

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS WEB



Disusun Oleh :

Irma rpiana - 4522210051

Dosen:

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

S1-Teknik Informatika

Fakultas Teknik Universitas Pancasila

2024

Membuat Aplikasi Buku Dengan OOP/PDO/MVC

- /config/database.php/

```
Database.php X
config > Database.php > ...
1 <?php
2 class Database {
3     private $host = "localhost"; // atau IP server database
4     private $db_name = "nama_database";
5     private $username = "root"; // atau username lain jika diatur
6     private $password = ""; // jika tidak ada password, biarkan kosong
7     public $conn;
8
9     public function getConnection(): PDO {
10         $this->conn = null;
11
12         try {
13             $this->conn = new PDO(dsn: "mysql:host=" . $this->host . ";dbname=" . $this->db_name, username: $this->username, password: $this->password);
14             $this->conn->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
15         } catch(PDOException $exception) {
16             echo "Connection error: " . $exception->getMessage();
17         }
18
19         return $this->conn;
20     }
21 }
22
```

- controllers/AdminController.php

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

- controllers/GuestController.php

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

```
// Fungsi untuk menghapus tamu
1 reference | 0 overrides
public function delete($id): bool {
    $this->guest->id = $id;
    return $this->guest->delete();
}

// Fungsi untuk mengupdate tamu
1 reference | 0 overrides
public function update($id, $name, $comment): bool {
    $this->guest->id = $id;
    $this->guest->name = $name;
    $this->guest->comment = $comment;
    return $this->guest->update();
}
}
?>
```

- /models/Admin.php

```

Admin.php X
models > Admin.php > ...
1 <?php
  1 reference | 0 implementations
2 class Admin {
  2 references
3 | private $conn;
4 | private $table_name = "admins";
5 |
6 | public $id;
  4 references
7 | public $username;
  2 references
8 | public $password;
9 |
10 | public function __construct($db) {
11 | | $this->conn = $db;
12 | }
13 |
14 | // Fungsi login admin
  1 reference | 0 overrides
15 public function login(): bool {
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(string: strip_tags(string: $this->username));
21 |
22 | // Bind data
23 | $stmt->bindParam(":username", $this->username);
24 | $stmt->execute();
25 |

```

```

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

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

```

- /models/Guest.php

```

Guest.php X
models > Guest.php > ...
1 <?php
  1 reference | 0 implementations
2 class Guest {
  5 references
3 | private $conn;
  4 references
4 | private $table_name = "guests";
5 |
6 | public $id;
  6 references
7 | public $name;
  8 references
8 | public $comment;
  0 references
9 | public $created_at;
10 |
  1 reference | 0 overrides
11 public function __construct($db) {
12 | | $this->conn = $db;
13 | }
14 |
15 | // Menambahkan data tamu baru
  1 reference | 0 overrides
16 public function create(): bool {
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(string: strip_tags(string: $this->name));
22 | $this->comment = htmlspecialchars(string: strip_tags(string: $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 1 reference | 0 overrides
36 public function readAll(): mixed {
37     $query = "SELECT * FROM " . $this->table_name. " ORDER BY created_at DESC";
38     $stmt = $this->conn->prepare($query);
39     $stmt->execute();
40     return $stmt;
41 }
42
43 // Menghapus data tamu
44 1 reference | 0 overrides
45 public function delete(): bool {
46     $query = "DELETE FROM " . $this->table_name. " WHERE id = ?";
47     $stmt = $this->conn->prepare($query);
48     $stmt->bindParam(1, $this->id);
49
50     if($stmt->execute()) {
51         return true;
52     }
53     return false;
54 }

```

```

// Mengupdate data tamu
1 reference | 0 overrides
public function update(): bool {
    $query = "UPDATE " . $this->table_name. " SET name=:name, comment=:comment WHERE id=:id";
    $stmt = $this->conn->prepare($query);

    // Bersihkan data
    $this->name = htmlspecialchars(string: strip_tags(string: $this->name));
    $this->comment = htmlspecialchars(string: strip_tags(string: $this->comment));
    $this->id = htmlspecialchars(string: strip_tags(string: $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;
}
}
?>

```

- /views/admin_dashboard.php

```

admin_dashboard.php X
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(string: $row['name']); ?></td>
14            <td><?php echo htmlspecialchars(string: $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?')">Ha
19            </td>
20        </tr>
21    <?php } ?>
22 </table>

```

- /views/admin_login.php

```
admin_login.php X
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><br>
5   <label for="password">Password:</label><br>
6   <input type="password" name="password" required><br><br>
7   <input type="submit" value="Login">
8 </form>
9
```

- /views/gues_form.php

```
guest_form.php X
views > guest_form.php
1 <h2>Selamat Datang di Buku Tamu</h2>
2 <form action="index.php" method="post">
3   <label for="name">Nama:</label><br>
4   <input type="text" name="name" required><br><br>
5   <label for="comment">Komentar:</label><br>
6   <textarea name="comment" rows="5" required></textarea><br><br>
7   <input type="submit" value="Kirim">
8 </form>
```

- /views/guest_list.php

```
guest_list.php X
views > guest_list.php
1 <h3>Daftar Komentar</h3>
2 <table border="1" cellpadding="5" cellspacing="0">
3   <tr>
4     <th>Nama</th>
5     <th>Komentar</th>
6     <th>Waktu</th>
7   </tr>
8   <?php while ($row = $guests->fetch(PDO::FETCH_ASSOC)) { ?>
9     <tr>
10      <td><?php echo htmlspecialchars(string: $row['name']); ?></td>
11      <td><?php echo htmlspecialchars(string: $row['comment']); ?></td>
12      <td><?php echo $row['created_at']; ?></td>
13    </tr>
14    <?php } ?>
15 </table>
```

- Admin.php

```
Admin.php X
Admin.php
1 <?php
2 session_start();
3 if (!isset($_SESSION['admin'])) {
4     header(header: "Location: login.php");
5     exit();
6 }
7
8 require_once 'controllers/GuestController.php';
9
10 $guestController = new GuestController();
11
12 if (isset($_GET['delete_id'])) {
13     $id = $_GET['delete_id'];
14     $guestController->delete(id: $id);
15     header(header: "Location: Admin.php");
16 }
17
18 $guests = $guestController->readAll();
19
20 include 'views/admin_dashboard.php';
21 ?>
```

- edit.php

```
edit.php X
edit.php
1 <?php
2 session_start();
3 if (!isset($_SESSION['admin'])) {
4     header(header: "Location: login.php");
5     exit();
6 }
7
8 require_once 'controllers/guestcontrollers.php';
9
10 $guestController = new GuestController();
11
12 if (isset($_GET['id'])) {
13     $id = $_GET['id'];
14     $stmt = $guestController->readAll();
15     while ($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: $id, name: $name, comment: $comment);
29     header(header: "Location: admin.php");
30 }
31
```

```
>>
<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(string: $name); ?>" required><br><br>
    <label for="comment">Komentar:</label><br>
    <textarea name="comment" rows="5" required><?php echo htmlspecialchars(string: $comment); ?></textarea><br><br>
    <input type="submit" value="Update">
</form>
```

- index.php


```

index.php
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: $name, comment: $comment);
10     header(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

```

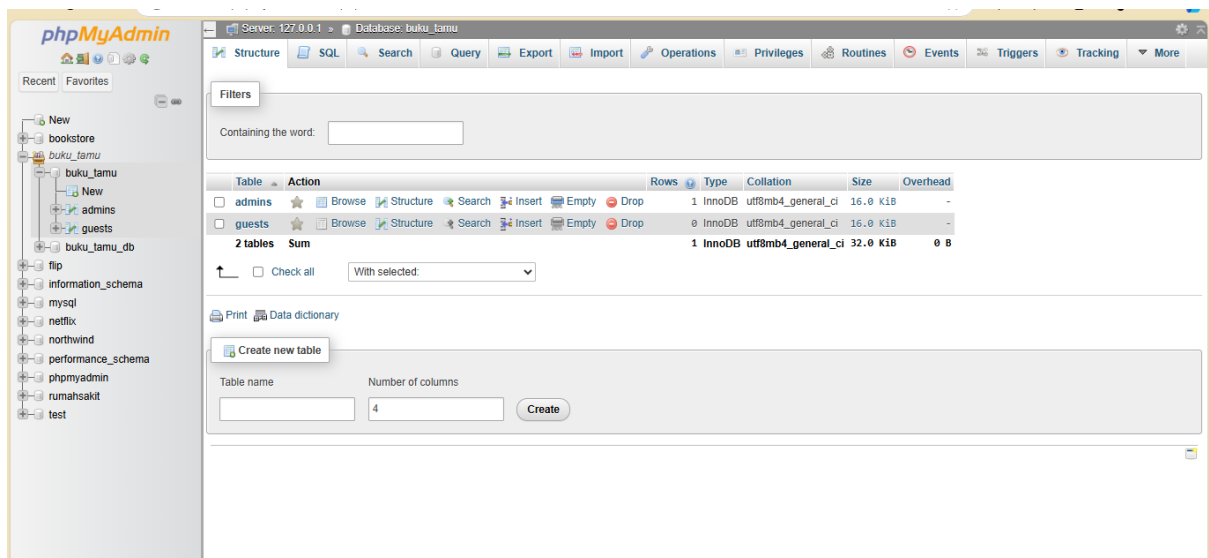
login.php
1  <?php
2  session_start();
3  require_once 'controllers/AdminController.php';
4
5  $adminController = new AdminController();
6
7  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
8      $username = $_POST['username'];
9      $password = $_POST['password'];
10
11     if ($adminController->login(username: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

```

- logout.php

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

- database di php



- hasil running

Selamat Datang di Buku Tamu

Nama:

Komentar:

Kirim

Daftar Komentar

Nama	Komentar	Waktu
Irma kunyuk	hoppeless	2024-09-27 11:25:34
Nadiyah Qasamah	IIWD METTY	2024-09-27 10:42:42
erinna	samawa	2024-09-27 10:26:13