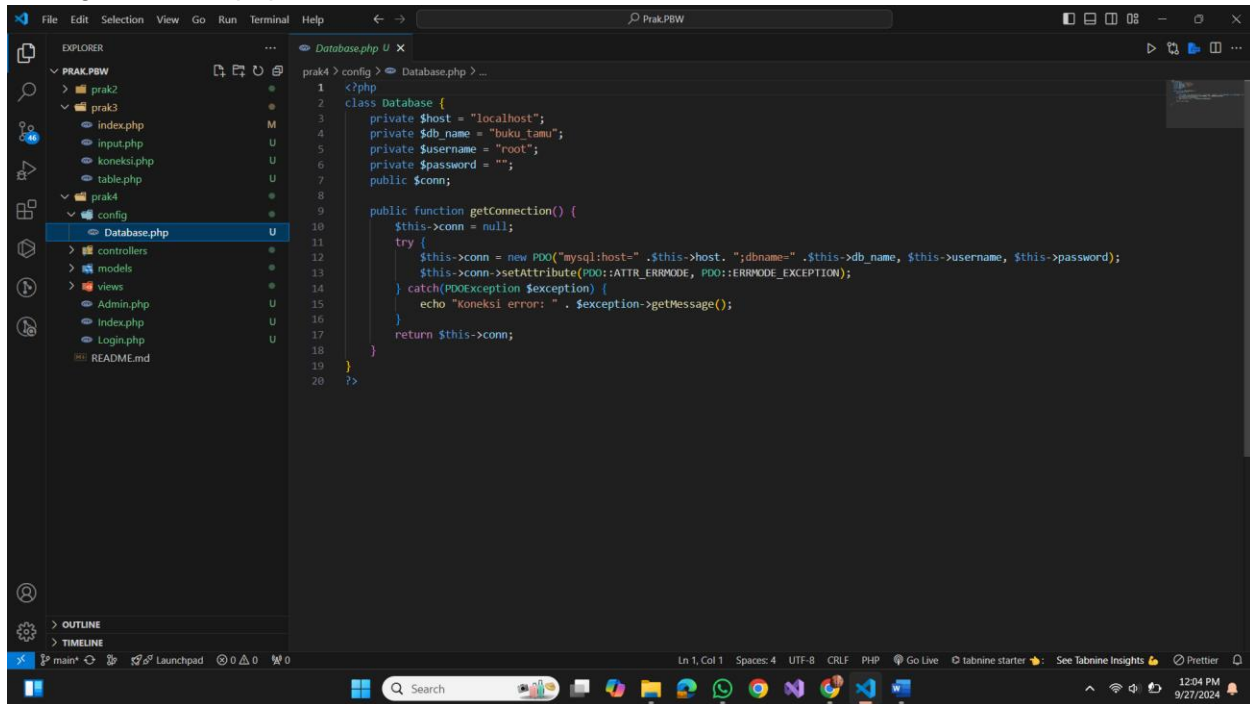
**S1 – Teknik Informatika**

**Fakultas Teknik Universitas Pancasila**

**2024/2025**

Program:

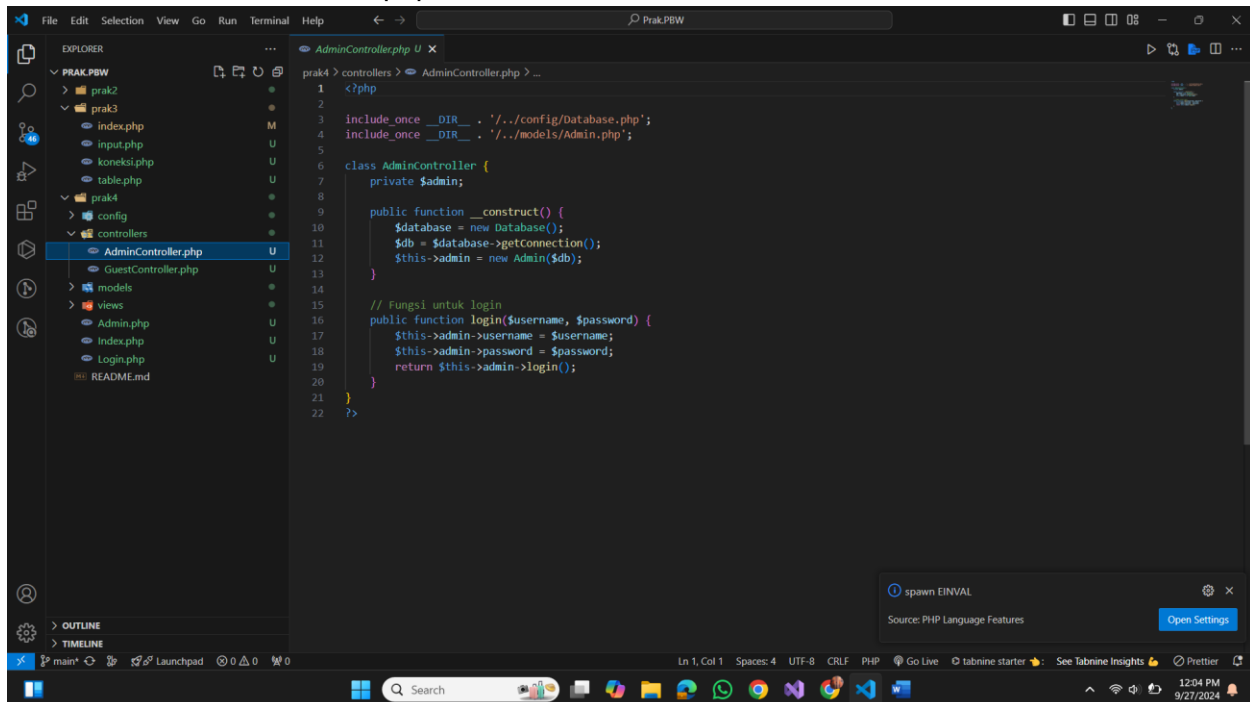
/config/Database.php



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project structure. The file `Database.php` is selected under the `config` directory. The main editor window displays the PHP code for `Database.php`, which defines a `Database` class with private properties for host, database name, username, and password, and a public `getConnection()` method.

```
1 <?php
2 class Database {
3     private $host = "localhost";
4     private $db_name = "buku_tamu";
5     private $username = "root";
6     private $password = "";
7     public $conn;
8
9     public function getConnection() {
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 $exception) {
15             echo "Koneksi error: " . $exception->getMessage();
16         }
17         return $this->conn;
18     }
19 }
20 ?>
```

/Controllers/AdminController.php

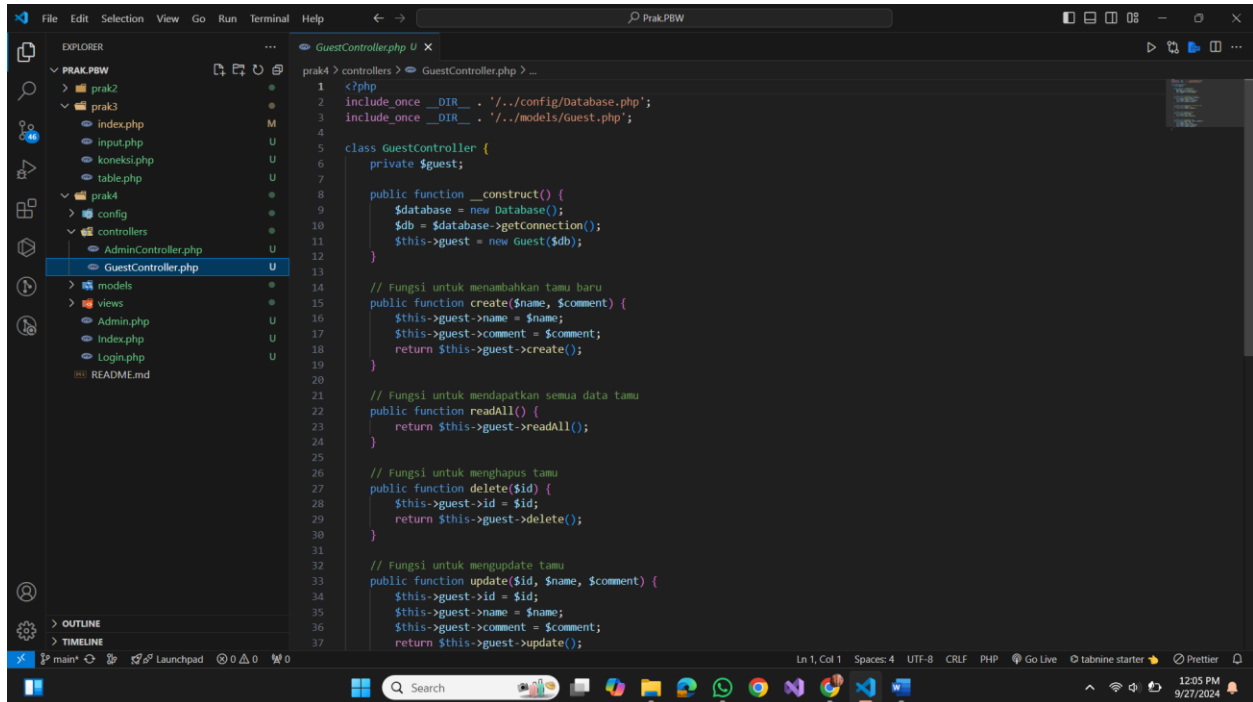


The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The file `AdminController.php` is selected under the `controllers` directory. The main editor window displays the PHP code for `AdminController.php`, which includes the `Database.php` file and defines an `AdminController` class with a `__construct()` method and a `login()` method.

```
1 <?php
2
3 include_once __DIR__ . '/../config/Database.php';
4 include_once __DIR__ . '/../models/Admin.php';
5
6 class AdminController {
7     private $admin;
8
9     public function __construct() {
10         $database = new Database();
11         $db = $database->getConnection();
12         $this->admin = new Admin($db);
13     }
14
15     // Fungsi untuk login
16     public function login($username, $password) {
17         $this->admin->username = $username;
18         $this->admin->password = $password;
19         return $this->admin->login();
20     }
21 }
22 ?>
```

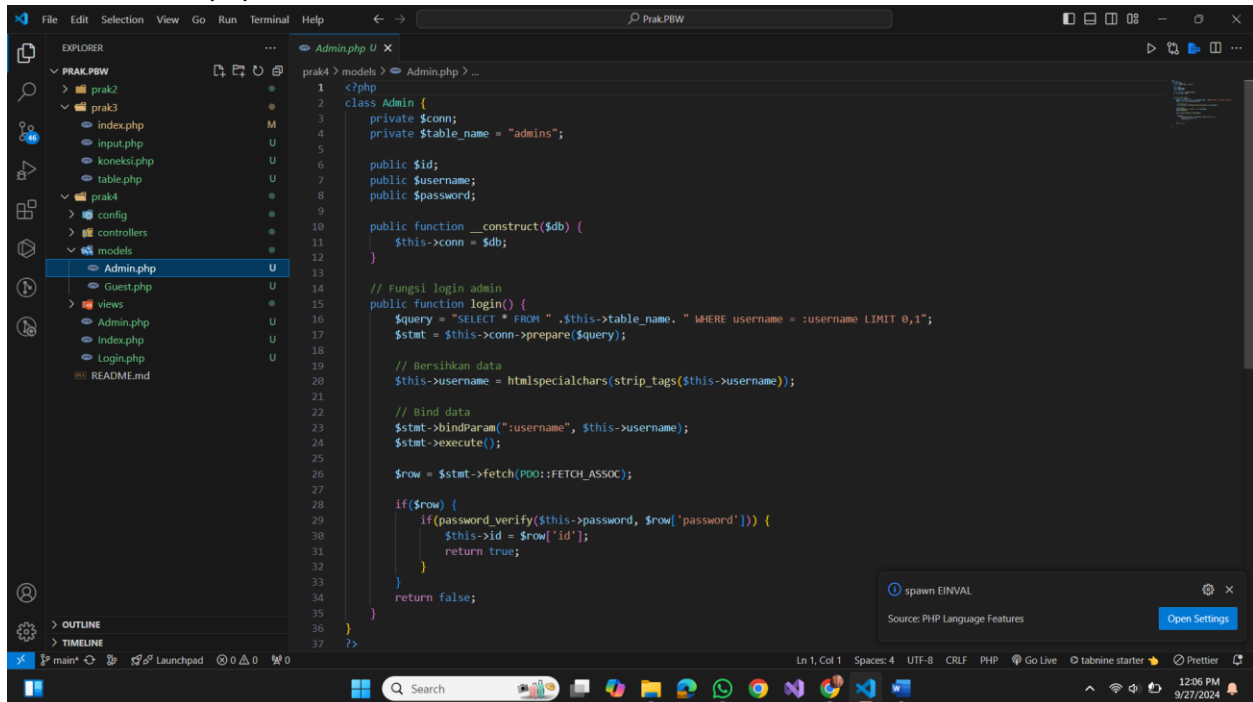
A notification box in the bottom right corner indicates a "spawn EINTR" error from the "Source: PHP Language Features" and provides a link to "Open Settings".

/controller/GuestController.php



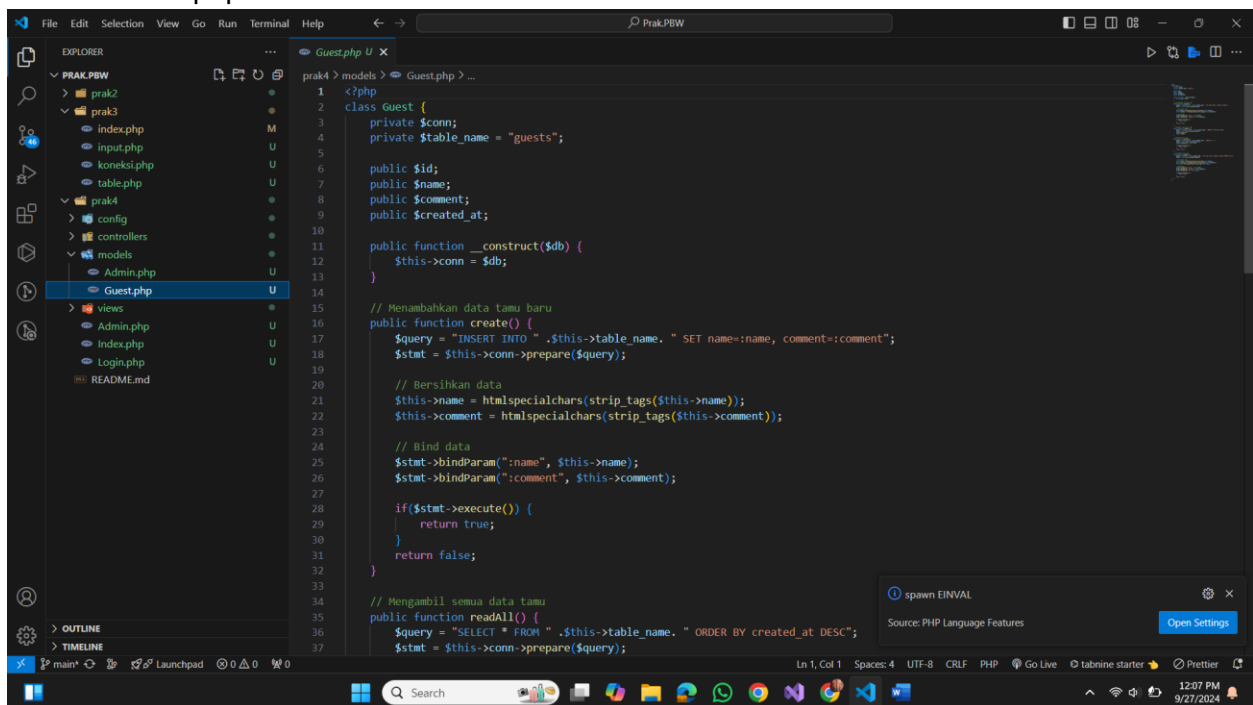
```
1 <?php
2 include_once __DIR__ . '/../config/Database.php';
3 include_once __DIR__ . '/../models/Guest.php';
4
5 class GuestController {
6     private $guest;
7
8     public function __construct() {
9         $database = new Database();
10        $db = $database->getConnection();
11        $this->guest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

## /models/Admin.php



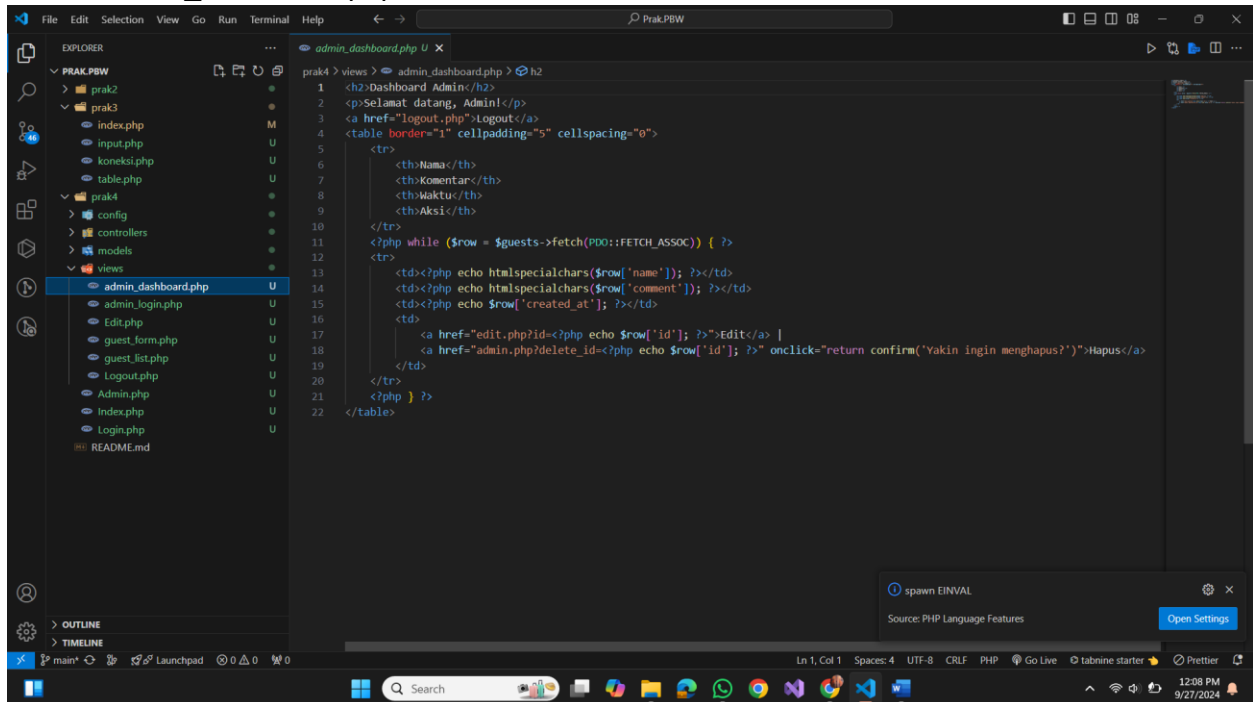
```
1 <?php
2 class Admin {
3     private $conn;
4     private $table_name = "admins";
5
6     public $id;
7     public $username;
8     public $password;
9
10    public function __construct($db) {
11        $this->conn = $db;
12    }
13
14    // Fungsi login admin
15    public function login() {
16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

## /Admin/Guest.php



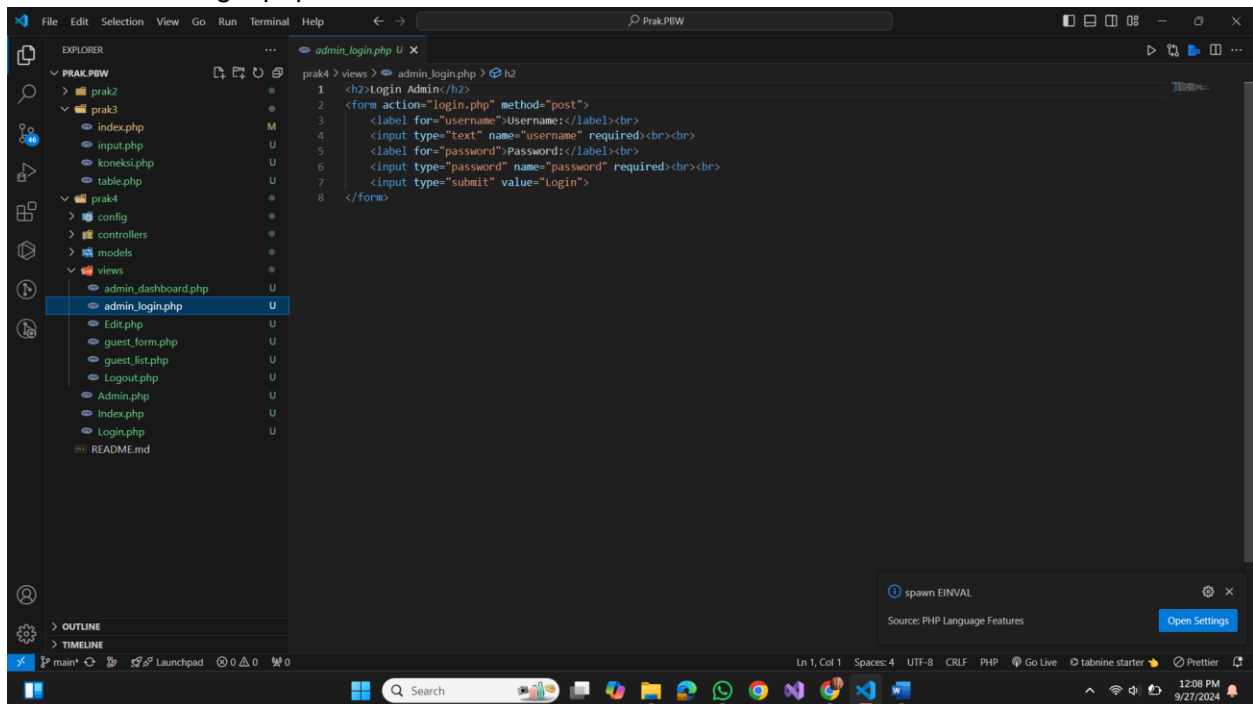
```
1 <?php
2 class Guest {
3     private $conn;
4     private $table_name = "guests";
5
6     public $id;
7     public $name;
8     public $comment;
9     public $created_at;
10
11    public function __construct($db) {
12        $this->conn = $db;
13    }
14
15    // Menambahkan data tamu baru
16  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

/views/ admin\_dashboard.php



```
1 <h2>Dashboard Admin</h2>
2 <p>Selamat datang, Admin!</p>
3 <a href="logout.php">Logout</a>
4 <table border="1" cellpadding="5" cellspacing="0">
5   <tr>
6     <th>Nama</th>
7     <th>Komentar</th>
8     <th>Waktu</th>
9     <th>Aksi</th>
10  </tr>
11  <?php while ($row = $guests->fetch(PDO::FETCH_ASSOC)) { ?>
12    <tr>
13      <td><?php echo htmlspecialchars($row['name']); ?></td>
14      <td><?php echo htmlspecialchars($row['comment']); ?></td>
15      <td><?php echo $row['created_at']; ?></td>
16      <td>
17        <a href="edit.php?id=<?php echo $row['id']; ?>">Edit</a> |
18        <a href="admin.php?delete_id=<?php echo $row['id']; ?>" onclick="return confirm('Yakin ingin menghapus?')">Hapus</a>
19      </td>
20    </tr>
21  <?php } ?>
22 </table>
```

/views/admin\_login.php



```
1 <h2>Login Admin</h2>
2 <form action="login.php" method="post">
3   <label for="username">Username:</label><br>
4   <input type="text" name="username" required><br>
5   <label for="password">Password:</label><br>
6   <input type="password" name="password" required><br>
7   <input type="submit" value="Login">
8 </form>
```

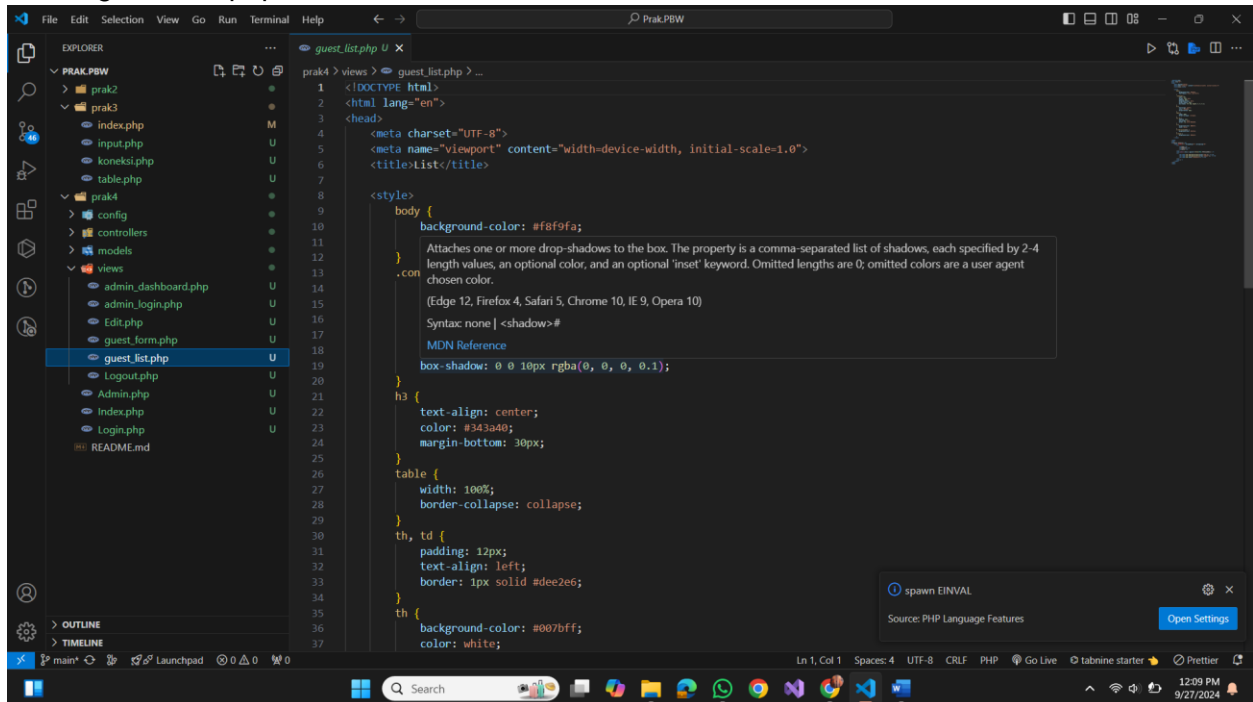
/views/Edit.php

```
1 <?php
2 session_start();
3 if (!isset($_SESSION['admin'])) {
4     header("Location: login.php");
5     exit();
6 }
7
8 require_once 'controllers/GuestController.php';
9
10 $guestController = new GuestController();
11
12 if (isset($_GET['id'])) {
13     $id = $_GET['id'];
14     $stmt = $guestController->find($id);
15     while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
16         if ($row['id'] == $id) {
17             $name = $row['name'];
18             $comment = $row['comment'];
19             break;
20         }
21     }
22 }
23
24 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
25     $id = $_POST['id'];
26     $name = $_POST['name'];
27     $comment = $_POST['comment'];
28     $guestController->update($id, $name, $comment);
29     header("Location: admin.php");
30 }
31
32 ?>
33
34 <h2>Edit Komentar</h2>
35 <form action="edit.php" method="post">
36     <input type="hidden" name="id" value="<?php echo $id; ?>">
37     <label for="name">Nama:</label><br>
```

/views/guest\_form.php

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Guest Form</title>
7
8 <style>
9     body {
10         background-color: #f8f9fa;
11         font-family: Arial, sans-serif;
12         padding: 20px;
13     }
14     .container {
15         max-width: 600px;
16         margin: 50px auto;
17         padding: 20px;
18         background-color: #fff;
19         border-radius: 10px;
20         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
21     }
22     h2 {
23         text-align: center;
24         color: #343a40;
25         margin-bottom: 20px;
26     }
27     label {
28         font-weight: bold;
29         display: block;
30         margin-bottom: 5px;
31     }
32     input[type="text"], textarea {
33         width: 100%;
34         padding: 10px;
35         margin-bottom: 20px;
36         border: 1px solid #ced4da;
37         border-radius: 5px;
```

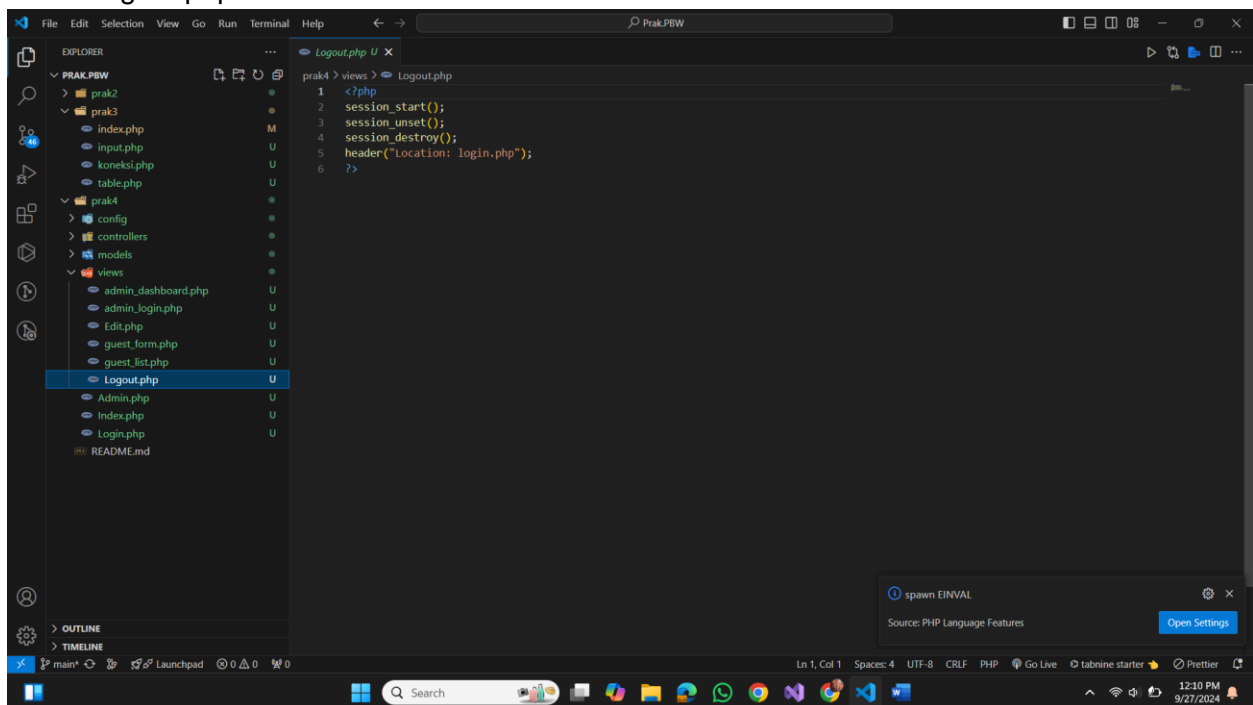
/views/guest\_list.php



The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays a project structure with folders 'prak2', 'prak3', and 'prak4'. The 'prak4' folder is expanded, showing subfolders 'config', 'controllers', 'models', and 'views'. The 'views' folder is further expanded, listing files like 'admin\_dashboard.php', 'admin\_login.php', 'Edit.php', 'guest\_form.php', 'guest\_list.php' (which is selected), 'Logout.php', 'Admin.php', 'Index.php', and 'Login.php'. The main editor window displays the content of 'guest\_list.php', which is an HTML file. The code includes a DOCTYPE declaration, HTML and head tags, a meta charset, a viewport meta tag, a title 'List', and a style block. The style block contains CSS for a body background color, a container with a box shadow, and a table with specific styling. A tooltip is visible over the 'box-shadow' property, explaining its syntax and providing a reference. The status bar at the bottom indicates the current line and column (Ln 1, Col 1), the number of spaces (4), the encoding (UTF-8), and the file type (PHP).

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>List</title>
7
8 <style>
9     body {
10         background-color: #f8f9fa;
11     }
12     .container {
13         padding: 10px;
14         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
15     }
16     h3 {
17         text-align: center;
18         color: #343a40;
19         margin-bottom: 30px;
20     }
21     table {
22         width: 100%;
23         border-collapse: collapse;
24     }
25     th, td {
26         padding: 12px;
27         text-align: left;
28         border: 1px solid #dee2e6;
29     }
30     th {
31         background-color: #007bff;
32         color: white;
33     }
34
35
36
37
```

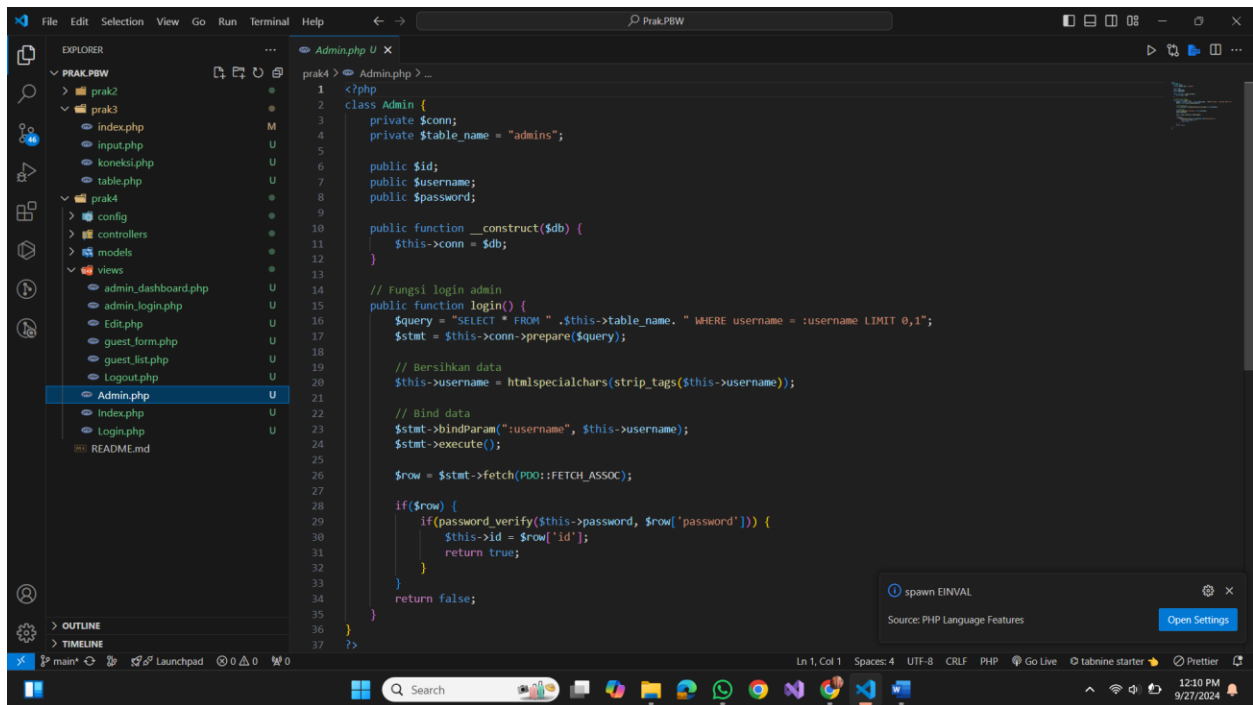
/views/logout.php



The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays the same project structure as the previous image. The 'prak4' folder is expanded, and the 'views' folder is further expanded. The file 'Logout.php' is now selected. The main editor window displays the content of 'Logout.php', which is a PHP file. The code includes a PHP opening tag, session management functions (session\_start, session\_unset, session\_destroy), and a header redirecting to 'login.php'. The status bar at the bottom indicates the current line and column (Ln 1, Col 1), the number of spaces (4), the encoding (UTF-8), and the file type (PHP).

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

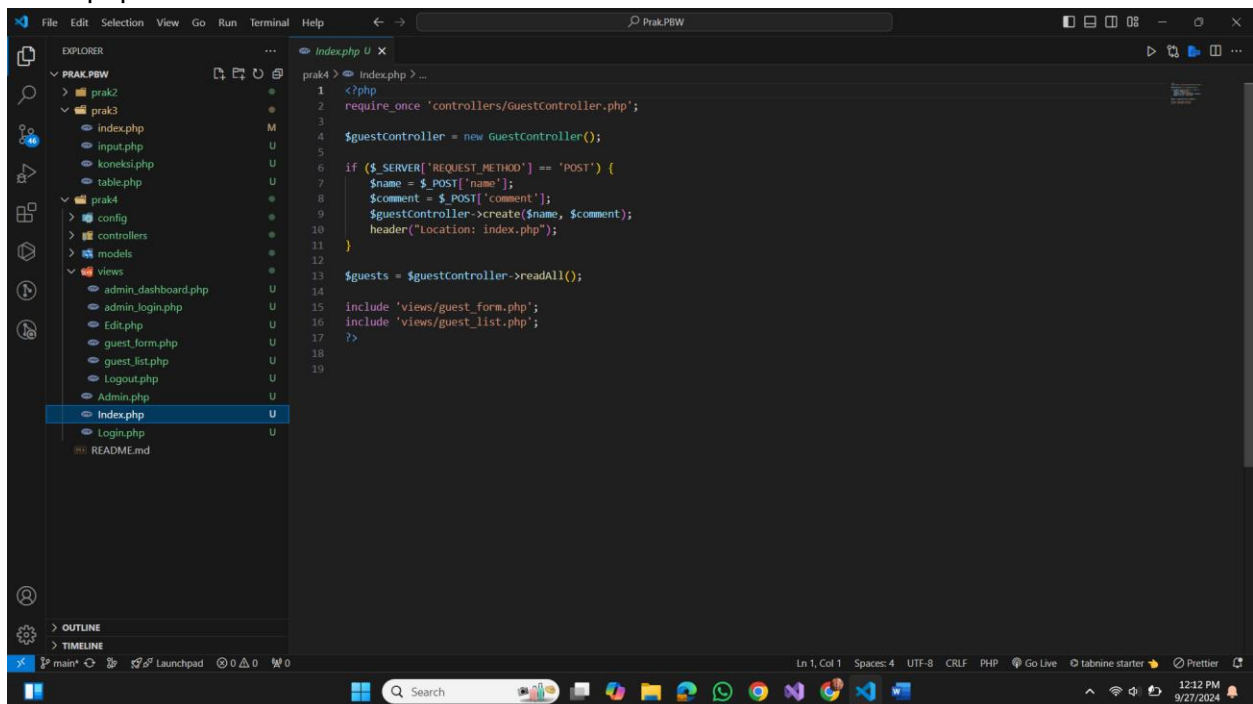
Admin.php



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project structure for 'PRAK.PBW'. The file 'Admin.php' is selected in the 'views' directory. The main editor window shows the PHP code for 'Admin.php', which defines an 'Admin' class with properties for connection, table name, ID, username, and password. It includes a constructor, a login function, and a password verification method. A status bar at the bottom indicates the current position is Line 1, Column 1.

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

## Index.php

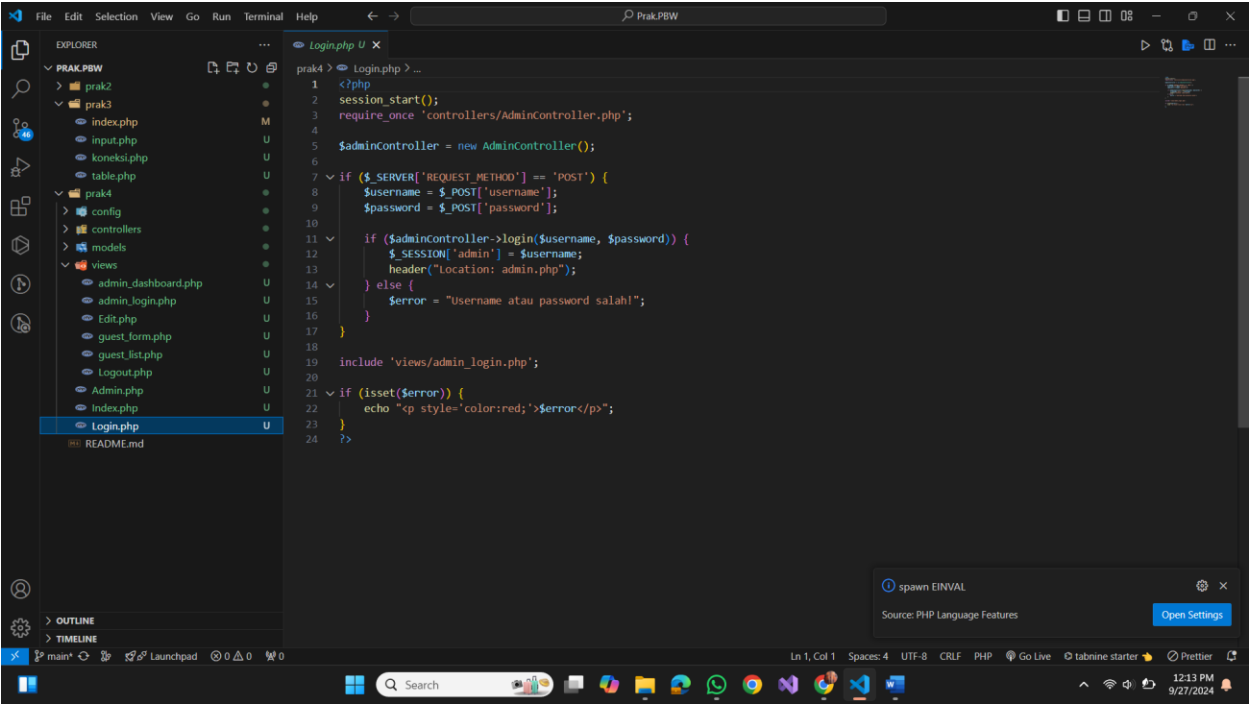


The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the same project structure. The file 'Index.php' is selected in the 'views' directory. The main editor window shows the PHP code for 'Index.php', which includes a require\_once statement for 'controllers/GuestController.php', creates a new 'GuestController' object, and handles a POST request to create a guest and redirect to 'index.php'. It also fetches all guests and includes the 'guest\_form.php' and 'guest\_list.php' views. A status bar at the bottom indicates the current position is Line 1, Column 1.

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

## Login.php





Database:

Server: 127.0.0.1 » Database: buku_tamu									
Struktur SQL Cari Kueri Ekspor Impor Operasi Hak Akses Routine Event Trigger									
Filters									
Mengandung kata: <input type="text"/>									
Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban			
<input type="checkbox"/> admins	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	utf8mb4_general_ci	16.0 KB	-			
<input type="checkbox"/> guests	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	2	InnoDB	utf8mb4_general_ci	16.0 KB	-			
2 tabel	Jumlah	3	InnoDB	utf8mb4_general_ci	32.0 KB	0 B			

Server: 127.0.0.1 » Database: buku\_tamu » Tabel: admins

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses

✓ Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0,0002 detik.)

`SELECT * FROM `admins``

Profil [ Edit dikotak ] [ Ubah ] [ Jelaskan SQL ] [ Buat kode PHP ] [ Segarkan ]

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini

Extra options

	id	username	password
<input type="checkbox"/> Ubah Salin Hapus	1	admin	\$2y\$10\$e0NRz6iVb6q5VhB8GzJ4JeQc2YJwZgQ8hGgFXc7V3K6...

Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

Server: 127.0.0.1 » Database: buku\_tamu » Tabel: guests

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi

✓ Menampilkan baris 0 - 1 (total 2, Pencarian dilakukan dalam 0,0005 detik.)

`SELECT * FROM `guests``

Profil [ Edit dikotak ] [ Ubah ] [ Jelaskan SQL ] [ Buat kode PHP ] [ Segarkan ]

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key: Tidak ada

Extra options

	id	name	comment	created_at
<input type="checkbox"/> Ubah Salin Hapus	1	fathan	haiii	2024-09-27 10:47:29
<input type="checkbox"/> Ubah Salin Hapus	2	mamat	apa yahh	2024-09-27 10:48:32

Hasil Run:

The image shows a web browser window with multiple tabs. The active tab is 'localhost / 127.0.0.1 / b...'. The browser address bar shows 'localhost:8080/index.php'. The web page has a light gray background. In the center, there is a white rectangular box with a thin gray border. Inside this box, the text 'Selamat Datang di Buku Tamu' is centered at the top. Below this text, there are two form fields. The first is labeled 'Nama:' and is a single-line text input. The second is labeled 'Komentar:' and is a multi-line text area. Below these fields is a blue button with the text 'Kirim' in white. Below the white box, the text 'Daftar Komentar' is centered. Underneath this text is a table with three columns: 'Nama', 'Komentar', and 'Waktu'. The table has a blue header row. There are two data rows. The first row shows 'mamat' in the 'Nama' column, 'apa yahh' in the 'Komentar' column, and '2024-09-27 10:48:32' in the 'Waktu' column. The second row shows 'fathan' in the 'Nama' column, 'haiii' in the 'Komentar' column, and '2024-09-27 10:47:29' in the 'Waktu' column. The Windows taskbar is visible at the bottom of the screen, showing the Start button, a search bar, and several application icons including File Explorer, Edge, WhatsApp, and others. The system clock in the bottom right corner shows '11:55 AM 9/27/2024'.