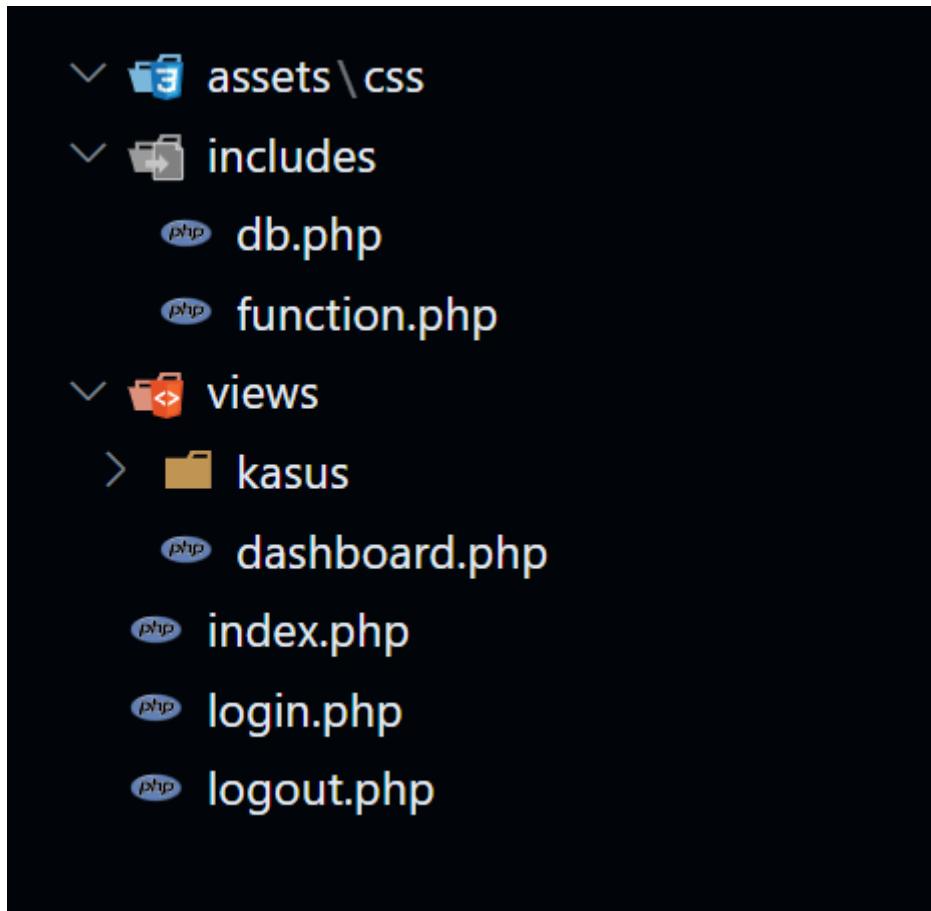


Implementasi Fitur Lanjutan

Pada tahap ini, kita akan melanjutkan pengembangan aplikasi dari **Tahap I**, menambahkan fitur login dan pengaturan hak akses.

Struktur Folder



Sistem ini terdiri dari tiga peran utama, yaitu **Siswa**, **Guru**, dan **Admin**, dengan masing-masing peran memiliki tugas dan akses yang berbeda:

- **Siswa:**
Siswa hanya memiliki akses untuk login dan melihat dashboard mereka. Dashboard ini menampilkan daftar kasus yang terkait dengan siswa tersebut, tanpa hak untuk mengubah atau menambah data.
- **Guru:**
Guru memiliki akses untuk mengelola data siswa dan mencatat kasus siswa. Guru dapat menambah data siswa baru serta mencatat kasus yang berkaitan dengan siswa tertentu, mendukung proses pelaporan dan dokumentasi yang efektif.
- **Admin:**
Admin memiliki hak akses penuh untuk mengelola semua data dalam sistem. Admin

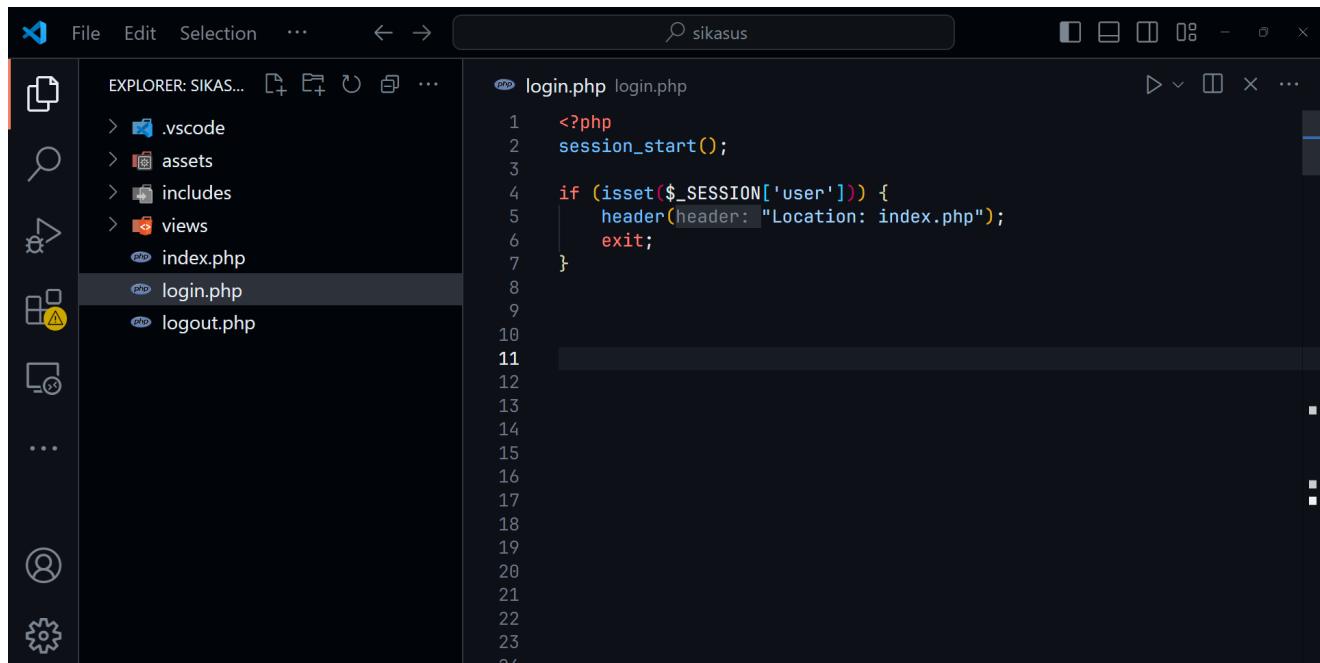
dapat membuat dan mengelola data kelas, guru, siswa, serta kasus siswa, memastikan integritas dan pengaturan sistem berjalan dengan baik.

Membuat Halaman Login

Pertama-tama, buat file bernama `login.php` di root proyek Anda jika belum ada. File ini akan menjadi halaman login utama untuk siswa, wali kelas, dan admin.

Selanjutnya, tambahkan sesi pengguna di bagian paling atas file untuk mengelola login. Gunakan kode berikut:

```
session_start();
if (isset($_SESSION['user'])) {
    header("Location: index.php");
    exit;
}
```

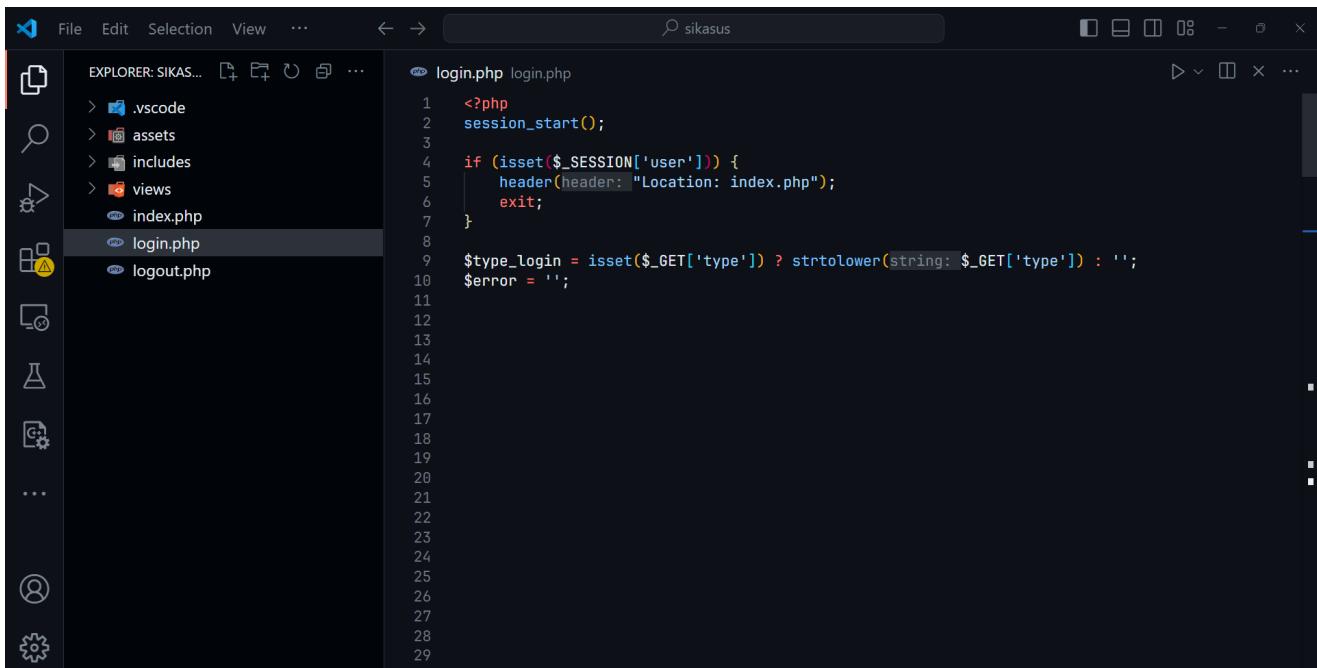


Penjelasan:

- Fungsi `session_start()` akan memulai sesi PHP.
- `isset($_SESSION['user'])` akan mengecek apakah pengguna sudah login. Jika iya, pengguna akan diarahkan ke halaman utama (`index.php`).

Kemudian, tentukan jenis login berdasarkan parameter `type` di URL. Tambahkan kode berikut:

```
$type_login = isset($_GET['type']) ? strtolower($_GET['type']) : '';
$error = '';
```



```
<?php
session_start();

if (isset($_SESSION['user'])) {
    header(header: "Location: index.php");
    exit;
}

$type_login = isset($_GET['type']) ? strtolower(string: $_GET['type']) : '';
$error = '';
```

Penjelasan:

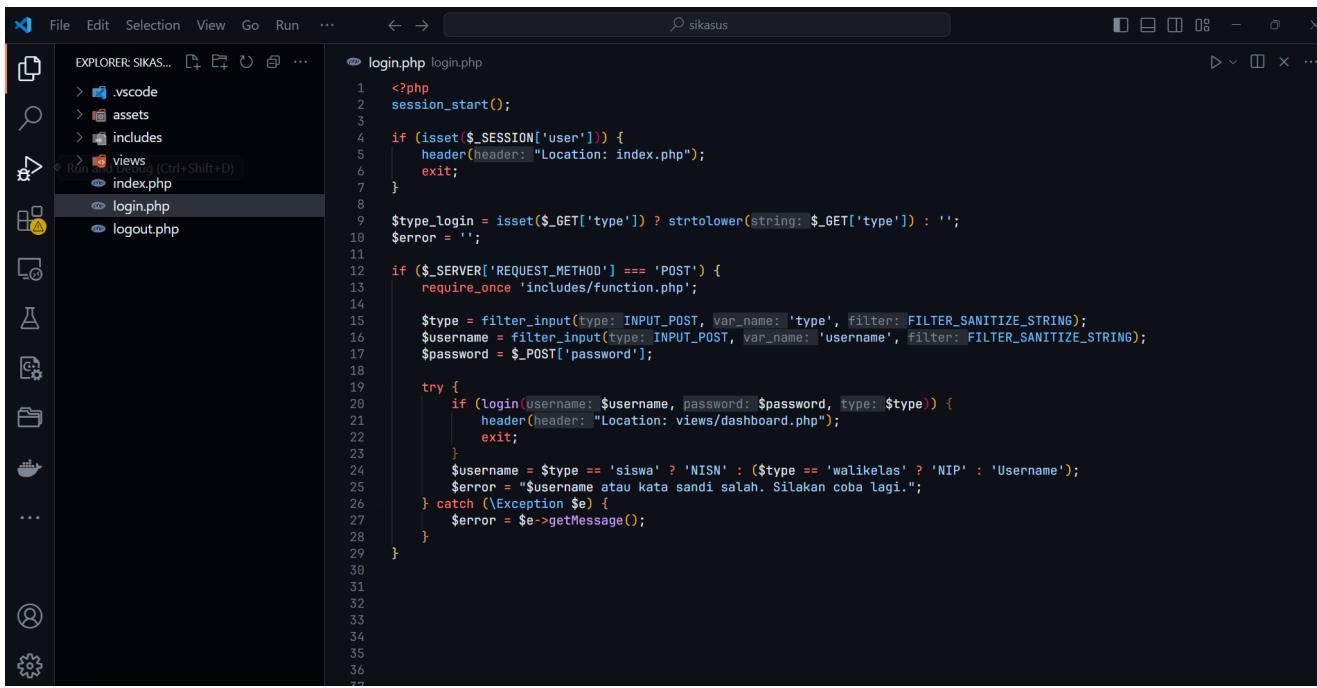
- Variabel `$type_login` menangkap jenis login (`siswa`, `walikelas`, atau `admin`).
- Variabel `$error` akan menyimpan pesan kesalahan jika login gagal.

Berikutnya, tambahkan logika untuk menangani form login jika metode request adalah POST. Gunakan kode:

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    require_once 'includes/function.php';

    $type = htmlspecialchars($_POST['type']);
    $username = htmlspecialchars($_POST['username']);
    $password = $_POST['password'];

    try {
        if (login($username, $password, $type)) {
            header("Location: views/dashboard.php");
            exit;
        }
        $username = $type == 'siswa' ? 'NISN' : ($type == 'walikelas' ?
        'NIP' : 'Username');
        $error = "$username atau kata sandi salah. Silakan coba lagi.";
    } catch (\Exception $e) {
        $error = $e->getMessage();
    }
}
```



```
<?php
session_start();

if (isset($_SESSION['user'])) {
    header(header: "Location: index.php");
    exit;
}

$type_login = isset($_GET['type']) ? strtolower(string: $_GET['type']) : '';
$error = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    require_once 'includes/function.php';

    $type = filter_input(type: INPUT_POST, var_name: 'type', filter: FILTER_SANITIZE_STRING);
    $username = filter_input(type: INPUT_POST, var_name: 'username', filter: FILTER_SANITIZE_STRING);
    $password = $_POST['password'];

    try {
        if (login(username: $username, password: $password, type: $type)) {
            header(header: "Location: views/dashboard.php");
            exit;
        }
        $username = $type == 'siswa' ? 'NISN' : ($type == 'walikelas' ? 'NIP' : 'Username');
        $error = "$username atau kata sandi salah. Silakan coba lagi.";
    } catch (\Exception $e) {
        $error = $e->getMessage();
    }
}
}

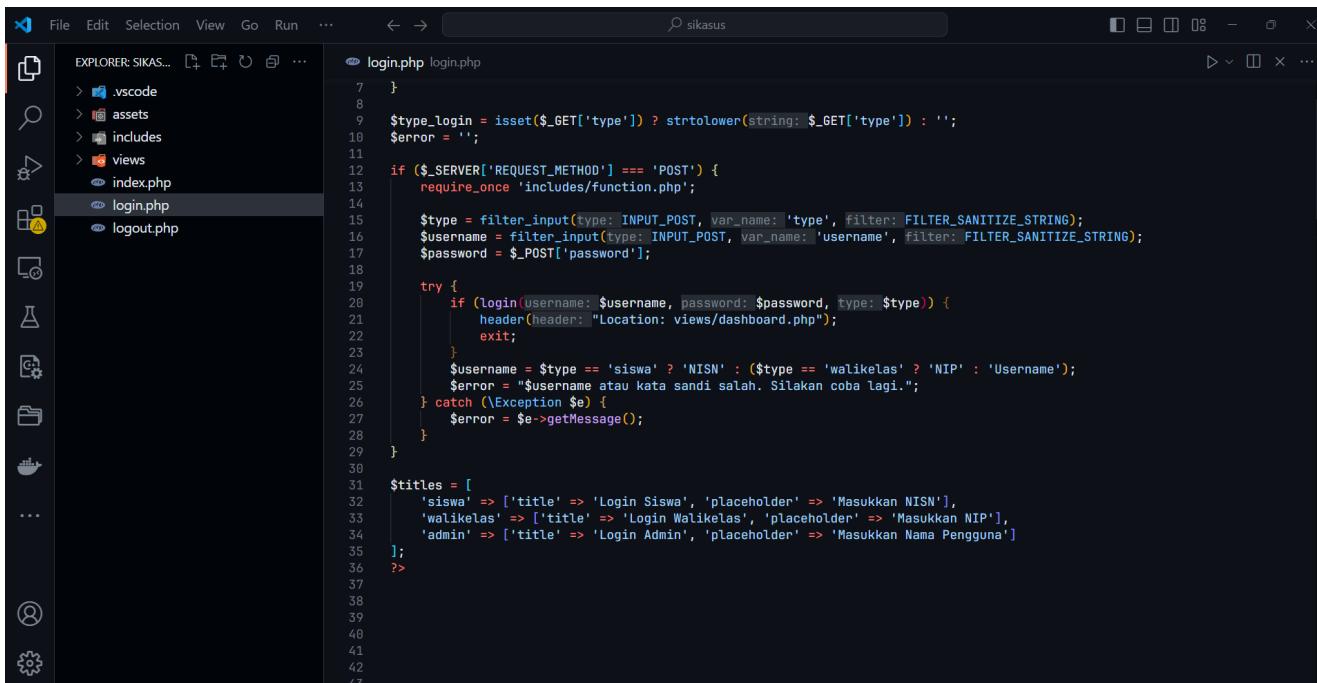
30
31
32
33
34
35
36
37
```

Penjelasan:

- Variabel `$type` menentukan jenis pengguna (misalnya siswa atau walikelas) untuk memastikan login sesuai dengan peran pengguna
- Kata sandi diambil langsung dari input tanpa filter khusus karena akan diverifikasi oleh fungsi `login()`.
- Fungsi `login($username, $password, $type)` memvalidasi kredensial berdasarkan tipe pengguna (siswa, walikelas, atau admin).
- Blok `try-catch` digunakan untuk menangkap kesalahan tak terduga selama proses login, misalnya jika koneksi database gagal.

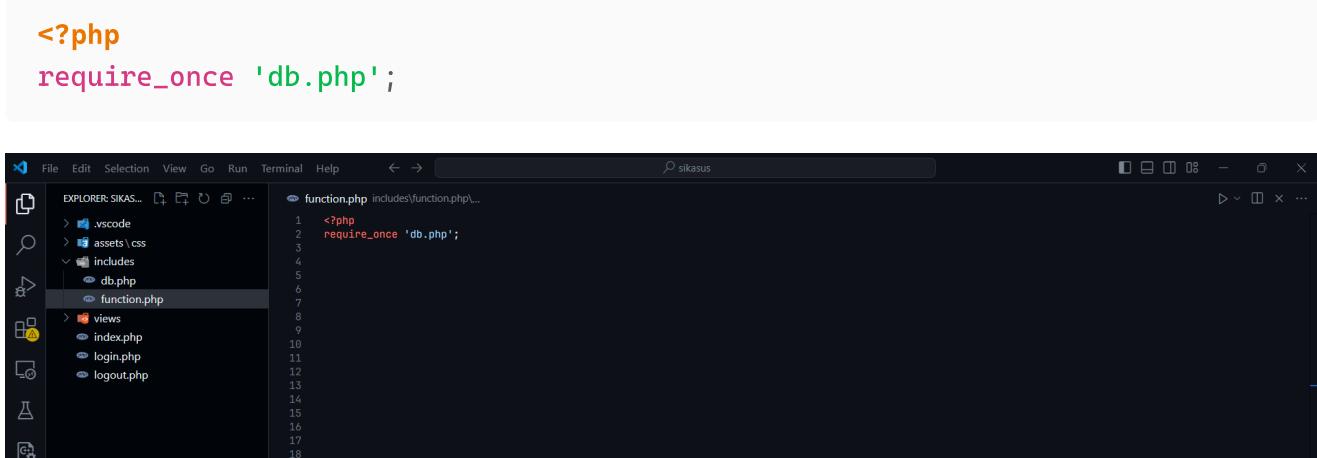
Selanjutnya, buat array `$titles` untuk menentukan tampilan form login berdasarkan tipe pengguna:

```
$titles = [
    'siswa' => ['title' => 'Login Siswa', 'placeholder' => 'Masukkan NISN'],
    'walikelas' => ['title' => 'Login Walikelas', 'placeholder' => 'Masukkan NIP'],
    'admin' => ['title' => 'Login Admin', 'placeholder' => 'Masukkan Nama Pengguna']
];
```



```
File Edit Selection View Go Run ... ⏪ ⏩ ⏴ ⏵ ⏹ ⏸ ⏷ ⏹ sikasus
EXPLORER: SIKAS... ⏹ ⏷ ⏸ ⏹ ⏵ ...
login.php login.php
7 }
8
9 $type_login = isset($_GET['type']) ? strtolower(string: $_GET['type']) : '';
10 $error = '';
11
12 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
13     require_once 'includes/function.php';
14
15     $type = filter_input(type: INPUT_POST, var_name: 'type', filter: FILTER_SANITIZE_STRING);
16     $username = filter_input(type: INPUT_POST, var_name: 'username', filter: FILTER_SANITIZE_STRING);
17     $password = $_POST['password'];
18
19     try {
20         if (login(username: $username, password: $password, type: $type)) {
21             header(header: "Location: views/dashboard.php");
22             exit;
23         }
24         $username = $type == 'siswa' ? 'NISN' : ($type == 'walikelas' ? 'NIP' : 'Username');
25         $error = "$username atau kata sandi salah. Silakan coba lagi.";
26     } catch (\Exception $e) {
27         $error = $e->getMessage();
28     }
29
30     $titles = [
31         'siswa' => ['title' => 'Login Siswa', 'placeholder' => 'Masukkan NISN'],
32         'walikelas' => ['title' => 'Login Walikelas', 'placeholder' => 'Masukkan NIP'],
33         'admin' => ['title' => 'Login Admin', 'placeholder' => 'Masukkan Nama Pengguna']
34     ];
35     ?>
36
37
38
39
40
41
42
43
```

Setelah itu, buat file `function.php` di folder `includes/` jika belum ada. File ini akan berfungsi untuk menangani logika login dan validasi pengguna.



```
<?php
require_once 'db.php';
```

Penjelasan :

- `require_once 'db.php';` : Baris ini digunakan untuk memuat file `db.php` yang berisi koneksi ke database. File ini diperlukan untuk melakukan query ke database dan memverifikasi kredensial login.

Buat fungsi `login` dengan 3 parameter yang diperlukan.

```
function login($username, $password, $type): bool
{
    global $conn;
```

The screenshot shows the VS Code interface. The left sidebar displays the file structure under 'EXPLORER: SIKAS...'. The 'function.php' file is selected in the 'includes' folder. The right pane shows the code content:

```
<?php
require_once 'db.php';

function login($username, $password, $type): bool
{
    global $conn;

    if ($type === "admin") {
        $stmt = $conn->prepare("SELECT * FROM tb_users WHERE username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $result = $stmt->get_result();
        $user = $result->fetch_assoc();
    }
}
```

Penjelasan :

- Fungsi `login()` menerima tiga parameter: `$username`, `$password`, dan `$type` untuk mengidentifikasi kredensial pengguna dan jenis pengguna yang login.
- Variabel `$conn` adalah koneksi database yang diimpor dari `db.php`.

Selanjutnya, tambahkan logika untuk menangani login tipe `admin`.

```
if ($type === "admin") {
    $stmt = $conn->prepare("SELECT * FROM tb_users WHERE username = ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();
    $user = $result->fetch_assoc();
}
```

The screenshot shows the VS Code interface with the same file structure and code as the previous screenshot, but the code in the right pane has been modified to include the logic for handling admin login:

```
<?php
require_once 'db.php';

function login($username, $password, $type): bool
{
    global $conn;

    if ($type === "admin") {
        $stmt = $conn->prepare("SELECT * FROM tb_users WHERE username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $result = $stmt->get_result();
        $user = $result->fetch_assoc();
    }
}
```

Query yang digunakan

```
SELECT * FROM tb_users WHERE username = ?
```

Penjelasan : Query ini digunakan untuk mencari data pengguna dengan `username` tertentu dalam tabel `tb_users`. Jika tipe pengguna adalah "admin", query mempersiapkan perintah SQL untuk memilih semua data dari `tb_users` di mana `username` sesuai dengan parameter yang diberikan. Parameter `username` diikat menggunakan `bind_param` dengan tipe data string (`s`), kemudian query dijalankan dengan `execute`. Hasilnya diambil dan disimpan dalam variabel `$user` sebagai array asosiasi.

Penjelasan:

- Kondisi `if ($type === "admin")` memastikan hanya pengguna dengan tipe `admin` yang menjalani blok kode ini.
- Query SQL mencari pengguna berdasarkan `username`.
- `prepare()` dan `bind_param()` digunakan untuk menghindari SQL injection.
- `get_result()` mengambil hasil query, dan `fetch_assoc()` mengonversi hasil menjadi array asosiatif.

Periksa apakah password yang dimasukkan cocok dengan yang ada di database untuk tipe `admin`.

```
if ($user && password_verify($password, $user['password'])) {  
    $_SESSION['user'] = $user;  
    $_SESSION['user']['role'] = 'admin';  
    return true;
```

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a tree view of files and folders. The current file being edited is `function.php`, which is located in the `includes` folder. The code in `function.php` contains a `login` function that uses prepared statements to check if a user exists in the database based on their `username` and `password`. It also includes a `password_verify` call to check if the provided password matches the hashed password in the database.

```
<?php  
require_once 'db.php';  
  
function login($username, $password, $type): bool  
{  
    global $conn;  
  
    if ($type === "admin") [  
        $stmt = $conn->prepare(query: "SELECT * FROM tb_users WHERE username = ?");  
        $stmt->bind_param(types: "s", var: &$username);  
        $stmt->execute();  
        $result = $stmt->get_result();  
        $user = $result->fetch_assoc();  
    ]  
}
```

Penjelasan:

- Fungsi `password_verify()` memeriksa apakah password yang dimasukkan cocok dengan password yang terenkripsi di database.

Kemudian, tambahkan logika untuk menangani login tipe `siswa`.

```

if ($type === "siswa") {
    $stmt = $conn->prepare("SELECT * FROM tb_siswa WHERE nisn = ? AND nisn =
?");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $result = $stmt->get_result();
}

```

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a tree view of the project structure. The current file, 'function.php', is selected and shown in the main editor area. The code implements a login function that checks against two database tables ('tb_users' for admin and 'tb_siswa' for siswa) and stores the user information in a session variable.

```

<?php
require_once 'db.php';

function login($username, $password, $type): bool
{
    global $conn;

    if ($type === "admin") {
        $stmt = $conn->prepare(query: "SELECT * FROM tb_users WHERE username = ?");
        $stmt->bind_param(types: "s", var: &$username);
        $stmt->execute();
        $result = $stmt->get_result();
        $user = $result->fetch_assoc();
    }

    if ($user && password_verify(password: $password, hash: $user['password'])) {
        $_SESSION['user'] = $user;
        $_SESSION['user']['role'] = 'admin';
        return true;
    }
}

if ($type === "siswa") {
    $stmt = $conn->prepare(query: "SELECT * FROM tb_siswa WHERE nisn = ? AND nisn = ?");
    $stmt->bind_param(types: "ss", var: &$username, vars: &$password);
    $stmt->execute();
    $result = $stmt->get_result();
}

```

Periksa apakah ada hasil yang ditemukan untuk tipe `siswa` dan simpan informasi pengguna dalam session.

```

if ($result->num_rows > 0) {
    $user = $result->fetch_assoc();
    $_SESSION['user'] = $user;
    $_SESSION['user']['role'] = 'siswa';
    return true;
}

```

```
function login($username, $password, $type): bool
{
    $stmt = $conn->prepare("SELECT * FROM tb_siswa WHERE nisn = ? AND nisn = ?");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $user = $result->fetch_assoc();
        $_SESSION['user'] = $user;
        $_SESSION['user']['role'] = 'siswa';
        return true;
    }
}

if ($type === "siswa") {
    $stmt = $conn->prepare("SELECT * FROM tb_siswa WHERE nisn = ? AND nisn = ?");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $user = $result->fetch_assoc();
        $_SESSION['user'] = $user;
        $_SESSION['user']['role'] = 'siswa';
        return true;
    }
}
```

Query yang digunakan

```
SELECT * FROM tb_siswa WHERE nisn = ? AND nisn = ?
```

Penjelasan : Query ini digunakan untuk mencari data siswa di tabel `tb_siswa` berdasarkan kecocokan dua nilai `nisn`, yang masing-masing diwakili oleh `username` dan `password`. Jika tipe pengguna "siswa", query dijalankan dengan parameter yang diikat dan hasilnya diambil dengan `get_result`.

Penjelasan:

- Kondisi `if ($type === "siswa")` memastikan blok kode ini hanya dijalankan untuk pengguna bertipe siswa .
- Query SQL mencari siswa berdasarkan `nisn` dan `password` .
- `num_rows` memeriksa apakah ada baris hasil query yang ditemukan.

Selanjutnya, tambahkan logika untuk menangani login tipe `walikelas` .

```
if ($type === "walikelas") {
    $stmt = $conn->prepare("SELECT * FROM tb_walikelas WHERE nip = ? AND nip = ?");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $result = $stmt->get_result();
}
```

```
function login($username, $password, $type): bool
{
    $user = $result->fetch_assoc();
    $_SESSION['user'] = $user;
    $_SESSION['user']['role'] = 'siswa';
    return true;
}

if ($type === "walikelas") {
    $stmt = $conn->prepare("SELECT * FROM tb_walikelas WHERE nip = ? AND nip = ?");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $result = $stmt->get_result();
}
```

Periksa apakah ada hasil untuk tipe walikelas dan simpan informasi pengguna dalam session.

```
if ($result->num_rows > 0) {
    $user = $result->fetch_assoc();
    $_SESSION['user'] = $user;
    $_SESSION['user']['role'] = 'walikelas';
    return true;
}
```

```
function login($username, $password, $type): bool
{
    $user = $result->fetch_assoc();
    $_SESSION['user'] = $user;
    $_SESSION['user']['role'] = 'siswa';
    return true;
}

if ($type === "walikelas") {
    $stmt = $conn->prepare("SELECT * FROM tb_walikelas WHERE nip = ? AND nip = ?");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $user = $result->fetch_assoc();
        $_SESSION['user'] = $user;
        $_SESSION['user']['role'] = 'walikelas';
        return true;
    }
}
```

Query yang digunakan

```
SELECT * FROM tb_walikelas WHERE nip = ? AND nip = ?
```

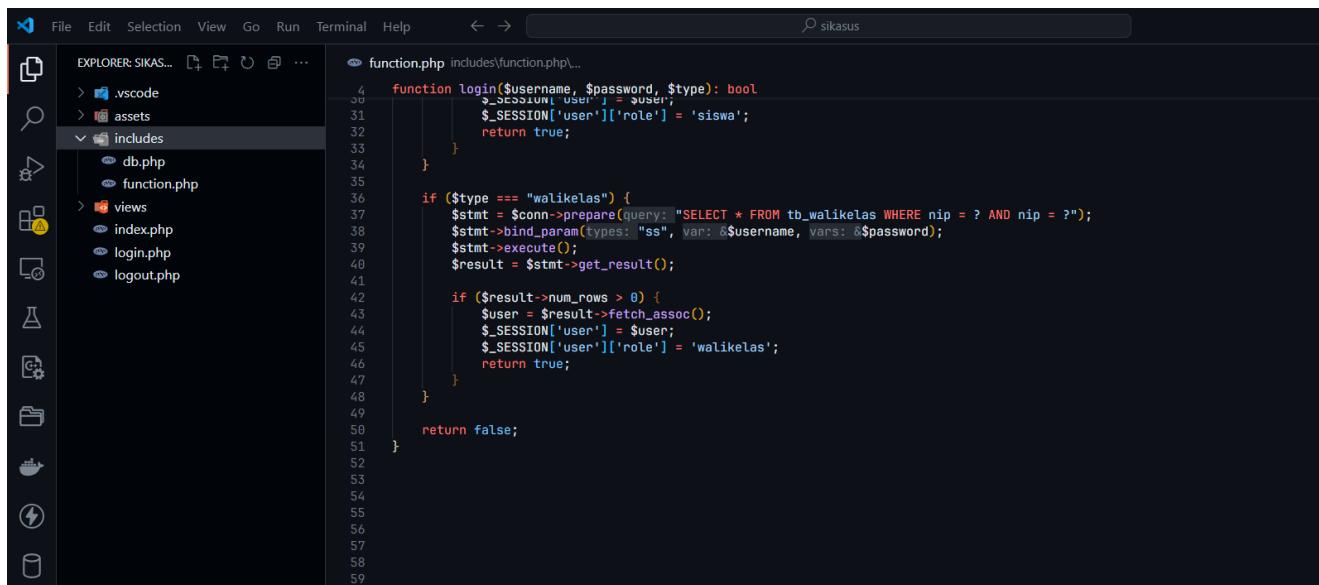
Penjelasan : Query ini digunakan untuk mencari data wali kelas di tabel tb_walikelas berdasarkan kecocokan dua nilai nip , yang masing-masing diwakili oleh username dan password . Jika tipe pengguna adalah "walikelas", query dijalankan dengan parameter yang diikat dan hasilnya diambil dengan get_result . Namun, query ini memiliki redundansi karena kedua blok kode identik.

Penjelasan:

- Kondisi `if ($type === "walikelas")` memastikan hanya pengguna dengan tipe `walikelas` yang menjalani blok kode ini.
- Query SQL mencari wali kelas berdasarkan `nip` dan `password`.
- `num_rows` memeriksa apakah ada baris yang ditemukan.

Selanjutnya, kembalikan `false` jika tidak ada kecocokan.

```
return false;
```



```
function login($username, $password, $type): bool {
    $_SESSION['user'] = $user;
    $_SESSION['user']['role'] = 'siswa';
    return true;
}

if ($type === "walikelas") {
    $stmt = $conn->prepare("SELECT * FROM tb_walikelas WHERE nip = ? AND nip = ?");
    $stmt->bind_param(types: "ss", var: &$username, var: &$password);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $user = $result->fetch_assoc();
        $_SESSION['user'] = $user;
        $_SESSION['user']['role'] = 'walikelas';
        return true;
    }
}

return false;
}
```

Penjelasan:

- Jika tidak ada pengguna yang cocok dengan kredensial yang dimasukkan, fungsi mengembalikan `false` untuk menunjukkan bahwa login gagal.

Kemudian, kembali lagi ke halaman `login.php` setelah itu, tambahkan HTML untuk menampilkan form login. Pastikan form ditampilkan sesuai dengan tipe login. Contoh kode:

```
<div class="container">
    <?php if (isset($titles[$type_login])): ?>
        <h2><?= $titles[$type_login]['title'] ?></h2>
        <form method="POST" autocomplete="on">
            <input type="hidden" name="type" value="<?= $type_login ?>">
            <div class="input-group">
                <input type="text" name="username" autofocus placeholder="<?
                    = $titles[$type_login]['placeholder'] ?>" required>
            </div>
            <div class="input-group">
                <input type="password" name="password" placeholder="Password" required>
            </div>
            <button type="submit">Login</button>
        </form>
    <?php endif; ?>
</div>
```

```

<?php if ($error): ?>
    <p class="error-message"><?= $error ?></p>
<?php endif; ?>
<a href="login.php">Kembali</a>
</form>
<?php else: ?>
    <h2>Silakan Pilih Jenis Login</h2>
    <div class="btn-group">
        <a href="?type=siswa">Siswa</a>
        <a href="?type=walikelas">Walikelas</a>
        <a href="?type=admin">Admin</a>
    </div>
<?php endif; ?>
</div>

```

The screenshot shows the VS Code interface with the 'login.php' file open. The file content is identical to the one above, showing PHP code for a login page with user type selection and error handling.

Selanjutnya, kita akan mempercantik tampilan login menggunakan CSS dengan metode *internal style*. Tambahkan elemen `<style>` di bagian `<head>` dalam file HTML Anda.

```

<style>
*, html {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

/* Gaya body untuk tata letak tengah */
body {
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
}

```

```
height: 100vh;
background-color: #f9f9f9;
}

/* Kontainer login */
.container {
    max-width: 400px;
    width: 100%;
    padding: 20px;
    background-color: white;
    border: 1px solid #ccc;
    border-radius: 8px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    text-align: center;
}

/* Grup input */
.input-group {
    margin-bottom: 15px;
}

input,
button,
a {
    width: 100%;
    padding: 10px;
    margin: 5px 0;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 4px;
}

input:focus {
    border-color: #007bff;
    outline: none;
}

button {
    background-color: #007bff;
    color: white;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: #0056b3;
}

.error-message {
    color: red;
}
```

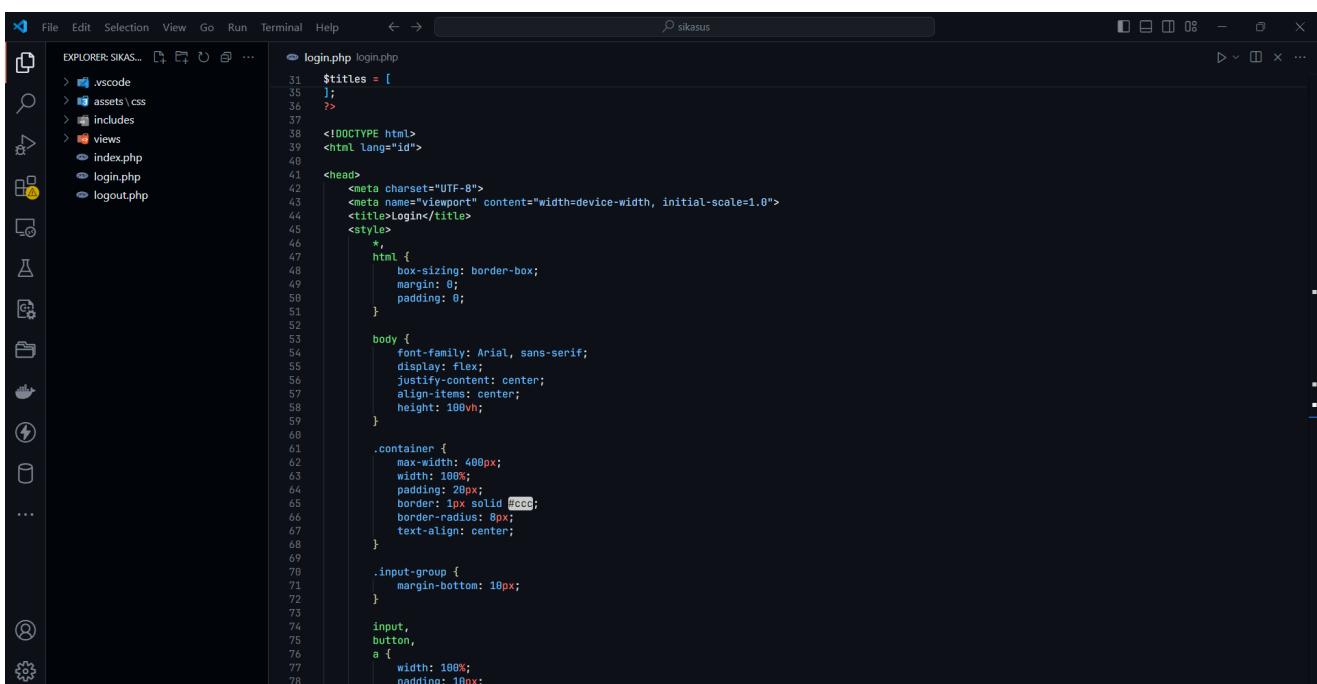
```
    font-size: 14px;
    margin-top: -10px;
    margin-bottom: 10px;
}

/* Grup tombol atau tautan */
.btn-group a {
    display: inline-block;
    margin: 5px;
    padding: 10px 20px;
    border: 1px solid #ccc;
    text-decoration: none;
    border-radius: 4px;
    color: #007bff;
}

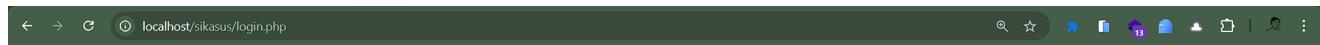
.btn-group a:hover {
    background-color: #f0f0f0;
}

h2 {
    margin-bottom: 1rem;
    color: #333;
}

</style>
```



Sebelum :



Silakan Pilih Jenis Login

[Siswa](#) [Walikelas](#) [Admin](#)

Sesudah :

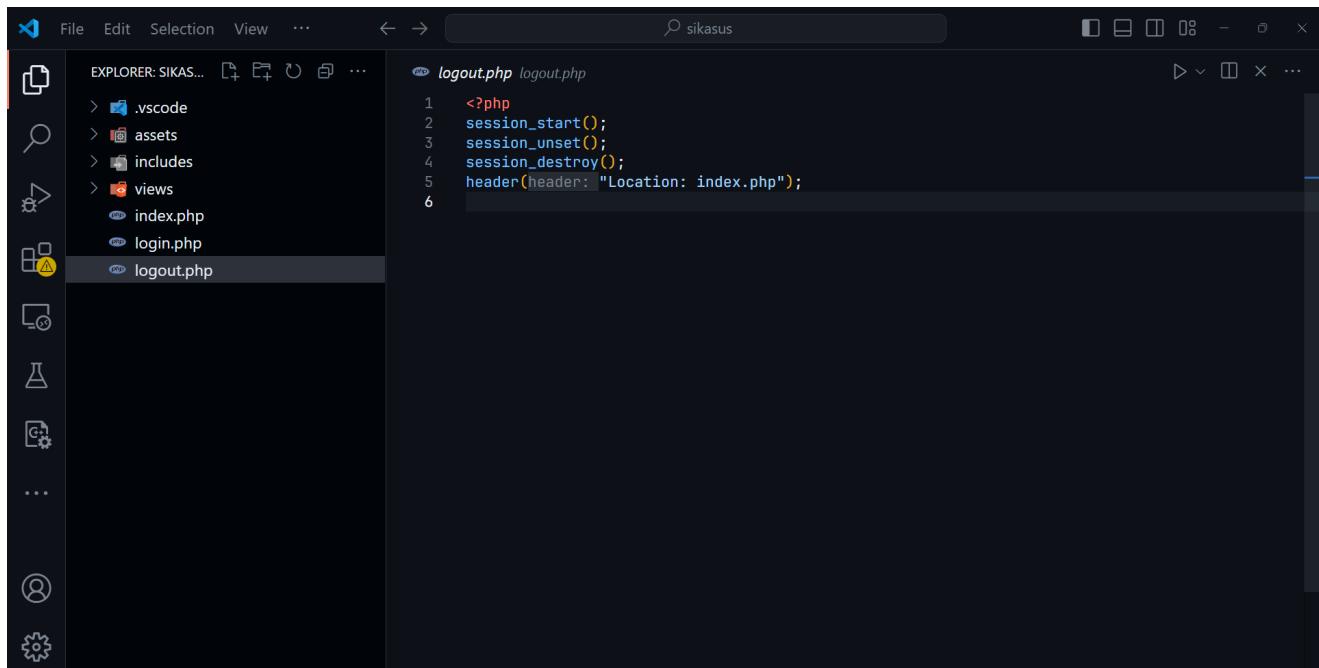


Silakan Pilih Jenis Login

Siswa
Walikelas
Admin

Kemudian, buat file `logout.php` untuk menghentikan sesi pengguna. Gunakan kode berikut:

```
session_start();
session_unset();
session_destroy();
header("Location: index.php");
```



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

Kemudian, buat file `dashboard.php` dalam folder `views` untuk menampilkan tampilan setelah berhasil login

```
<?php
session_start();

// Cek apakah user sudah login
if (!isset($_SESSION['user'])) {
    header("Location: login.php");
    exit;
}
```

Penjelasan :

- `session_start()` digunakan untuk memulai sesi PHP, yang memungkinkan kita mengakses data sesi, seperti informasi login.
- `if (!isset($_SESSION['user']))` adalah kondisi untuk memeriksa apakah sesi user tidak tersedia. Jika iya, pengguna akan diarahkan ke halaman `login.php` dengan fungsi `header()`, lalu skrip dihentikan menggunakan `exit`.

Selanjutnya, kita perlu menyertakan file `db.php` untuk terhubung ke database. File ini berisi konfigurasi koneksi yang akan digunakan dalam aplikasi.

```
require_once __DIR__ . '/../includes/db.php';
```

Penjelasan

- `require_once` memastikan file hanya dimuat sekali dalam satu eksekusi skrip.

- `__DIR__` mengacu pada direktori tempat file PHP saat ini berada. Dengan menambahkan `.. /`, kita berpindah ke folder sebelumnya untuk menemukan file `db.php`.
-

Lalu, mengambil data pengguna yang telah login dan menyimpan informasi penting seperti **role** mereka ke dalam variabel.

```
// Ambil data user dari session
$user = $_SESSION['user'];
$role = $user['role']; // role: 'admin', 'walikelas', atau 'siswa'
```

Penjelasan

- `$_SESSION['user']` adalah data pengguna yang disimpan saat login. Kita menyimpannya ke variabel `$user`.
 - `$role = $user['role']` mengambil informasi **role** pengguna, seperti apakah mereka seorang **admin**, **walikelas**, atau **siswa**.
-

Kemudian, mengarahkan pengguna berdasarkan peran. Jika pengguna memiliki peran **siswa**, mereka tidak diizinkan mengakses halaman walikelas dan akan diarahkan ke dashboard siswa.

```
if ($role === "siswa") {
    header("Location: siswa/dashboard.php");
    exit;
}
```

Penjelasan :

- `if ($role === "siswa")` memeriksa apakah peran pengguna adalah **siswa**.
 - Jika iya, pengguna diarahkan ke halaman **siswa/dashboard.php** menggunakan fungsi `header()`.
-

Setelah validasi dan logika selesai, kita mulai membuat struktur halaman HTML. Hal ini mencakup elemen-elemen dasar seperti `<head>` untuk metadata dan `<body>` untuk konten utama.

```
<!DOCTYPE html>
<html lang="id">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Admin Panel</title>
    <link rel="stylesheet" href="/sikasus/assets/css/styles.css" />
  </head>

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

Penjelasan :

- `<meta charset="UTF-8" />` menentukan encoding karakter sebagai UTF-8.
 - `<meta name="viewport" content="width=device-width, initial-scale=1.0" />` memastikan halaman terlihat responsif di perangkat mobile.
 - `<link rel="stylesheet" href="/sikasus/assets/css/styles.css" />` menyertakan file CSS untuk tampilan halaman.
-

Sidebar digunakan untuk navigasi. Sidebar ini akan dimuat dari file terpisah menggunakan fungsi **include** agar mudah dikelola.

```
<!-- Sidebar -->
<?php include($_SERVER['DOCUMENT_ROOT'] .
  '/sikasus/views/component/sidebar.php'); ?>
```

Penjelasan :

- `include()` memuat file **sidebar.php** ke dalam halaman.
 - `$_SERVER['DOCUMENT_ROOT']` mengacu pada direktori root server, memastikan jalur file **sidebar.php** benar.
-

Terakhir, membuat kode untuk sambutan pengguna.

```
<!-- Main Content -->
<div class="main-content">
  <header>
    <h1>Dashboard</h1>
  </header>
  <section>
```

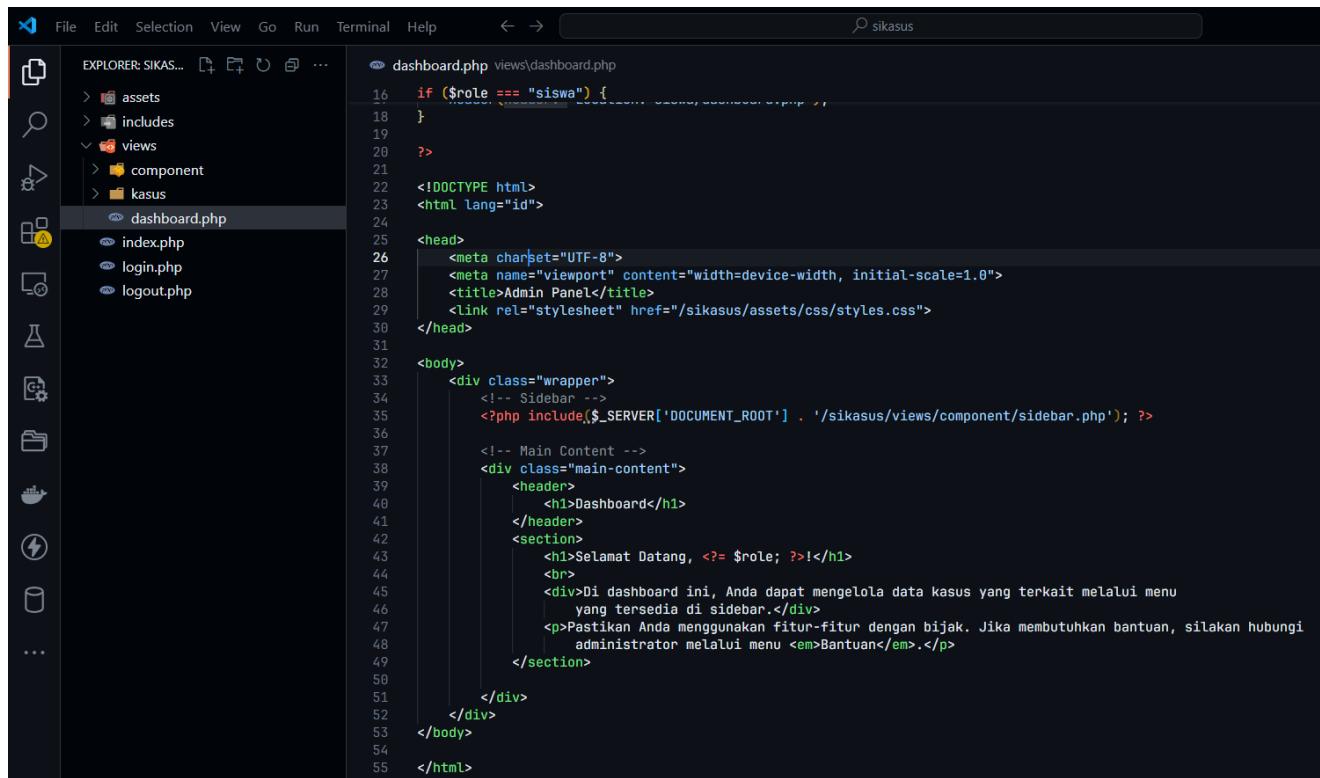
```

<h1>
    Selamat Datang,
    <?= $role; ?>
</h1>
<br />
<div>
    Di dashboard ini, Anda dapat mengelola data kasus yang terkait melalui
    menu yang tersedia di sidebar.
</div>
<p>
    Pastikan Anda menggunakan fitur-fitur dengan bijak. Jika membutuhkan
    bantuan, silakan hubungi administrator melalui menu <em>Bantuan</em>.
</p>
</section>
</div>

```

Penjelasan :

- <?= \$role; ?> menampilkan peran pengguna di dashboard, misalnya **admin** atau **walikelas**.
- Elemen <section> digunakan untuk mendefinisikan blok konten utama, seperti deskripsi fitur yang tersedia.



```

File Edit Selection View Go Run Terminal Help ⏎ ⏎ sikasus
EXPLORER: SIKAS...
> assets
> includes
views
> component
> kasus
dashboard.php
index.php
login.php
logout.php

dashboard.php
16 if ($role === "siswa") {
17     // Siswa, buka sidebar
18 }
19 ?>
20 <!DOCTYPE html>
21 <html lang="id">
22     <head>
23         <meta charset="UTF-8">
24         <meta name="viewport" content="width=device-width, initial-scale=1.0">
25         <title>Admin Panel</title>
26         <link rel="stylesheet" href="/sikasus/assets/css/styles.css">
27     </head>
28     <body>
29         <div class="wrapper">
30             <!-- Sidebar -->
31             <?php include($_SERVER['DOCUMENT_ROOT'] . '/sikasus/views/component/sidebar.php'); ?>
32
33             <!-- Main Content -->
34             <div class="main-content">
35                 <header>
36                     <h1>Dashboard</h1>
37                 </header>
38                 <section>
39                     <h1>Selamat Datang, <?= $role; ?>!</h1>
40                     <br>
41                     <div>Di dashboard ini, Anda dapat mengelola data kasus yang terkait melalui menu
42                         yang tersedia di sidebar.</div>
43                     <p>Pastikan Anda menggunakan fitur-fitur dengan bijak. Jika membutuhkan bantuan, silakan hubungi
44                         administrator melalui menu <em>Bantuan</em>.</p>
45                 </section>
46             </div>
47         </div>
48     </body>
49 </html>

```

Mengecek Hasil Halaman Login

Setelah implementasi halaman login, berikut adalah langkah-langkah untuk mengecek apakah fitur sudah berfungsi dengan benar:

1. Pertama-tama, buka browser dan akses halaman `login.php`.

- URL: `http://localhost/sikasus/login.php`.



2. Selanjutnya, coba klik salah satu tipe login, misalnya **Siswa**.

- Perhatikan bahwa judul halaman berubah menjadi "Login Siswa".
- Placeholder input harus sesuai dengan tipe login (contoh: "Masukkan NISN").



3. Kemudian, masukkan kredensial yang salah, misalnya:

- **NISN**: 12345678
- **Password**: salahpassword
- Klik tombol **Login**.
- **Hasil yang diharapkan**: Pesan kesalahan ditampilkan: "*NISN atau kata sandi salah. Silakan coba lagi.*"

Login Siswa

12345678

.....

Login

NISN atau kata sandi salah. Silakan coba lagi.
[Kembali](#)

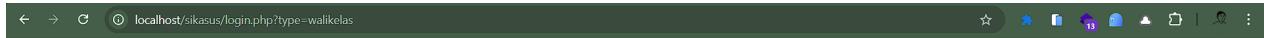
4. **Selanjutnya**, masukkan kredensial yang benar sesuai data di database. - Jika login berhasil, Anda akan diarahkan ke `views/dashboard.php`. - **Hasil yang diharapkan:** Dashboard tampil sesuai hak akses pengguna.



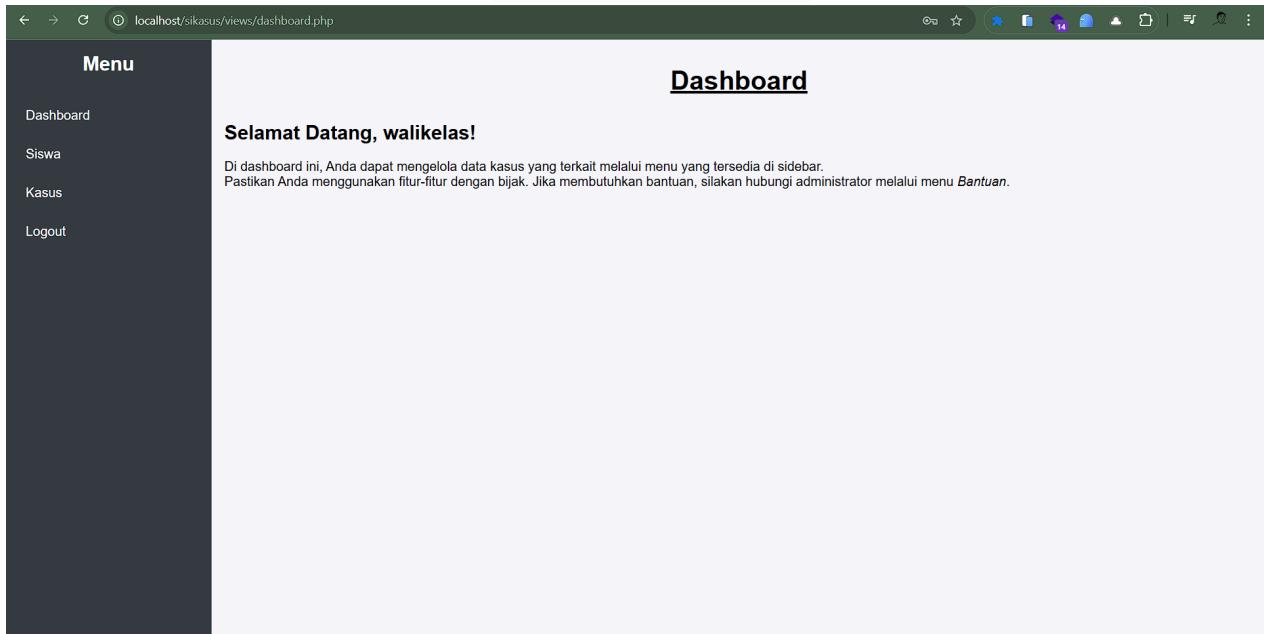
Halaman Dashboard Test

5. **Berikutnya**, ulangi langkah yang sama untuk tipe login **Wali Kelas** dan **Admin**: - Periksa apakah placeholder input berubah menjadi "Masukkan NIP" untuk wali kelas dan "Masukkan Nama Pengguna" untuk admin. - Pastikan hanya pengguna dengan tipe

login yang sesuai dapat mengakses dashboard mereka.



6. Selanjutnya, masukkan kredensial yang benar sesuai data di database. - Jika login berhasil, Anda akan diarahkan ke `views/dashboard.php`.



7. Selanjutnya, uji fitur **Logout**.

- Klik tombol **Logout** (atau kunjungi url `http://localhost/sikasus/logout.php`).
- **Hasil yang diharapkan:** Sesi terhapus, dan pengguna diarahkan kembali ke halaman utama (`index.php`).

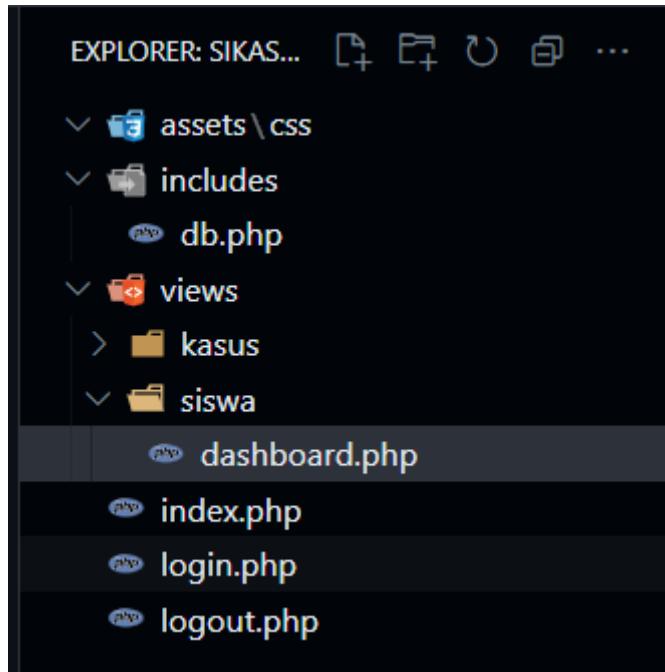
Kesimpulan

Fitur login dalam aplikasi ini memisahkan akses antara siswa dan wali kelas. Hal ini bertujuan untuk memastikan bahwa hanya pengguna dengan peran yang sesuai yang dapat mengakses data dan fitur yang relevan dengan tugas mereka, seperti siswa yang hanya

dapat melihat data pribadi dan wali kelas yang memiliki akses untuk mengelola data kelas dan siswa.

Siswa Hanya Melihat Kasusnya Sendiri

Struktur Folder



Pertama-tama, buat file baru bernama `dashboard.php` di dalam folder `siswa` sesuai struktur proyek yang diberikan.

Selanjutnya, Hubungkan ke Koneksi Database

```
<?php
session_start();

// Cek apakah user sudah login sebagai siswa
if (!isset($_SESSION['user']) || $_SESSION['user']['role'] !== 'siswa') {
    header("Location: ../../login.php");
    exit;
}

$user = $_SESSION['user'];
$role = $user['role']; // role: 'admin', 'walikelas', atau 'siswa'
```

```
<?php
session_start();

// Cek apakah user sudah login sebagai siswa
if (!isset($_SESSION['user']) || $_SESSION['user']['role'] !== 'siswa') {
    header(header: "Location: ../../Login.php");
    exit;
}

$user = $_SESSION['user'];
$role = $user['role']; // role: 'admin', 'walikelas', atau 'siswa'
```

Penjelasan:

- `session_start()` : Menginisialisasi sesi PHP.
- `isset($_SESSION['user'])` : Mengecek apakah sesi pengguna sudah diatur.
- `header("Location: ...")` : Mengarahkan pengguna yang tidak memenuhi syarat login ke halaman login.

Kemudian, Tambahkan logika untuk memastikan halaman hanya dapat diakses oleh siswa yang telah login.

Tambahkan koneksi ke database agar data siswa dapat diambil.

```
require_once __DIR__ . '/../../includes/db.php';
```

```
<?php
session_start();

// Cek apakah user sudah login sebagai siswa
if (!isset($_SESSION['user']) || $_SESSION['user']['role'] !== 'siswa') {
    header(header: "Location: ../../Login.php");
    exit;
}

$user = $_SESSION['user'];
$role = $user['role']; // role: 'admin', 'walikelas', atau 'siswa'

require_once __DIR__ . '/../../includes/db.php';
```

Penjelasan:

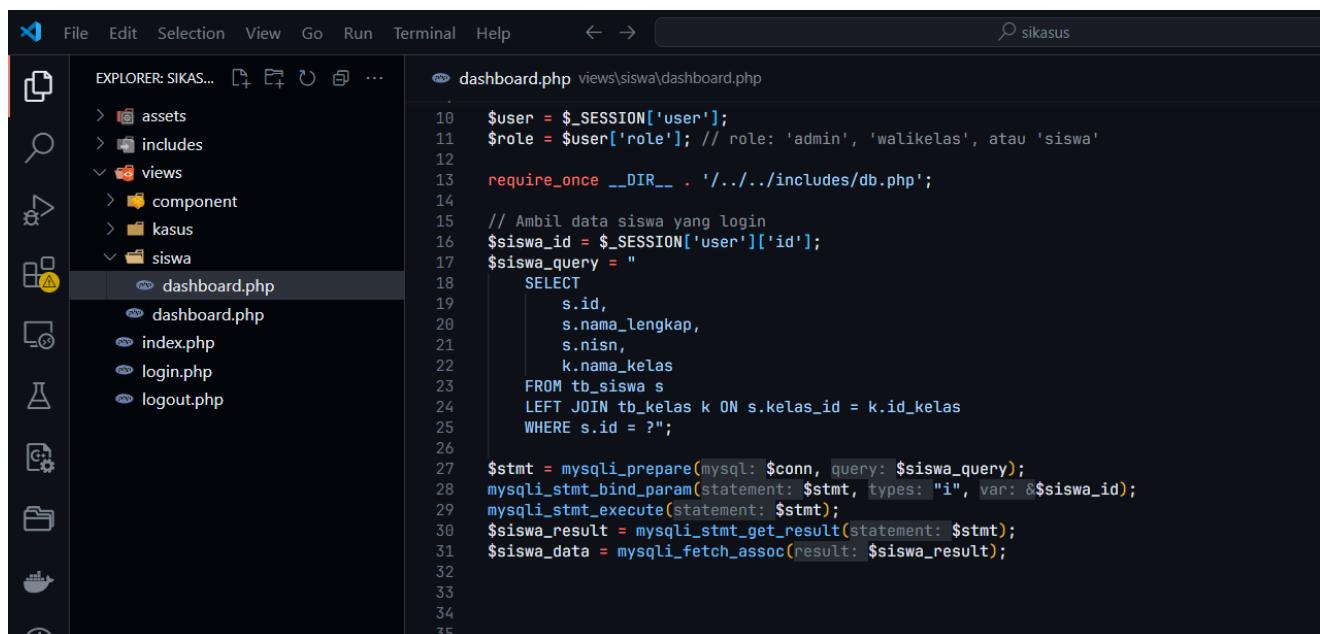
- `require_once` : Memastikan file koneksi hanya dimuat satu kali untuk menghindari duplikasi.
- `__DIR__` : Mengambil lokasi direktori file saat ini agar path tetap valid meski proyek dipindahkan.

Berikutnya, Ambil Data Siswa Berdasarkan ID

Gunakan ID dari sesi untuk mengambil informasi siswa dan relasi kelas mereka.

```
$siswa_id = $_SESSION['user']['id'];
$siswa_query = "
    SELECT
        s.id,
        s.nama_lengkap,
        s.nisn,
        k.nama_kelas
    FROM tb_siswa s
    LEFT JOIN tb_kelas k ON s.kelas_id = k.id_kelas
    WHERE s.id = ?";

$stmt = mysqli_prepare($conn, $siswa_query);
mysqli_stmt_bind_param($stmt, "i", $siswa_id);
mysqli_stmt_execute($stmt);
$siswa_result = mysqli_stmt_get_result($stmt);
$siswa_data = mysqli_fetch_assoc($siswa_result);
```



```
File Edit Selection View Go Run Terminal Help dashboard.php views\siswa\dashboard.php
10     $user = $_SESSION['user'];
11     $role = $user['role']; // role: 'admin', 'walikelas', atau 'siswa'
12
13     require_once __DIR__ . '/../../../../includes/db.php';
14
15     // Ambil data siswa yang login
16     $siswa_id = $_SESSION['user']['id'];
17     $siswa_query = "
18         SELECT
19             s.id,
20             s.nama_lengkap,
21             s.nisn,
22             k.nama_kelas
23         FROM tb_siswa s
24         LEFT JOIN tb_kelas k ON s.kelas_id = k.id_kelas
25         WHERE s.id = ?";
26
27     $stmt = mysqli_prepare(mysqli: $conn, query: $siswa_query);
28     mysqli_stmt_bind_param(statement: $stmt, types: "i", var: $siswa_id);
29     mysqli_stmt_execute(statement: $stmt);
30     $siswa_result = mysqli_stmt_get_result(statement: $stmt);
31     $siswa_data = mysqli_fetch_assoc(result: $siswa_result);
32
33
34
35
```

Query yang digunakan

```
SELECT s.id, s.nama_lengkap, s.nisn, k.nama_kelas FROM tb_siswa s LEFT JOIN
tb_kelas k ON s.kelas_id = k.id_kelas WHERE s.id = ?
```

Penjelasan : Query ini digunakan untuk mengambil data siswa berdasarkan `id` siswa yang disimpan dalam session. Data yang diambil meliputi `id`, `nama_lengkap`, `nism` dari tabel `tb_siswa`, dan `nama_kelas` dari tabel `tb_kelas`. Query menggunakan `LEFT JOIN` untuk menggabungkan tabel `tb_siswa` dengan `tb_kelas` berdasarkan `kelas_id`. `WHERE s.id = ?` memastikan hanya data siswa yang sesuai dengan `id` yang diambil. Parameter

`siswa_id` diikat dengan tipe data integer (`i`), query dijalankan, dan hasilnya disimpan dalam variabel `$siswa_data`.

Penjelasan:

- **Relasi Database:**
 - Tabel `tb_siswa` di-**LEFT JOIN** dengan `tb_kelas` menggunakan kolom `kelas_id`.
 - Data yang diambil mencakup ID siswa, nama lengkap, NISN, dan nama kelas.
- `mysqli_prepare` : Membuat query yang aman dari SQL Injection.

Selanjutnya, Hitung Total Kasus Siswa

Ambil jumlah total kasus yang terkait dengan siswa untuk ditampilkan di dashboard.

```
$total_query = "SELECT COUNT(*) as total FROM tb_kasus WHERE siswa_id = ?";
$stmt = mysqli_prepare($conn, $total_query);
mysqli_stmt_bind_param($stmt, "i", $siswa_id);
mysqli_stmt_execute($stmt);
$total_result = mysqli_stmt_get_result($stmt);
$total_cases = mysqli_fetch_assoc($total_result)['total'];
```

The screenshot shows a code editor interface with a sidebar on the left displaying a file tree. The tree includes 'assets', 'includes', 'views' (which contains 'component' and 'kasus'), and 'siswa' (which contains 'dashboard.php', 'index.php', 'login.php', and 'logout.php'). The main panel shows the content of 'dashboard.php'. The code is as follows:

```
20 |     s.nama_lengkap,
21 |     s.nisn,
22 |     k.nama_kelas
23 | FROM tb_siswa s
24 | LEFT JOIN tb_kelas k ON s.kelas_id = k.id_kelas
25 | WHERE s.id = ?;
26 |
27 | $stmt = mysqli_prepare(mysqli: $conn, query: $siswa_query);
28 | mysqli_stmt_bind_param(statement: $stmt, types: "i", var: &$siswa_id);
29 | mysqli_stmt_execute(statement: $stmt);
30 | $siswa_result = mysqli_stmt_get_result(statement: $stmt);
31 | $siswa_data = mysqli_fetch_assoc(result: $siswa_result);
32 |
33 | // Ambil total kasus siswa
34 | $total_query = "SELECT COUNT(*) as total FROM tb_kasus WHERE siswa_id = ?";
35 | $stmt = mysqli_prepare(mysqli: $conn, query: $total_query);
36 | mysqli_stmt_bind_param(statement: $stmt, types: "i", var: &$siswa_id);
37 | mysqli_stmt_execute(statement: $stmt);
38 | $total_result = mysqli_stmt_get_result(statement: $stmt);
39 | $total_cases = mysqli_fetch_assoc(result: $total_result)['total'];
40 |
41 |
42 |
```

Query yang digunakan

```
SELECT COUNT(*) as total FROM tb_kasus WHERE siswa_id = ?
```

Penjelasan : Query ini digunakan untuk menghitung jumlah total kasus yang terkait dengan siswa tertentu berdasarkan `siswa_id`. Perintah `SELECT COUNT(*) as total` menghitung jumlah baris dalam tabel `tb_kasus` yang memiliki `siswa_id` sesuai dengan parameter yang diberikan. Parameter `siswa_id` diikat dengan tipe data integer (`i`), query dijalankan, dan hasilnya diambil. Jumlah total kasus disimpan dalam variabel `$total_cases`.

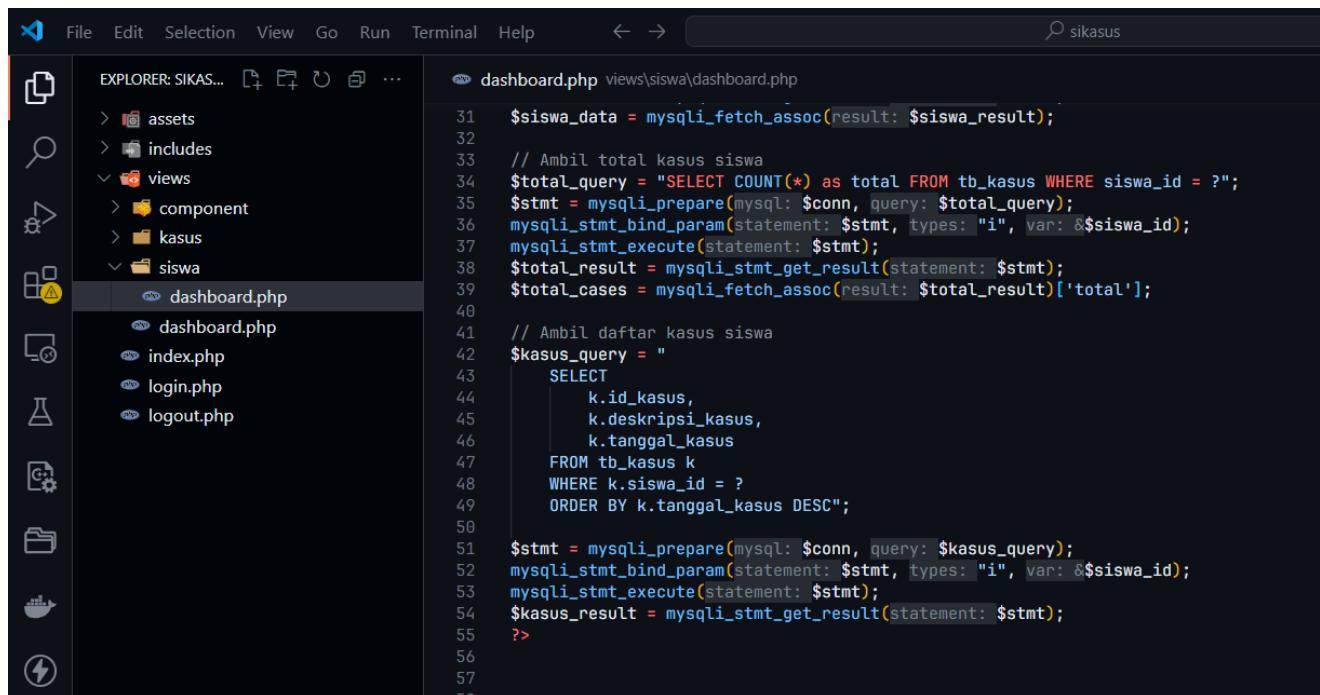
Penjelasan:

- `COUNT(*)` : Menghitung jumlah baris kasus dalam tabel `tb_kasus` untuk siswa tertentu.
- `WHERE siswa_id = ?` : Memastikan hanya kasus siswa yang sedang login yang dihitung.

Selanjutnya, Ambil Daftar Kasus Berdasarkan Siswa

```
$kasus_query = "
SELECT
    k.id_kasus,
    k.deskripsi_kasus,
    k.tanggal_kasus
FROM tb_kasus k
WHERE k.siswa_id = ?
ORDER BY k.tanggal_kasus DESC";

$stmt = mysqli_prepare($conn, $kasus_query);
mysqli_stmt_bind_param($stmt, "i", $siswa_id);
mysqli_stmt_execute($stmt);
$kasus_result = mysqli_stmt_get_result($stmt);
```



The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists project files and folders. The main area displays a PHP script named `dashboard.php`. The code implements a query to count total cases for a specific student and another to fetch a list of cases for that student, ordered by date.

```
File Edit Selection View Go Run Terminal Help ↶ → ⚙️ siskasus
EXPLORER: SIKAS...
> assets
> includes
views
  > component
  > kasus
  > siswa
    dashboard.php
    dashboard.php
    index.php
    login.php
    logout.php
  dashboard.php views\siswa\dashboard.php
31 $siswa_data = mysqli_fetch_assoc(result: $siswa_result);
32
33 // Ambil total kasus siswa
34 $total_query = "SELECT COUNT(*) as total FROM tb_kasus WHERE siswa_id = ?";
35 $stmt = mysqli_prepare(mysql: $conn, query: $total_query);
36 mysqli_stmt_bind_param(statement: $stmt, types: "i", var: &$siswa_id);
37 mysqli_stmt_execute(statement: $stmt);
38 $total_result = mysqli_stmt_get_result(statement: $stmt);
39 $total_cases = mysqli_fetch_assoc(result: $total_result)['total'];
40
41 // Ambil daftar kasus siswa
42 $kasus_query = "
43     SELECT
44         k.id_kasus,
45         k.deskripsi_kasus,
46         k.tanggal_kasus
47     FROM tb_kasus k
48     WHERE k.siswa_id = ?
49     ORDER BY k.tanggal_kasus DESC";
50
51 $stmt = mysqli_prepare(mysql: $conn, query: $kasus_query);
52 mysqli_stmt_bind_param(statement: $stmt, types: "i", var: &$siswa_id);
53 mysqli_stmt_execute(statement: $stmt);
54 $kasus_result = mysqli_stmt_get_result(statement: $stmt);
55 ?>
56
57
58
```

Query yang digunakan

```
SELECT k.id_kasus, k.deskripsi_kasus, k.tanggal_kasus FROM tb_kasus k WHERE
k.siswa_id = ? ORDER BY k.tanggal_kasus DESC
```

Penjelasan : Query ini digunakan untuk mengambil data kasus yang terkait dengan siswa tertentu berdasarkan `siswa_id`. Data yang diambil meliputi `id_kasus`, `deskripsi_kasus`, dan `tanggal_kasus` dari tabel `tb_kasus`. Kondisi `WHERE k.siswa_id = ?` memastikan hanya kasus dengan `siswa_id` yang sesuai yang akan diambil. Hasilnya diurutkan berdasarkan `tanggal_kasus` secara menurun (`DESC`). Parameter `siswa_id` diikat dengan

tipe data integer (`i`), query dijalankan, dan hasilnya disimpan dalam variabel `$kasus_result` .

Penjelasan:

- `ORDER BY k.tanggal_kasus DESC` : Mengurutkan hasil berdasarkan tanggal kasus terbaru.

Terakhir, Tampilkan Data di Halaman `dashboard.php`

```
<!DOCTYPE html>
<html lang="id">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Riwayat Kasus - <?= htmlspecialchars($siswa_data['nama_lengkap']) ?></title>

    <link rel="stylesheet" href="/sikasus/assets/css/styles.css">

</head>

<body>
    <div class="wrapper">
        <!-- Sidebar -->
        <?php include($_SERVER['DOCUMENT_ROOT'] .
'/sikasus/views/component/sidebar.php'); ?>

        <div class="main-content content-siswa">
            <!-- Profil Siswa -->
            <div class="profile-card">
                <div class="profile-header">
                    <div class="profile-avatar">👤 </div>
                    <div class="profile-info">
                        <h1><?=
htmlspecialchars($siswa_data['nama_lengkap']) ?></h1>
                        <p>NISN: <?= htmlspecialchars($siswa_data['nisn']) ?>
</p>
                        <p>Kelas: <?=
htmlspecialchars($siswa_data['nama_kelas'] ?? "Belum ditentukan") ?></p>
                        <p>Total Kasus: <?= $total_cases ?></p>
                    </div>
                </div>
            </div>

            <!-- Riwayat Kasus -->
            <div class="cases-card">
                <h2>Riwayat Kasus</h2>
```

```

<?php if (mysqli_num_rows($kasus_result) > 0): ?>
    <?php while ($kasus =
mysqli_fetch_assoc($kasus_result)): ?>
        <div class="case-item">
            <strong><?= date('d M Y',
strtotime($kasus['tanggal_kasus'])) ?></strong>
            <p><?= nl2br(htmlspecialchars($kasus['deskripsi_kasus'])) ?></p>
        </div>
    <?php endwhile; ?>
    <?php else: ?>
        <div class="empty-state">
            <p>Tidak Ada Riwayat Kasus</p>
        </div>
    <?php endif; ?>
</div>
</div>
</body>

</html>

```

Hasil Tampilan Dashboard Siswa

localhost/siklus/views/siswa/dashboard.php

Menu

Logout

Andi Muh. Raihan Alkawsar

NISN: 1234567894
Kelas: XII PPLG 1
Total Kasus: 39

Riwayat Kasus

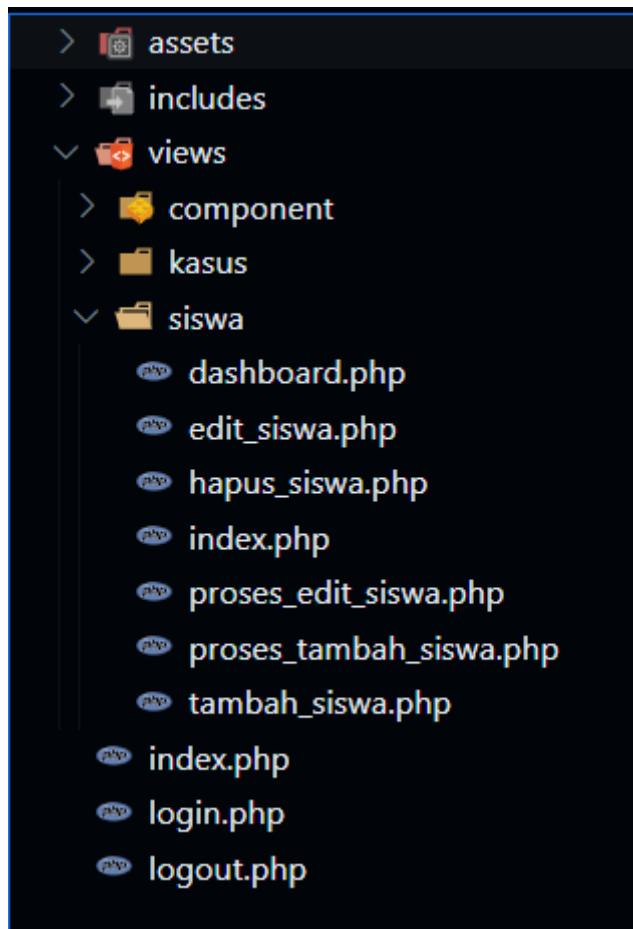
- 05 Dec 2024
Menghilangkan buku perpustakaan.
- 04 Dec 2024
Menggunakan barang milik sekolah tanpa izin.
- 03 Dec 2024
Mengabaikan tugas piket dengan sengaja.
- 01 Dec 2024
Mengganggu teman selama diskusi kelompok berlangsung.
- 30 Nov 2024
Tidak membawa alat tulis saat ulangan harian.
- 29 Nov 2024
Bericara tidak sopan terhadap guru piket.

Kesimpulan

Setiap siswa hanya dapat mengakses dan melihat data kasus yang berkaitan dengan dirinya sendiri. Dengan cara ini, aplikasi menjaga privasi dan relevansi informasi sehingga hanya siswa yang bersangkutan yang bisa mengakses kasus yang melibatkan dirinya.

CRUD Data Siswa

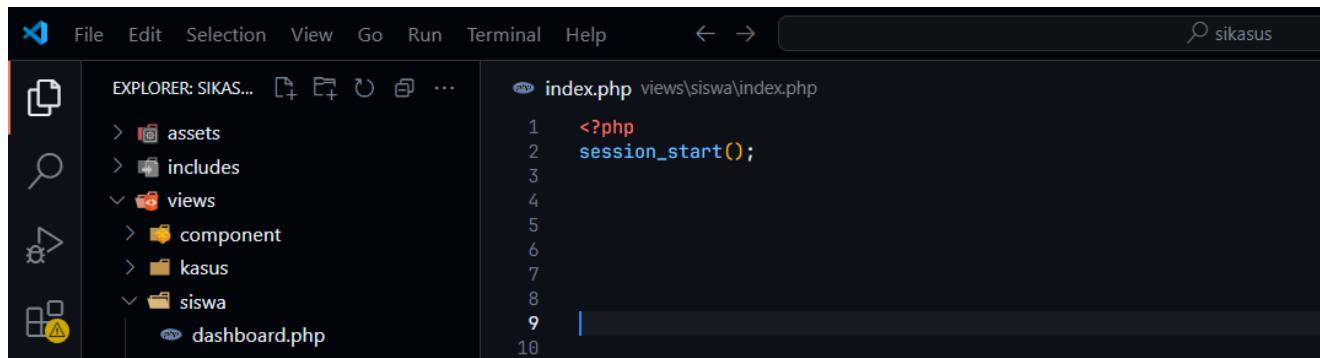
Struktur Folder



Membuat Halaman untuk Menampilkan Data Siswa

Pertama, buat file `index.php` di folder `views/siswa`. Di dalam file ini, kita akan menampilkan daftar siswa yang diambil dari database.

```
<?php  
session_start();
```

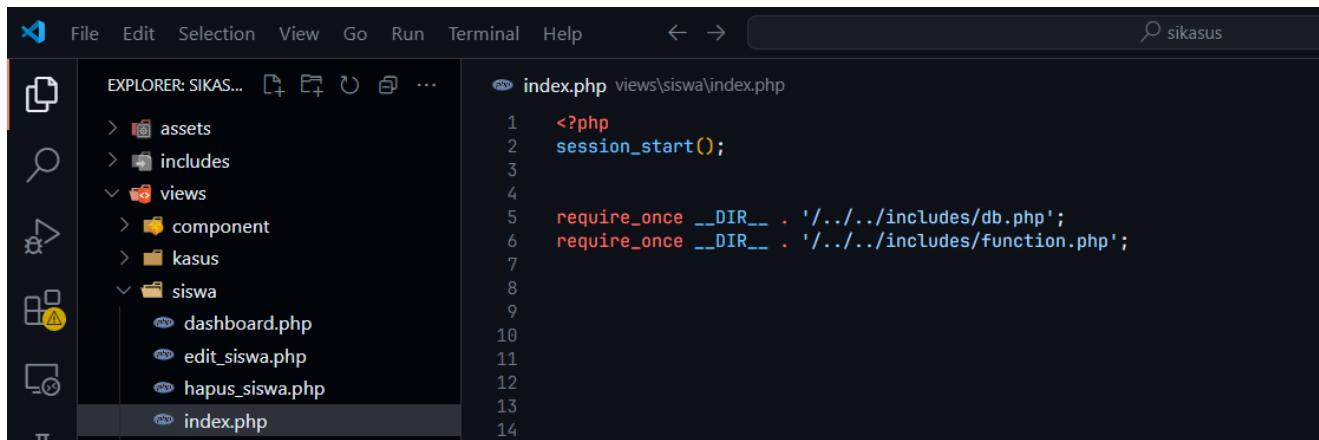


Penjelasan:

- Fungsi `session_start()` diperlukan untuk mengakses data sesi, seperti informasi pengguna yang login.

Selanjutnya, sertakan file db.php dan function.php agar Anda dapat terhubung ke database serta menggunakan fungsi yang telah dibuat.

```
require_once __DIR__ . '/../includes/db.php';
require_once __DIR__ . '/../includes/function.php';
```



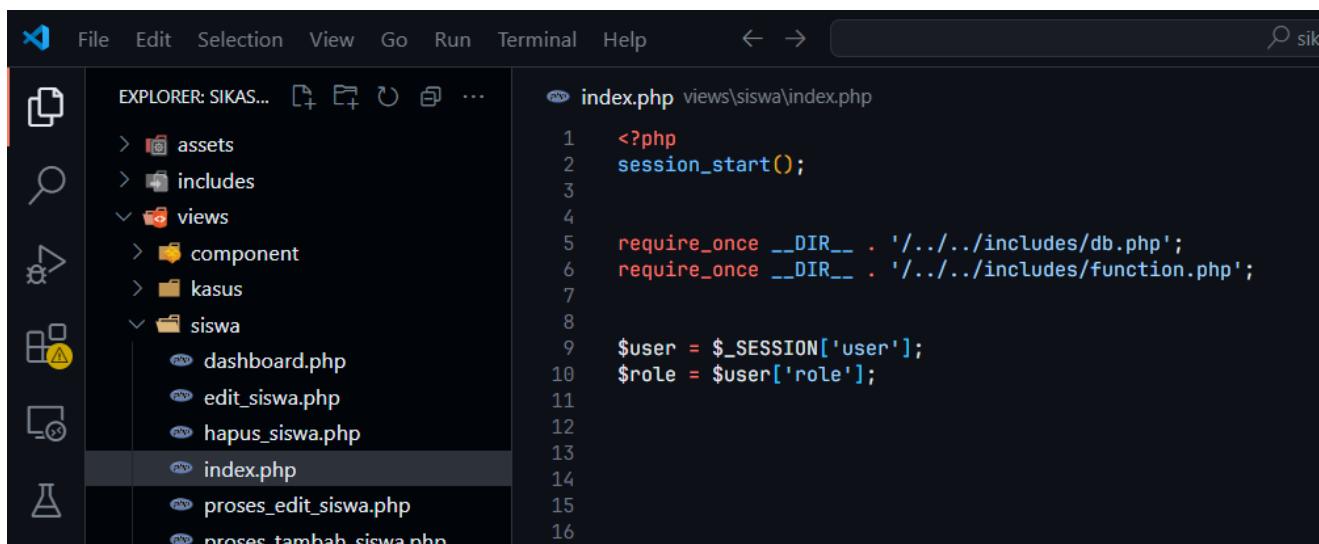
```
index.php views\siswa\index.php
1 <?php
2 session_start();
3
4
5 require_once __DIR__ . '/../includes/db.php';
6 require_once __DIR__ . '/../includes/function.php';
7
8
9
10
11
12
13
14
```

Penjelasan:

- `require_once` memastikan file hanya dimuat satu kali untuk mencegah konflik.
- `__DIR__` menunjuk direktori tempat file saat ini berada, sehingga memudahkan mengatur path.

Kemudian, ambil data pengguna dari sesi untuk menentukan hak akses berdasarkan perannya.

```
$user = $_SESSION['user'];
$role = $user['role'];
```



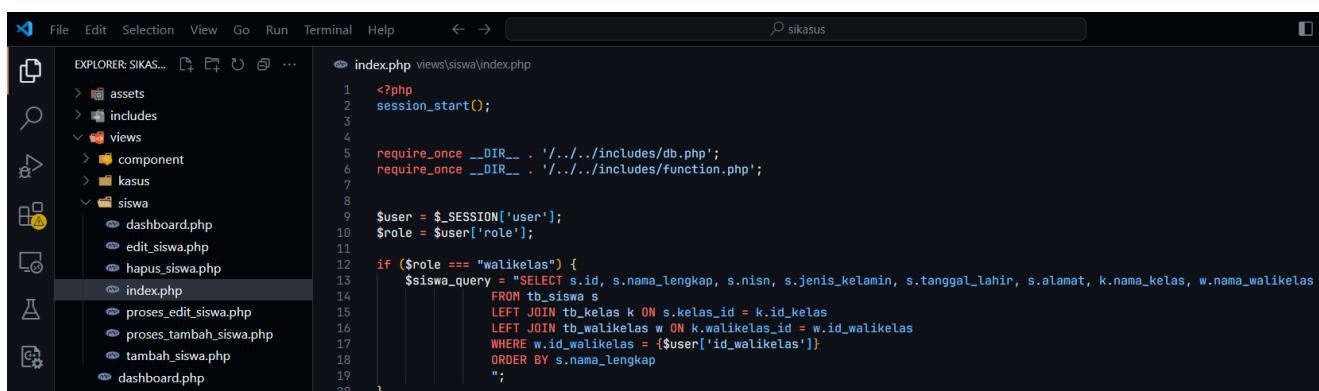
```
index.php views\siswa\index.php
1 <?php
2 session_start();
3
4
5 require_once __DIR__ . '/../includes/db.php';
6 require_once __DIR__ . '/../includes/function.php';
7
8
9 $user = $_SESSION['user'];
10 $role = $user['role'];
11
12
13
14
15
16
```

Penjelasan:

- Variabel `$user` menyimpan data pengguna yang sedang login.
- Variabel `$role` digunakan untuk mengetahui apakah pengguna adalah wali kelas.

Berikutnya, buat query untuk mengambil data siswa yang hanya terkait dengan wali kelas yang sedang login.

```
if ($role === "walikelas") {  
    $siswa_query = "SELECT s.id, s.nama_lengkap, s.nisn, s.jenis_kelamin,  
    s.tanggal_lahir, s.alamat, k.nama_kelas, w.nama_walikelas  
    FROM tb_siswa s  
    LEFT JOIN tb_kelas k ON s.kelas_id = k.id_kelas  
    LEFT JOIN tb_walikelas w ON k.walikelas_id = w.id_walikelas  
    WHERE w.id_walikelas = {$user['id_walikelas']}  
    ORDER BY s.nama_lengkap";  
}
```



The screenshot shows a code editor interface with a dark theme. On the left is the file explorer showing a project structure with files like dashboard.php, edit_siswa.php, hapus_siswa.php, index.php, proses_edit_siswa.php, proses_tambah_siswa.php, tambah_siswa.php, and dashboard.php. The main pane displays the PHP code for 'index.php'. The SQL query is highlighted in pink. The code includes session handling, database connection, and the specific query for students taught by the logged-in teacher.

```
<?php  
session_start();  
  
require_once __DIR__ . '../../../../../includes/db.php';  
require_once __DIR__ . '../../../../../includes/function.php';  
  
$user = $_SESSION['user'];  
$role = $user['role'];  
  
if ($role === "walikelas") {  
    $siswa_query = "SELECT s.id, s.nama_lengkap, s.nisn, s.jenis_kelamin, s.tanggal_lahir, s.alamat, k.nama_kelas, w.nama_walikelas  
    FROM tb_siswa s  
    LEFT JOIN tb_kelas k ON s.kelas_id = k.id_kelas  
    LEFT JOIN tb_walikelas w ON k.walikelas_id = w.id_walikelas  
    WHERE w.id_walikelas = {$user['id_walikelas']}  
    ORDER BY s.nama_lengkap";  
}
```

Query yang digunakan

```
SELECT s.id, s.nama_lengkap, s.nisn, s.jenis_kelamin, s.tanggal_lahir,  
s.alamat, k.nama_kelas, w.nama_walikelas FROM tb_siswa s LEFT JOIN tb_kelas  
k ON s.kelas_id = k.id_kelas LEFT JOIN tb_walikelas w ON k.walikelas_id =  
w.id_walikelas WHERE w.id_walikelas = {$user['id_walikelas']} ORDER BY  
s.nama_lengkap
```

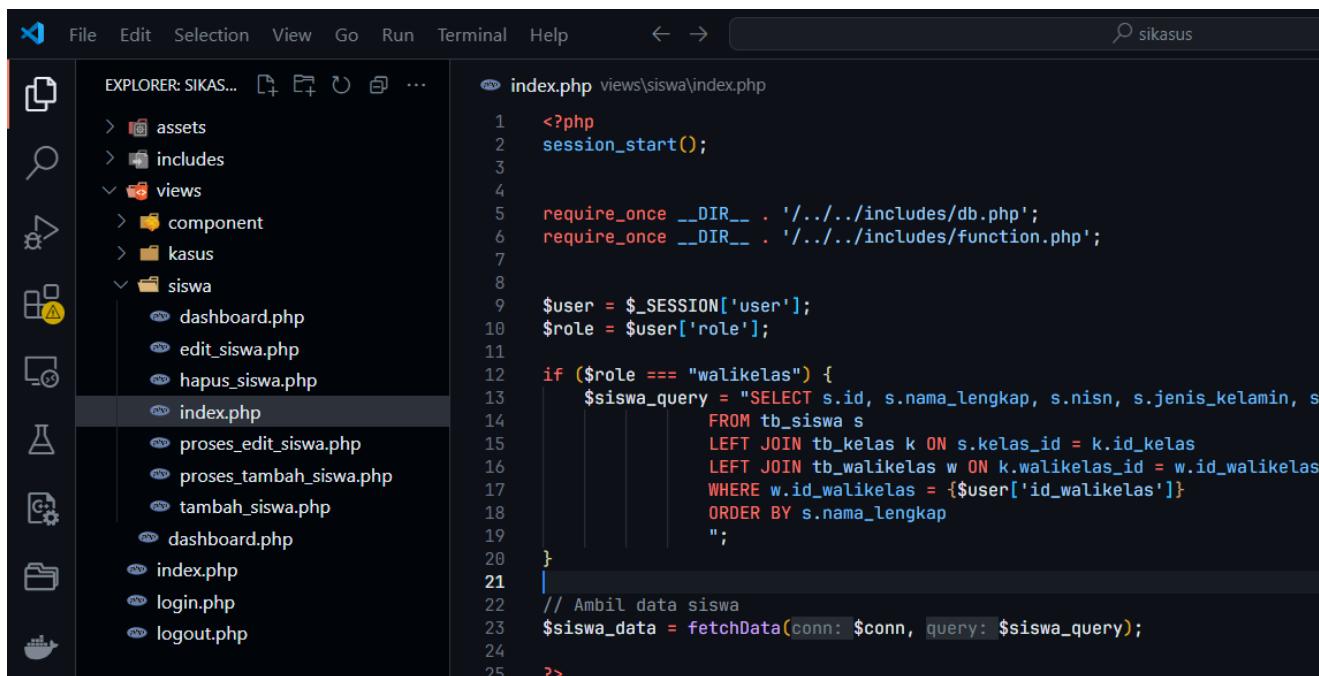
Penjelasan : Query ini digunakan untuk mengambil data siswa yang diajar oleh wali kelas tertentu. Data yang diambil meliputi `id`, `nama_lengkap`, `nism`, `jenis_kelamin`, `tanggal_lahir`, `alamat` dari tabel `tb_siswa`, serta `nama_kelas` dari tabel `tb_kelas` dan `nama_walikelas` dari tabel `tb_walikelas`. Tabel `tb_siswa` digabungkan dengan `tb_kelas` menggunakan `LEFT JOIN` berdasarkan `kelas_id`, dan `tb_kelas` digabungkan dengan `tb_walikelas` berdasarkan `walikelas_id`. Kondisi `WHERE w.id_walikelas = {$user['id_walikelas']}` memastikan hanya siswa yang diajar oleh wali kelas yang sesuai yang diambil. Data diurutkan berdasarkan `nama_lengkap` siswa.

Penjelasan:

- Query SQL ini berguna untuk mengambil data siswa berdasarkan ID wali kelas yang login.
- Tabel `tb_siswa` bergabung (join) dengan tabel `tb_kelas` dan `tb_walikelas` untuk mendapatkan nama kelas serta nama wali kelas.

Setelah query ditulis, eksekusi query tersebut dan simpan hasilnya.

```
$siswa_data = fetchData($conn, $siswa_query);
```



```
<?php
session_start();
require_once __DIR__ . '/../../includes/db.php';
require_once __DIR__ . '/../../includes/function.php';

$user = $_SESSION['user'];
$role = $user['role'];

if ($role === "walikelas") {
    $siswa_query = "SELECT s.id, s.nama_lengkap, s.nisn, s.jenis_kelamin, s
                    FROM tb_siswa s
                    LEFT JOIN tb_kelas k ON s.kelas_id = k.id_kelas
                    LEFT JOIN tb_walikelas w ON k.walikelas_id = w.id_walikelas
                    WHERE w.id_walikelas = {$user['id_walikelas']}
                    ORDER BY s.nama_lengkap
";
}

// Ambil data siswa
$siswa_data = fetchData(conn: $conn, query: $siswa_query);
?>
```

Penjelasan:

- Fungsi `fetchData` adalah fungsi bantu dari `function.php` yang mengeksekusi query dan mengembalikan data dalam bentuk array.

Selanjutnya, buat struktur dasar HTML untuk menampilkan daftar siswa.

```
<!DOCTYPE html>
<html lang="id">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Daftar Siswa</title>
        <link rel="stylesheet" href="/sikasus/assets/css/styles.css" />
    </head>
```

```

<body>
    <div class="wrapper"></div>
</body>
</html>

```

The screenshot shows the VS Code interface with the following details:

- File Explorer (Left):** Shows the project structure under 'EXPLORER: SIKAS...'. It includes 'assets', 'includes', and a 'views' folder containing 'component', 'kasus', and 'siswa'. Inside 'siswa', there are files: 'dashboard.php', 'edit_siswa.php', 'hapus_siswa.php', 'index.php', 'proses_edit_siswa.php', 'proses_tambah_siswa.php', 'tambah_siswa.php', and 'login.php'.
- Code Editor (Right):** Displays the content of 'index.php' located at 'views/siswa/index.php'. The code is as follows:

```

12 if ($role === "walikelas") {
13     ORDER BY s.nama_lengkap
14     ";
15 }
16
17 // Ambil data siswa
18 $siswa_data = fetchData(conn: $conn, query: $siswa_query);
19
20 ?>
21 <!DOCTYPE html>
22 <html lang="id">
23
24 <head>
25     <meta charset="UTF-8">
26     <meta name="viewport" content="width=device-width, initial-scale=1.0">
27     <title>Daftar Siswa</title>
28     <link rel="stylesheet" href="/assets/css/styles.css">
29 </head>
30
31 <body>
32     <div class="wrapper">
33
34         <!-- Sidebar -->
35         <?php include($_SERVER['DOCUMENT_ROOT'] . '/views/component/sidebar.php'); ?>
36
37     </div>
38
39
40
41
42

```

Penjelasan:

- Tag `<html>` dengan atribut `lang="id"` menunjukkan bahwa bahasa halaman ini adalah Bahasa Indonesia.
- File CSS eksternal `styles.css` ditautkan untuk memberikan gaya pada halaman.

Lalu, tambahkan sidebar menggunakan file `sidebar.php`.

```

<?php include($_SERVER['DOCUMENT_ROOT'])
.'sikasus/views/component/sidebar.php'); ?>

```

The screenshot shows the VS Code interface with the following details:

- File Explorer (Left):** Shows the project structure under 'EXPLORER: SIKAS...'. It includes 'assets', 'includes', and a 'views' folder containing 'component', 'kasus', and 'siswa'. Inside 'siswa', there are files: 'dashboard.php', 'edit_siswa.php', 'hapus_siswa.php', 'index.php', 'proses_edit_siswa.php', 'proses_tambah_siswa.php', 'tambah_siswa.php', 'index.php', 'login.php', and 'logout.php'.
- Code Editor (Right):** Displays the content of 'index.php' located at 'views/siswa/index.php'. The code is identical to the previous screenshot, but the final line has been added:

```

12 if ($role === "walikelas") {
13     ORDER BY s.nama_lengkap
14     ";
15 }
16
17 // Ambil data siswa
18 $siswa_data = fetchData(conn: $conn, query: $siswa_query);
19
20 ?>
21 <!DOCTYPE html>
22 <html lang="id">
23
24 <head>
25     <meta charset="UTF-8">
26     <meta name="viewport" content="width=device-width, initial-scale=1.0">
27     <title>Daftar Siswa</title>
28     <link rel="stylesheet" href="/assets/css/styles.css">
29 </head>
30
31 <body>
32     <div class="wrapper">
33
34         <!-- Sidebar -->
35         <?php include($_SERVER['DOCUMENT_ROOT'] . '/views/component/sidebar.php'); ?>
36
37     </div>
38
39
40
41
42

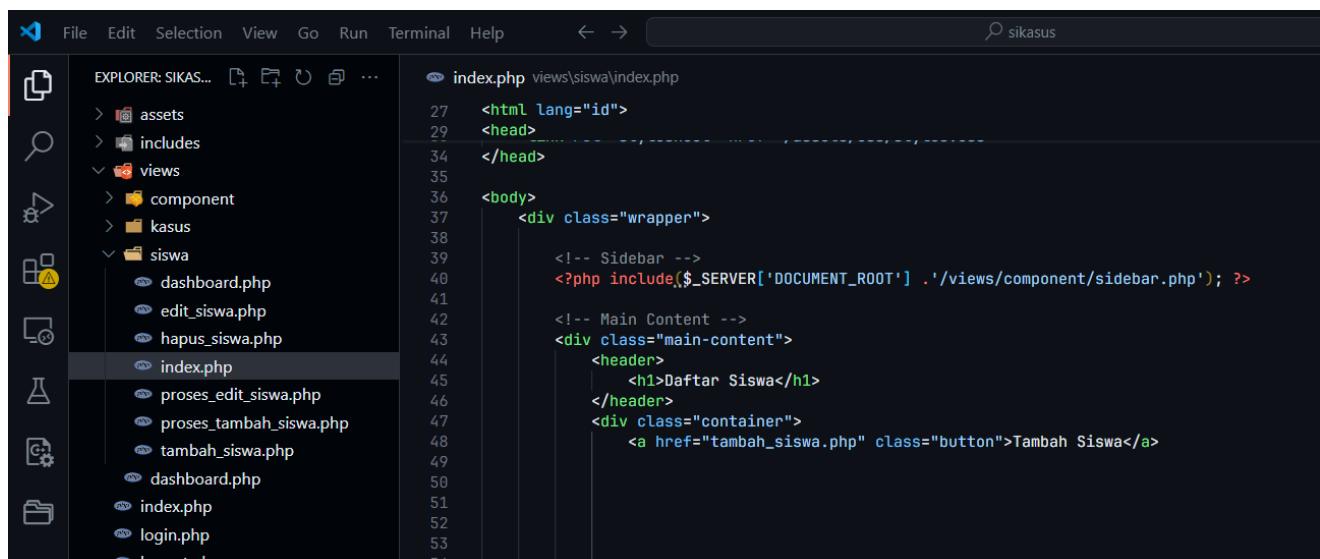
```

Penjelasan:

- Fungsi `include` digunakan untuk memasukkan file sidebar agar kode lebih modular.
 - `$_SERVER['DOCUMENT_ROOT']` menunjuk ke direktori root proyek, sehingga path tetap dinamis.
-

Setelah itu, tambahkan header dan tombol untuk menambah siswa.

```
<div class="main-content">
  <header>
    <h1>Daftar Siswa</h1>
  </header>
  <div class="container">
    <a href="tambah_siswa.php" class="button">Tambah Siswa</a>
  </div>
</div>
```



The screenshot shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer is titled 'EXPLORER: SIKAS...' and shows the project structure:

- assets
- includes
- views
 - component
 - kasus
 - siswa
 - dashboard.php
 - edit_siswa.php
 - hapus_siswa.php
 - index.php** (selected)
 - proses_edit_siswa.php
 - proses_tambah_siswa.php
 - tambah_siswa.php
 - dashboard.php
 - index.php
 - login.php
 - logout.php

The code editor displays the content of 'index.php' with line numbers:

```
27  <html lang="id">
28  <head>
29  <!-- Sidebar -->
30  <?php include($_SERVER['DOCUMENT_ROOT'] .'/views/component/sidebar.php'); ?>
31  <!-- Main Content -->
32  <div class="main-content">
33    <header>
34      <h1>Daftar Siswa</h1>
35    </header>
36    <div class="container">
37      <a href="tambah_siswa.php" class="button">Tambah Siswa</a>
38    </div>
39  </div>
```

Penjelasan:

- Bagian ini berisi elemen visual seperti judul halaman dan tombol untuk menambah data siswa.
-

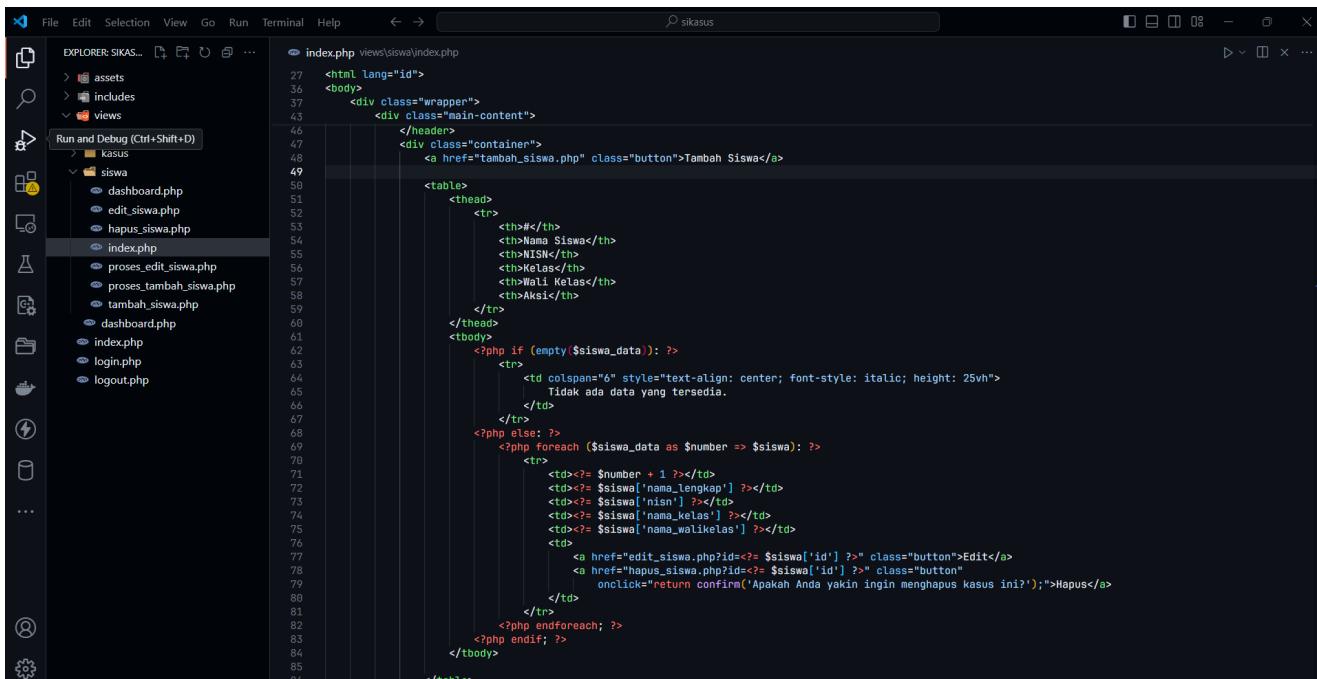
Kemudian, buat tabel untuk menampilkan data siswa dari hasil query.

```
<table>
  <thead>
    <tr>
      <th>#</th>
      <th>Nama Siswa</th>
```

```

<th>NISN</th>
<th>Kelas</th>
<th>Wali Kelas</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<?php if (empty($siswa_data)): ?>
<tr>
<td
    colspan="6"
    style="text-align: center; font-style: italic; height: 25vh"
>
    Tidak ada data yang tersedia.
</td>
</tr>
<?php else: ?>
<?php foreach ($siswa_data as $number =>
$siswa): ?>
<tr>
<td><?= $number + 1 ?></td>
<td><?= $siswa['nama_lengkap'] ?></td>
<td><?= $siswa['nisn'] ?></td>
<td><?= $siswa['nama_kelas'] ?></td>
<td><?= $siswa['nama_walikelas'] ?></td>
<td>
    <a href="edit_siswa.php?id=<?= $siswa['id'] ?>">Edit</a>
    <a href="hapus_siswa.php?id=<?= $siswa['id'] ?>">Hapus</a>
    >
</td>
</tr>
<?php endforeach; ?>
<?php endif; ?>
</tbody>
</table>

```



The screenshot shows a code editor interface with the following details:

- File Explorer (Left):** Shows a project structure with a folder named "SIKAS..." containing "assets", "includes", and "views". The "views/siswa/" folder is expanded, showing files: "index.php", "dashboard.php", "edit_siswa.php", "hapus_siswa.php", "index.php", "proses_edit_siswa.php", "proses_tambah_siswa.php", and "tambah_siswa.php".
- Code Editor (Right):** Displays the content of "index.php" under the "views/siswa/" folder. The code is a PHP script that generates an HTML table to list students. It includes columns for ID, Name, NISN, Class, and Teacher. It also includes links for editing and deleting each student.

```
<?php if (empty($siswa_data)): ?>
<tr>
    <td colspan="6" style="text-align: center; font-style: italic; height: 25vh">
        Tidak ada data yang tersedia.
    </td>
</tr>
<?php else: ?>
<php foreach ($siswa_data as $number => $siswa): ?>
<t>
    <td><?= $number + 1 ?></td>
    <td><?= $siswa['nama_lengkap'] ?></td>
    <td><?= $siswa['nisn'] ?></td>
    <td><?= $siswa['nama_kelas'] ?></td>
    <td><?= $siswa['nama_walikelas'] ?></td>
    <td>
        <a href="edit_siswa.php?id=<?= $siswa['id'] ?>" class="button">Edit</a>
        <a href="hapus_siswa.php?id=<?= $siswa['id'] ?>" class="button"
            onclick="return confirm('Apakah Anda ingin menghapus kasus ini?');">Hapus</a>
    </td>
</tr>
<?php endforeach; ?>
<?php endif; ?>
</tbody>
</table>
```

Penjelasan:

- Tabel ini menampilkan daftar siswa dengan nomor, nama, NISN, kelas, dan wali kelas.
- Tombol "Edit" mengarahkan ke halaman `edit_siswa.php` dan tombol "Hapus" menghapus data siswa.

Terakhir, tutup struktur HTML dengan tag penutup.

```
</div>
</div>
</body>
</html>
```

Penjelasan:

- Tag penutup memastikan bahwa dokumen HTML Anda sudah selesai.

Membuat Halaman Tambah Data Siswa

Pertama-tama, buat file baru dengan nama `tambah_siswa.php` di dalam folder `views/siswa/`. File ini akan digunakan untuk menampilkan formulir tambah data siswa.

```
<?php
session_start();
```

```
EXPLORER: SIKAS...  ⌂ ⌃ ⌚ ⌚ ...  
> assets  
> includes  
views  
    > component  
    > kasus  
    siswa  
        dashboard.php  
        edit_siswa.php  
        hapus_siswa.php  
        index.php  
        proses_edit_siswa.php
```

```
tambah_siswa.php views\kasus\tambah_siswa.php  
1 <?php  
2 session_start();  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16
```

Penjelasan:

- `session_start()` digunakan untuk mengakses atau memulai sesi yang sudah ada.
-

Selanjutnya, tambahkan file `db.php` untuk koneksi ke database dan `function.php` untuk menggunakan fungsi bantu.

```
require_once __DIR__ . '/../../includes/db.php';  
require_once __DIR__ . '/../../includes/function.php';
```

```
tambah_siswa.php views\siswa\tambah_siswa.php  
1 <?php  
2 session_start();  
3  
4  
5 require_once __DIR__ . '/../../includes/db.php';  
6 require_once __DIR__ . '/../../includes/function.php';  
7  
8  
9  
10  
11
```

Penjelasan:

- `db.php` : Mengelola koneksi ke database.
 - `function.php` : Berisi fungsi seperti `fetchData` untuk pengambilan data.
-

Berikutnya, periksa apakah pengguna adalah wali kelas. Jika ya, ambil data kelas yang terkait.

```
$user = $_SESSION['user'];
$role = $user['role'];

if ($role === "walikelas") {
    $kelas_query = "SELECT id_kelas FROM tb_kelas WHERE walikelas_id =
{$user['id_walikelas']}";
}

$kelas_data = fetchData($conn, $kelas_query);
```

```
tambah_siswa.php views\siswa\tambah_siswa.php
1 <?php
2 session_start();
3
4
5 require_once __DIR__ . '/../../includes/db.php';
6 require_once __DIR__ . '/../../includes/function.php';
7
8 $user = $_SESSION['user'];
9 $role = $user['role'];
10
11 if ($role === "admin") {
12     $kelas_query = "SELECT id_kelas, nama_kelas FROM tb_kelas";
13 } elseif ($role === "walikelas") {
14     $kelas_query = "SELECT id_kelas FROM tb_kelas WHERE walikelas_id = {$user['id_walikelas']}";
15 }
16
17 $kelas_data = fetchData(conn: $conn, query: $kelas_query); // Gunakan fetchData untuk hasilkan array data
18 ?>
19
20
21
```

Query yang digunakan

```
SELECT id_kelas FROM tb_kelas WHERE walikelas_id = {$user['id_walikelas']}
```

Penjelasan : Query ini digunakan untuk mengambil `id_kelas` dari tabel `tb_kelas` berdasarkan `walikelas_id` yang sesuai dengan ID wali kelas yang saat ini sedang login. Kondisi `WHERE walikelas_id = {$user['id_walikelas']}` memastikan hanya kelas yang diajar oleh wali kelas tersebut yang diambil. Setelah query disiapkan, fungsi `fetchData()` digunakan untuk mengeksekusi query dan mengambil data kelas yang sesuai, yang kemudian disimpan dalam variabel `$kelas_data`.

Penjelasan:

- Periksa peran pengguna dari sesi (`$_SESSION['user']`).
- Jika perannya adalah wali kelas, jalankan query untuk mengambil `id_kelas` dari `tb_kelas`.

Kemudian, buat struktur HTML untuk halaman.

```
<!DOCTYPE html>
<html lang="id">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Tambah Siswa</title>
        <link rel="stylesheet" href="/assets/css/styles.css" />
    </head>
    <body>
        <div class="wrapper"></div>
    </body>
</html>
```

```
tambah_siswa.php views\siswa\tambah_siswa.php
C:\xampp\htdocs\sikasus\views\siswa\tambah_siswa.php
14     $kelas_query = "SELECT * FROM tb_kelas WHERE walikelas_id = {$user['id_walikelas']}";
15 }
16
17 $kelas_data = fetchData($conn, $query); // Gunakan fetchData untuk hasilkan array data
18 ?>
19
20
21 <!DOCTYPE html>
22 <html lang="id">
23
24     <head>
25         <meta charset="UTF-8">
26         <meta name="viewport" content="width=device-width, initial-scale=1.0">
27         <title>Tambah Siswa</title>
28         <link rel="stylesheet" href="/sikasus/assets/css/styles.css" />
29     </head>
30
31     <body>
32         <div class="wrapper">
```

Penjelasan:

- `<!DOCTYPE html>` menentukan jenis dokumen sebagai HTML5.
- `<link>` menautkan file CSS untuk mempercantik tampilan.

Selanjutnya, tambahkan menu navigasi menggunakan file `sidebar.php`.

```
<?php include($_SERVER['DOCUMENT_ROOT'] . '/views/component/sidebar.php'); ?>
```

```
tambah_siswa.php views\siswa\tambah_siswa.php
C:\xampp\htdocs\sikasus\views\siswa\tambah_siswa.php
14     $kelas_query = "SELECT * FROM tb_kelas WHERE walikelas_id = {$user['id_walikelas']}";
15 }
16
17 $kelas_data = fetchData(conn: $conn, query: $kelas_query); // Gunakan fetch_data untuk hasilkan array data
18 ?>
19
20
21 <!DOCTYPE html>
22 <html lang="id">
23
24 <head>
25     <meta charset="UTF-8">
26     <meta name="viewport" content="width=device-width, initial-scale=1.0">
27     <title>Tambah Siswa</title>
28     <link rel="stylesheet" href="/sikasus/assets/css/styles.css">
29 </head>
30
31 <body>
32     <div class="wrapper">
```

Penjelasan:

- `include()` memasukkan file `sidebar.php` agar kode lebih modular.
-

Selanjutnya, tambahkan form untuk input data siswa.

```
<!-- Main Content -->
<div class="main-content">
<header>
<h1>Tambah Data Siswa</h1>
</header>
<div class="container">
<form action="proses_tambah_siswa.php" method="post">
    <div>
        <label for="nama_lengkap">Nama Lengkap:</label>
        <input type="text" id="nama_lengkap" name="nama_lengkap" required>
    </div>

    <div>
        <label for="nisn">NISN:</label>
        <input type="text" id="nisn" name="nisn" required>
    </div>

    <div>
        <label for="jenis_kelamin">Jenis Kelamin:</label>
        <select id="jenis_kelamin" name="jenis_kelamin" required>
            <option value="">Pilih Jenis Kelamin</option>
            <option value="Laki-laki">Laki-laki</option>
            <option value="Perempuan">Perempuan</option>
        </select>
    </div>
</div>
```

```

<div>
    <label for="tanggal_lahir">Tanggal Lahir:</label>
    <input type="date" id="tanggal_lahir" name="tanggal_lahir" required>
</div>

<div>
    <label for="alamat">Alamat:</label>
    <textarea id="alamat" name="alamat" required></textarea>
</div>

<?php if ($role === "walikelas"): ?>
    <input type="hidden" name="kelas_id" value="<?= $kelas_data[0]
['id_kelas'] ?>">
<?php endif; ?>

<button type="submit">Simpan</button>
<a href="index.php" class="button">Kembali</a>
</form>
</div>
</div>
</div>

```

```

tambah_siswa.php views\siswa\tambah_siswa.php
22 <html lang="id">
31 <body>
32     <div class="wrapper">
37         <!-- Main Content -->
38         <div class="main-content">
39             <header>
40                 <h1>Tambah Data Siswa</h1>
41             </header>
42             <div class="container">
43                 <form action="proses_tambah_siswa.php" method="post">
44                     <div>
45                         <label for="nama_lengkap">Nama Lengkap:</label>
46                         <input type="text" id="nama_lengkap" name="nama_lengkap" required>
47                     </div>
48
49                     <div>
50                         <label for="nisn">NISN:</label>
51                         <input type="text" id="nisn" name="nisn" required>
52                     </div>
53
54                     <div>
55                         <label for="jenis_kelamin">Jenis Kelamin:</label>
56                         <select id="jenis_kelamin" name="jenis_kelamin" required>
57                             <option value="">Pilih Jenis Kelamin</option>
58                             <option value="Laki-laki">Laki-laki</option>
59                             <option value="Perempuan">Perempuan</option>
60                         </select>
61                     </div>
62
63                     <div>
64                         <label for="tanggal_lahir">Tanggal Lahir:</label>
65                         <input type="date" id="tanggal_lahir" name="tanggal_lahir" required>
66                     </div>
67
68                     <div>
69                         <label for="alamat">Alamat:</label>
70                         <textarea id="alamat" name="alamat" required></textarea>
71                     </div>

```

Penjelasan:

- Setiap input field memiliki atribut `name` agar data dapat diakses di file proses.

- Tambahkan input tersembunyi (`hidden input`) untuk memasukkan ID kelas jika pengguna adalah wali kelas.
-

Kemudian, mengirim Data ke `proses_tambah_siswa.php` . Formulir ini akan mengirimkan data menggunakan metode POST ke file `proses_tambah_siswa.php` , yang akan menangani penyimpanan data ke dalam database.

```
<?php  
session_start();
```

Tambahkan `db.php` untuk koneksi database dan `function.php` untuk fungsi bantu.

```
include __DIR__ . '/../../includes/db.php';  
include __DIR__ . '/../../includes/function.php';
```

Ambil data dari formulir dan validasi inputnya.

```
$nama_lengkap = mysqli_real_escape_string($conn, $_POST['nama_lengkap']);  
$nisn = mysqli_real_escape_string($conn, $_POST['nisn']);  
$jenis_kelamin = mysqli_real_escape_string($conn, $_POST['jenis_kelamin']);  
$tanggal_lahir = mysqli_real_escape_string($conn, $_POST['tanggal_lahir']);  
$alamat = mysqli_real_escape_string($conn, $_POST['alamat']);  
$kelas_id = mysqli_real_escape_string($conn, $_POST['kelas_id']);
```

Selanjutnya, buat query untuk menyimpan data ke tabel `tb_siswa` .

```
$insert_query = "INSERT INTO tb_siswa (nama_lengkap, nisn, jenis_kelamin,  
tanggal_lahir, alamat, kelas_id)  
VALUES ('$nama_lengkap', '$nisn', '$jenis_kelamin',  
'$tanggal_lahir', '$alamat', '$kelas_id');
```

```
if (mysqli_query($conn, $insert_query)) {  
    header("Location: ./");  
} else {  
    header("Location: ./");  
}  
exit();
```

Query yang digunakan

```
INSERT INTO tb_siswa (nama_lengkap, nisn, jenis_kelamin, tanggal_lahir, alamat, kelas_id) VALUES ('$nama_lengkap', '$nisn', '$jenis_kelamin', '$tanggal_lahir', '$alamat', '$kelas_id')
```

Penjelasan : Query ini digunakan untuk menambahkan data siswa baru ke dalam tabel `tb_siswa`. Data yang dimasukkan meliputi `nama_lengkap`, `nisn`, `jenis_kelamin`, `tanggal_lahir`, `alamat`, dan `kelas_id`. Nilai-nilai ini dimasukkan melalui variabel PHP yang diteruskan ke query. Setelah query dijalankan menggunakan `mysqli_query()`, jika berhasil, pengguna akan diarahkan kembali ke halaman utama (`header("Location: ./")`). Jika gagal, proses yang sama dilakukan, meskipun tidak ada penanganan kesalahan yang lebih lanjut di sini. Setelah itu, `exit()` digunakan untuk menghentikan eksekusi lebih lanjut dari skrip.

Penjelasan:

- Query SQL `INSERT INTO` digunakan untuk menambah data baru ke tabel `tb_siswa`.
- Jika query berhasil, pengguna diarahkan kembali ke halaman `index.php`.

```
proses_tambah_siswa.php views\siswa\proses_tambah_siswa.php
1  <?php
2  session_start();
3
4  include __DIR__ . '../../../../../includes/db.php';
5  include __DIR__ . '../../../../../includes/function.php';
6
7  $user = $_SESSION['user'];
8  $role = $user['role'];
9
10 // Validasi input
11 $nama_lengkap = mysqli_real_escape_string(mysql: $conn, string: $_POST['nama_lengkap']);
12 $nisn = mysqli_real_escape_string(mysql: $conn, string: $_POST['nisn']);
13 $jenis_kelamin = mysqli_real_escape_string(mysql: $conn, string: $_POST['jenis_kelamin']);
14 $tanggal_lahir = mysqli_real_escape_string(mysql: $conn, string: $_POST['tanggal_lahir']);
15 $alamat = mysqli_real_escape_string(mysql: $conn, string: $_POST['alamat']);
16 $kelas_id = mysqli_real_escape_string(mysql: $conn, string: $_POST['kelas_id']);
17
18 // Masukkan data ke database
19 $insert_query = "INSERT INTO tb_siswa (nama_lengkap, nisn, jenis_kelamin, tanggal_lahir, alamat, kelas_id)
20 |           VALUES ('$nama_lengkap', '$nisn', '$jenis_kelamin', '$tanggal_lahir', '$alamat', '$kelas_id')";
21
22 if (mysqli_query(mysql: $conn, query: $insert_query)) {
23     header(header: "Location: ./");
24 } else {
25     header(header: "Location: ./");
26 }
27
28 exit();|
```

Membuat Halaman Edit Data Siswa

Pertama-tama, pastikan file `edit_siswa.php` ada di folder `views/siswa`. File ini digunakan untuk menampilkan form edit data siswa berdasarkan ID yang dipilih.

Selanjutnya, tambahkan kode berikut di awal file untuk memulai session, mengimpor koneksi database, dan fungsi tambahan:

```
session_start();
require_once __DIR__ . '/../../includes/db.php';
require_once __DIR__ . '/../../includes/function.php';
```

- `session_start();` digunakan untuk memulai sesi PHP, memungkinkan Anda mengakses data sesi seperti informasi pengguna.
- `require_once` mengimpor file `db.php` dan `function.php` untuk koneksi database dan fungsi tambahan.

Setelah itu, buat kode untuk membaca ID siswa dari URL dan mengambil data siswa berdasarkan ID tersebut:

```
$id = isset($_GET['id']) ? (int)$_GET['id'] : 0;

$siswa_query = "SELECT * FROM tb_siswa WHERE id = $id";
$siswa = $conn->query($siswa_query)->fetch_assoc();

if (!$siswa) {
    echo "<script>alert('Data siswa tidak ditemukan.');" . window.location.href
= 'index.php';</script>";
    exit;
}
```

Query yang digunakan

```
SELECT * FROM tb_siswa WHERE id = $id
```

Penjelasan: Query ini digunakan untuk mengambil semua data dari tabel `tb_siswa` berdasarkan `id` siswa yang diterima melalui URL parameter `$_GET['id']`. Nilai `id` dipastikan sebagai integer dengan menggunakan ternary operator. Jika data siswa ditemukan, hasil query disimpan dalam variabel `$siswa`. Jika tidak ditemukan, pengguna diberi peringatan melalui alert dan diarahkan kembali ke halaman `index.php`.

Penjelasan

- `$_GET['id']` membaca parameter `id` dari URL.
- Query `SELECT * FROM tb_siswa WHERE id = $id` mengambil data siswa dari database berdasarkan ID.
- Jika data siswa tidak ditemukan, kode akan menampilkan alert dan mengarahkan kembali ke halaman utama.

Kemudian, tambahkan logika untuk mendapatkan data kelas yang hanya diampu oleh wali kelas:

```
if ($role === "walikelas") {  
    $kelas_query = "SELECT id_kelas FROM tb_kelas WHERE walikelas_id =  
{$user['id_walikelas']}";  
}  
  
$kelas_data = fetchData($conn, $kelas_query);
```

Query yang digunakan

```
SELECT id_kelas FROM tb_kelas WHERE walikelas_id = {$user['id_walikelas']}
```

Penjelasan: Query ini digunakan untuk mengambil `id_kelas` dari tabel `tb_kelas` yang memiliki `walikelas_id` sesuai dengan ID wali kelas yang saat ini sedang login. Kondisi `WHERE walikelas_id = {$user['id_walikelas']}` memastikan hanya kelas yang diajar oleh wali kelas yang sesuai yang diambil. Fungsi `fetchData()` kemudian digunakan untuk mengeksekusi query ini dan mengambil hasilnya, yang disimpan dalam variabel `$kelas_data`.

Penjelasan :

- Variabel `$role` diperoleh dari data sesi untuk menentukan apakah pengguna adalah wali kelas.
- Jika `role` adalah wali kelas, query hanya mengambil kelas yang diampu pengguna tersebut.

```
edit_siswa.php views\siswa\edit_siswa.php
1 <?php
2 session_start();
3
4 require_once __DIR__ . '/../../includes/db.php';
5 require_once __DIR__ . '/../../includes/function.php';
6
7 $user = $_SESSION['user'];
8 $role = $user['role'];
9
10 // Ambil ID siswa dari URL
11 $id = isset($_GET['id']) ? (int) $_GET['id'] : 0;
12
13 // Ambil data siswa berdasarkan ID
14 $siswa_query = "SELECT * FROM tb_siswa WHERE id = $id";
15 $siswa = $conn->query(query: $siswa_query)->fetch_assoc();
16
17 if (!$siswa) {
18     echo "<script>alert('Data siswa tidak ditemukan.');" . window.location.href = 'index.php';</script>";
19     exit;
20 }
21
22 // Ambil data kelas untuk dropdown
23 if ($role === "admin") {
24     $kelas_query = "SELECT id_kelas, nama_kelas FROM tb_kelas";
25 } elseif ($role === "walikelas") {
26     $kelas_query = "SELECT id_kelas FROM tb_kelas WHERE walikelas_id = {$user['id_walikelas']}";
27 }
28 $kelas_data = fetchData(conn: $conn, query: $kelas_query);
29 ?>
```

Kemudian, buat struktur HTML untuk halaman.

```
<!DOCTYPE html>
<html lang="id">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Tambah Siswa</title>
        <link rel="stylesheet" href="/assets/css/styles.css" />
    </head>
    <body></body>
</html>
```

Penjelasan:

- `<!DOCTYPE html>` menentukan jenis dokumen sebagai HTML5.
 - `<link>` menautkan file CSS untuk mempercantik tampilan.
-

Selanjutnya, buat form HTML yang menampilkan data siswa untuk diedit:

```
<div class="wrapper">
    <!-- Sidebar -->
```

```
<?php include($_SERVER['DOCUMENT_ROOT'] .  
'/sikasus/views/component/sidebar.php'); ?>  
  
<!-- Main Content -->  
<div class="main-content">  
    <header>  
        <h1>Edit Data Siswa</h1>  
    </header>  
    <div class="container">  
        <form action="proses_edit_siswa.php" method="post">  
            <input type="hidden" name="id" value="<?= $siswa['id'] ?>">  
  
            <div>  
                <label for="nama_lengkap">Nama Lengkap:</label>  
                <input type="text" id="nama_lengkap" name="nama_lengkap"  
                    value="<?= htmlspecialchars($siswa['nama_lengkap']) ?>"  
                >  
                <div>  
                    <label for="nisn">NISN:</label>  
                    <input type="text" id="nisn" name="nisn" value="<?=  
                        htmlspecialchars($siswa['nisn']) ?>"  
                    >  
                    <div>  
                        <label for="jenis_kelamin">Jenis Kelamin:</label>  
                        <select id="jenis_kelamin" name="jenis_kelamin"  
                            required>  
                            <option value="Laki-laki" <?=  
                                $siswa['jenis_kelamin'] === "Laki-laki" ? "selected" : "" ?>>  
                                Laki-laki</option>  
                            <option value="Perempuan" <?=  
                                $siswa['jenis_kelamin'] === "Perempuan" ? "selected" : "" ?>>  
                                Perempuan</option>  
                        </select>  
                    </div>  
  
                    <div>  
                        <label for="tanggal_lahir">Tanggal Lahir:</label>  
                        <input type="date" id="tanggal_lahir"  
                            name="tanggal_lahir"  
                            value="<?= htmlspecialchars($siswa['tanggal_lahir']) ?>"  
                        >  
                        <div>  
                            <label for="alamat">Alamat:</label>  
                            <textarea id="alamat" name="alamat"  
                                value="<?= htmlspecialchars($siswa['alamat']) ?>">  
                        </div>  
                    </div>  
                </div>  
            </div>  
        </form>  
    </div>  
</div>
```

```

    required><?= htmlspecialchars($siswa['alamat']) ?>
</textarea>
</div>

<?php if ($role === "admin"): ?>
<div>
<label for="kelas_id">Kelas:</label>
<select id="kelas_id" name="kelas_id" required>
<?php foreach ($kelas_data as $kelas): ?>
    <option value="<?= $kelas['id_kelas'] ?>" <?
= $siswa['kelas_id'] == $kelas['id_kelas'] ? "selected" : "" ?>>
        <?=
htmlspecialchars($kelas['nama_kelas']) ?>
    </option>
    <?php endforeach; ?>
</select>
</div>
<?php else: ?>
    <input type="hidden" name="kelas_id" value="<?=
$siswa['kelas_id'] ?>">
    <?php endif; ?>

    <button type="submit">Simpan</button>
    <a href="index.php" class="button">Kembali</a>
</form>
</div>
</div>
</div>

```

Penjelasan

- Form menggunakan method POST dan action menuju proses_edit_siswa.php .
- Field seperti nama_lengkap , nisn , dan lainnya diisi dengan data siswa yang diambil dari database.
- Dropdown hanya ditampilkan untuk admin, sedangkan wali kelas menggunakan nilai tersembunyi (hidden).

Pertama, pastikan file proses_edit_siswa.php ada di folder views/siswa . File ini akan memproses data dari form edit.

Tambahkan kode berikut di awal file untuk memvalidasi bahwa request berasal dari form:

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $id = (int)$_POST['id'];
    $nama_lengkap = mysqli_real_escape_string($conn,
$_POST['nama_lengkap']);

```

```

$nisn = mysqli_real_escape_string($conn, $_POST['nisn']);
$jenis_kelamin = mysqli_real_escape_string($conn,
$_POST['jenis_kelamin']);
$tanggal_lahir = mysqli_real_escape_string($conn,
$_POST['tanggal_lahir']);
$alamat = mysqli_real_escape_string($conn, $_POST['alamat']);
$kelas_id = (int)$_POST['kelas_id'];
}

```

Penjelasan:

- Validasi method POST memastikan hanya form yang dapat mengakses file ini.
- Fungsi `mysqli_real_escape_string` mencegah serangan SQL Injection.

Setelah itu, buat query untuk memperbarui data siswa:

```

$update_query = "
UPDATE tb_siswa
SET
    nama_lengkap = '$nama_lengkap',
    nisn = '$nisn',
    jenis_kelamin = '$jenis_kelamin',
    tanggal_lahir = '$tanggal_lahir',
    alamat = '$alamat',
    kelas_id = $kelas_id
WHERE id = $id
";
if (mysqli_query($conn, $update_query)) {
    echo "<script>alert('Data siswa berhasil diupdate.');" .
    "window.location.href = 'index.php';</script>";
} else {
    echo "<script>alert('Terjadi kesalahan saat mengupdate data.');" .
    "window.history.back();</script>";
}

```

Query yang digunakan

```

UPDATE tb_siswa
SET
    nama_lengkap = '$nama_lengkap',
    nisn = '$nisn',
    jenis_kelamin = '$jenis_kelamin',
    tanggal_lahir = '$tanggal_lahir',
    alamat = '$alamat',
    kelas_id = $kelas_id
WHERE id = $id

```

Penjelasan: Query ini digunakan untuk memperbarui data siswa pada tabel `tb_siswa` berdasarkan `id`. Kolom yang diperbarui meliputi `nama_lengkap`, `nism`, `jenis_kelamin`, `tanggal_lahir`, `alamat`, dan `kelas_id`, dengan nilai yang diberikan melalui variabel PHP. Kondisi `WHERE id = $id` memastikan hanya data siswa dengan `id` yang sesuai yang akan diperbarui. Setelah query dijalankan menggunakan `mysqli_query()`, jika berhasil, pengguna akan diberi notifikasi bahwa data berhasil diupdate dan diarahkan ke halaman utama. Jika gagal, pengguna akan menerima pesan kesalahan dan diarahkan kembali ke halaman sebelumnya.

Penjelasan :

- Query `UPDATE` digunakan untuk memperbarui data siswa berdasarkan ID.
- Jika berhasil, pesan sukses akan muncul dan pengguna diarahkan kembali ke halaman utama.

```
hapus_siswa.php views\siswa\hapus_siswa.php
1 <?php
2 session_start();
3
4 require_once __DIR__ . '/../../includes/db.php';
5 require_once __DIR__ . '/../../includes/function.php';
6
7 // Validasi parameter ID
8 if (!isset($_GET['id']) || !is_numeric(value: $_GET['id'])) {
9     jsAlert(message: 'ID siswa tidak ditemukan.', redirectUrl: 'index.php');
10    exit();
11 }
12
13 $id_siswa = intval(value: $_GET['id']);
14
15 // Query untuk menghapus siswa
16 $delete_query = "DELETE FROM tb_siswa WHERE id = $id_siswa";
17
18 // Eksekusi query dan redirect dengan notifikasi
19 if (mysqli_query(mysql: $conn, query: $delete_query)) {
20     jsAlert(message: 'Siswa berhasil dihapus.', redirectUrl: 'index.php');
21 } else {
22     jsAlert(message: 'Kesalahan saat menghapus siswa.', redirectUrl: 'index.php');
23 }
24
25 exit();
```

Membuat Halaman Hapus Data Siswa

Pertama-tama, buat file bernama `hapus_siswa.php` di dalam folder `views/siswa`.

Selanjutnya, tambahkan kode untuk membuka session di awal file. Hal ini bertujuan untuk memastikan pengguna yang mengakses halaman ini memiliki hak akses yang valid.

```
<?php
session_start();
```

Penjelasan:

- `session_start()` : Memulai sesi untuk melacak data pengguna seperti status login.
 - Pastikan file ini tidak dapat diakses langsung oleh pengguna yang tidak berwenang.
-

Kemudian, sertakan file pendukung yang diperlukan, seperti `db.php` untuk koneksi database dan `function.php` untuk fungsi tambahan.

```
require_once __DIR__ . '/../../includes/db.php';
require_once __DIR__ . '/../../includes/function.php';
```

Penjelasan:

- `db.php` : Mengelola koneksi ke database.
 - `function.php` : Berisi fungsi tambahan seperti `jsAlert` yang digunakan untuk menampilkan notifikasi.
-

Berikutnya, tambahkan kode untuk memvalidasi parameter `id` yang dikirim melalui URL.

```
if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
    jsAlert('ID siswa tidak ditemukan.', 'index.php');
    exit();
}
$id_siswa = intval($_GET['id']);
```

Penjelasan:

- `isset($_GET['id'])` : Memeriksa apakah parameter `id` ada dalam URL.
 - `is_numeric($_GET['id'])` : Memastikan bahwa nilai `id` adalah angka.
 - `intval($_GET['id'])` : Mengonversi nilai `id` menjadi bilangan bulat untuk mencegah serangan injeksi.
 - Jika parameter tidak valid, pengguna diarahkan kembali ke halaman utama dengan notifikasi kesalahan.
-

Setelah itu, siapkan query SQL untuk menghapus data siswa berdasarkan ID yang diterima.

```
$delete_query = "DELETE FROM tb_siswa WHERE id = $id_siswa";
```

Query yang digunakan

```
DELETE FROM tb_siswa WHERE id = $id_siswa
```

Penjelasan: Query ini menghapus data siswa dari tabel `tb_siswa` berdasarkan `id` yang sesuai dengan `$id_siswa`.

Penjelasan:

Query `DELETE` ini akan menghapus baris data dari tabel `tb_siswa` di mana kolom `id` sama dengan `$id_siswa`.

Selanjutnya, eksekusi query untuk menghapus data dan tampilkan notifikasi berdasarkan hasil eksekusi tersebut.

```
if (mysqli_query($conn, $delete_query)) {
    jsAlert('Siswa berhasil dihapus.', 'index.php');
} else {
    jsAlert('Kesalahan saat menghapus siswa.', 'index.php');
}
exit();
```

Penjelasan:

- `mysqli_query($conn, $delete_query)` : Menjalankan query ke database.
 - Jika query berhasil, fungsi `jsAlert` akan menampilkan pesan "**Siswa berhasil dihapus**" dan mengarahkan pengguna kembali ke halaman utama.
 - Jika terjadi kesalahan, pesan "**Kesalahan saat menghapus siswa**" akan muncul.
-

```
hapus_siswa.php views\siswa\hapus_siswa.php

1 <?php
2 session_start();
3
4 require_once __DIR__ . '/../../includes/db.php';
5 require_once __DIR__ . '/../../includes/function.php';
6
7 // Validasi parameter ID
8 if (!isset($_GET['id']) || !is_numeric(value: $_GET['id'])) {
9     jsAlert(message: 'ID siswa tidak ditemukan.', redirectUrl: 'index.php');
10    exit();
11 }
12
13 $id_siswa = intval(value: $_GET['id']);
14
15 // Query untuk menghapus siswa
16 $delete_query = "DELETE FROM tb_siswa WHERE id = $id_siswa";
17
18 // Eksekusi query dan redirect dengan notifikasi
19 if (mysqli_query(mysql: $conn, query: $delete_query)) {
20     jsAlert(message: 'Siswa berhasil dihapus.', redirectUrl: 'index.php');
21 } else {
22     jsAlert(message: 'Kesalahan saat menghapus siswa.', redirectUrl: 'index.php');
23 }
24
25 exit();
```

Wali Kelas - CRUD Kasus

Menampilkan Daftar Kasus

Sebelumnya, dalam file `index.php` yang sudah ada, data kasus ditampilkan untuk seluruh siswa tanpa batasan per kelas atau wali kelas. Ini berarti, semua pengguna akan melihat semua data kasus tanpa pembatasan. Namun, untuk aplikasi yang lebih relevan dan sesuai dengan kebutuhan **wali kelas**, kita perlu menyesuaikan query SQL untuk **memfilter data** agar hanya kasus-kasus yang terkait dengan siswa di kelas yang dipimpin oleh wali kelas yang sedang login.

Langkah pertama yang perlu dilakukan adalah menambahkan variabel `$walikelas_id` yang akan menyimpan ID wali kelas dari session. Hal ini memungkinkan kita untuk memfilter data yang hanya terkait dengan kelas yang diawasi oleh wali kelas tersebut.

```
$walikelas_id = $user['id_walikelas'] ?? null;
```

Penjelasan:

- `$user['id_walikelas'] ?? null` : Menggunakan operator **null coalescing** untuk mengambil ID wali kelas jika ada, atau memberikan nilai `null` jika tidak tersedia.
- Variabel ini penting untuk membatasi data kasus yang akan ditampilkan hanya sesuai dengan kelas wali kelas yang sedang login.

Selanjutnya, tambahkan logika untuk menyesuaikan query SQL berdasarkan peran pengguna. Jika pengguna memiliki peran `walikelas`, tambahkan klausa `WHERE` untuk membatasi data sesuai ID wali kelas.

```
if ($role === 'walikelas') {
    $kasus_query = "SELECT k.id_kasus, k.siswa_id, k.deskripsi_kasus,
    k.tanggal_kasus, s.nama_lengkap
        FROM tb_kasus k
        LEFT JOIN tb_siswa s ON k.siswa_id = s.id
        JOIN tb_kelas kl ON s.kelas_id = kl.id_kelas
        WHERE kl.walikelas_id = $walikelas_id
        ORDER BY k.tanggal_kasus DESC";
}
```



The screenshot shows the VS Code interface. On the left is the Explorer sidebar with project files like .vscode, admin-template-sb, assets, includes, views (with subfolders component and kasus), and a file named index.php which is currently selected. The main area is the code editor displaying the PHP code. The code includes logic to check if the user's role is 'walikelas' and then constructs a query to select data from tb_kasus, tb_siswa, and tb_kelas tables, applying a WHERE clause to filter by walikelas_id and ordering by tanggal_kasus in descending order.

```
<?php
session_start();
require_once __DIR__ . '/../../../../includes/db.php';
require_once __DIR__ . '/../../../../includes/function.php';
$user = $_SESSION['user'];
$role = $user['role'];
$walikelas_id = $user['id_walikelas'] ?? null;
if ($role === 'walikelas') {
    $kasus_query = "SELECT k.id_kasus, k.siswa_id, k.deskripsi_kasus, k.tanggal_kasus, s.nama_lengkap
        FROM tb_kasus k
        LEFT JOIN tb_siswa s ON k.siswa_id = s.id
        JOIN tb_kelas kl ON s.kelas_id = kl.id_kelas
        WHERE kl.walikelas_id = $walikelas_id
        ORDER BY k.tanggal_kasus DESC";
}
```

Query yang digunakan

```
SELECT k.id_kasus, k.siswa_id, k.deskripsi_kasus, k.tanggal_kasus,
s.nama_lengkap
FROM tb_kasus k
LEFT JOIN tb_siswa s ON k.siswa_id = s.id
JOIN tb_kelas kl ON s.kelas_id = kl.id_kelas
WHERE kl.walikelas_id = $walikelas_id
ORDER BY k.tanggal_kasus DESC
```

Penjelasan: Query ini digunakan untuk mengambil data kasus yang terkait dengan siswa dalam kelas yang diajar oleh wali kelas tertentu. Tabel `tb_kasus` digabungkan dengan `tb_siswa` menggunakan `LEFT JOIN` berdasarkan `siswa_id`, dan tabel `tb_siswa` digabungkan dengan `tb_kelas` menggunakan `JOIN` berdasarkan `kelas_id`. Kondisi `WHERE kl.walikelas_id = $walikelas_id` memastikan hanya kasus dari kelas yang diajar oleh wali kelas yang sesuai yang diambil. Hasilnya diurutkan berdasarkan `tanggal_kasus` secara menurun (`DESC`).

Penjelasan:

- `if ($role === 'walikelas')` : Memastikan bahwa logika khusus ini hanya berjalan untuk pengguna dengan peran wali kelas.
 - `JOIN tb_kelas kl ON s.kelas_id = kl.id_kelas` : Menghubungkan tabel siswa dengan tabel kelas untuk mendapatkan informasi wali kelas.
 - `WHERE kl.walikelas_id = $walikelas_id` : Membatasi data hanya pada kelas yang diawasi oleh wali kelas yang sedang login.
-

Kemudian, tambahkan kode untuk mengeksekusi query SQL dengan fungsi `fetchData`, tanpa mengubah kode sebelumnya untuk eksekusi query.

```
$kasus_result = fetchData($conn, $kasus_query);

0 $walikelas_id = $user['id_walikelas'] ?? null;
1
2 if ($role === 'walikelas') {
3     $kasus_query = "SELECT k.id_kasus, k.siswa_id, k.deskripsi_kasus, k.tanggal_kasus, s.nama_lengkap
4         FROM tb_kasus k
5             LEFT JOIN tb_siswa s ON k.siswa_id = s.id
6             JOIN tb_kelas kl ON s.kelas_id = kl.id_kelas
7             WHERE kl.walikelas_id = $walikelas_id
8             ORDER BY k.tanggal_kasus DESC";
9 }
0
1 // Eksekusi query dan ambil data
2 $kasus_result = fetchData(conn: $conn, query: $kasus_query);
3 ?>
```

Penjelasan:

- `fetchData` : Fungsi ini akan menjalankan query ke database dan mengembalikan hasilnya dalam bentuk array. Pastikan fungsi ini sudah didefinisikan di file `function.php`.
-

Halaman Tambah Data Kasus

Pada bagian ini, kita melakukan perubahan untuk memastikan bahwa hanya wali kelas yang dapat melihat siswa-siswa dalam kelas yang mereka pimpin, bukan semua siswa di seluruh sekolah.

Langkah pertama yang perlu dilakukan adalah mengambil data pengguna yang sedang login melalui session. Data ini sangat penting untuk menentukan peran pengguna (role), seperti apakah dia seorang admin, wali kelas, atau siswa.

```
$user = $_SESSION['user'];
$role = $user['role'];
```

The screenshot shows the VS Code interface. On the left is the Explorer sidebar with icons for files, folders, and search. The main area shows the file structure of 'SIKAS...' with a tree view of '.vscode', 'admin-template-sb', 'assets', 'includes', 'views' (which contains 'component'), and a .gitignore file. To the right is the code editor window displaying 'tambah_kasus.php' from the 'views\kasus\tambah_kasus.php' path. The code is as follows:

```
<?php
session_start();
// Menghubungkan ke database
require_once __DIR__ . '/../../../../includes/db.php';
$user = $_SESSION['user'];
$role = $user['role'];
```

Penjelasan:

- `$_SESSION['user']` menyimpan data pengguna yang login. Dari sini kita bisa mengambil informasi terkait pengguna.
- `$role` digunakan untuk mengetahui peran pengguna tersebut. Misalnya, apakah pengguna tersebut adalah wali kelas atau admin.

Selanjutnya, kita akan mengecek apakah pengguna yang login adalah **wali kelas**. Jika iya, maka hanya siswa di kelas yang dipimpin oleh wali kelas tersebut yang akan ditampilkan dalam form.

```
$walikelas_id = $user['id_walikelas'] ?? null;

// Query berdasarkan role user
if ($role === "admin") {
$query = "SELECT tb_siswa.id, tb_siswa.nama_lengkap
FROM tb_siswa
JOIN tb_kelas ON tb_siswa.kelas_id = tb_kelas.id_kelas";
} elseif ($role === "walikelas") {
$query = "SELECT tb_siswa.id, tb_siswa.nama_lengkap
FROM tb_siswa
JOIN tb_kelas ON tb_siswa.kelas_id = tb_kelas.id_kelas
WHERE tb_kelas.walikelas_id = $walikelas_id";
}
```

```
tambah_kasus.php views\kasus\tambah_kasus.php
1 <?php
2 session_start();
3
4 // Menghubungkan ke database
5 require_once __DIR__ . '/../../includes/db.php';
6
7 $user = $_SESSION['user'];
8 $role = $user['role'];
9
10 $walikelas_id = $user['id_walikelas'] ?? null;
11
12 // Query berdasarkan role user
13 if ($role === "admin") {
14     $query = "SELECT tb_siswa.id, tb_siswa.nama_lengkap
15             FROM tb_siswa
16             JOIN tb_kelas ON tb_siswa.kelas_id = tb_kelas.id_kelas";
17 } elseif ($role === "walikelas") {
18     $query = "SELECT tb_siswa.id, tb_siswa.nama_lengkap
19             FROM tb_siswa
20             JOIN tb_kelas ON tb_siswa.kelas_id = tb_kelas.id_kelas
21             WHERE tb_kelas.walikelas_id = $walikelas_id";
22 }
```

Query yang digunakan

```
SELECT tb_siswa.id, tb_siswa.nama_lengkap
FROM tb_siswa
JOIN tb_kelas ON tb_siswa.kelas_id = tb_kelas.id_kelas
```

Penjelasan: Query ini digunakan untuk mengambil `id` dan `nama_lengkap` siswa dari tabel `tb_siswa`, dengan melakukan `JOIN` pada tabel `tb_kelas` berdasarkan `kelas_id`. Ini berlaku untuk pengguna dengan role "admin" yang dapat mengakses data siswa dari seluruh kelas.

```
SELECT tb_siswa.id, tb_siswa.nama_lengkap
FROM tb_siswa
JOIN tb_kelas ON tb_siswa.kelas_id = tb_kelas.id_kelas
WHERE tb_kelas.walikelas_id = $walikelas_id
```

Penjelasan: Query ini digunakan untuk mengambil `id` dan `nama_lengkap` siswa dari tabel `tb_siswa`, dengan `JOIN` pada tabel `tb_kelas` berdasarkan `kelas_id`, hanya untuk kelas yang diajar oleh wali kelas yang memiliki `walikelas_id` sesuai dengan `$walikelas_id`. Ini berlaku untuk pengguna dengan role "walikelas" yang hanya dapat mengakses data siswa dalam kelas yang diajarnya.

Penjelasan:

- **\$walikelas_id** berfungsi untuk mengambil ID wali kelas yang ada di data pengguna.

- Jika **\$role** adalah `walikelas` , maka query akan mencari siswa yang ada di kelas yang dipimpin oleh wali kelas tersebut. Query ini akan menggabungkan tabel **tb_siswa** dan **tb_kelas** berdasarkan ID wali kelas.
-

Selanjutnya, kita akan menjalankan query ini untuk mendapatkan daftar siswa yang sesuai.

```
1 // $walikelas_id = user['id_walikelas'] ?? null;
2
3 // Query berdasarkan role user
4 if ($role === "admin") {
5     $query = "SELECT tb_siswa.id, tb_siswa.nama_lengkap
6             FROM tb_siswa
7             JOIN tb_kelas ON tb_siswa.kelas_id = tb_kelas.id_kelas";
8 } elseif ($role === "walikelas") {
9     $query = "SELECT tb_siswa.id, tb_siswa.nama_lengkap
10            FROM tb_siswa
11            JOIN tb_kelas ON tb_siswa.kelas_id = tb_kelas.id_kelas
12            WHERE tb_kelas.walikelas_id = $walikelas_id";
13 }
14
15 $result = mysqli_query(mysql: $conn, query: $query);
16 ?>
17
18 <!DOCTYPE html>
19 <html lang="id">
20
```

Penjelasan:

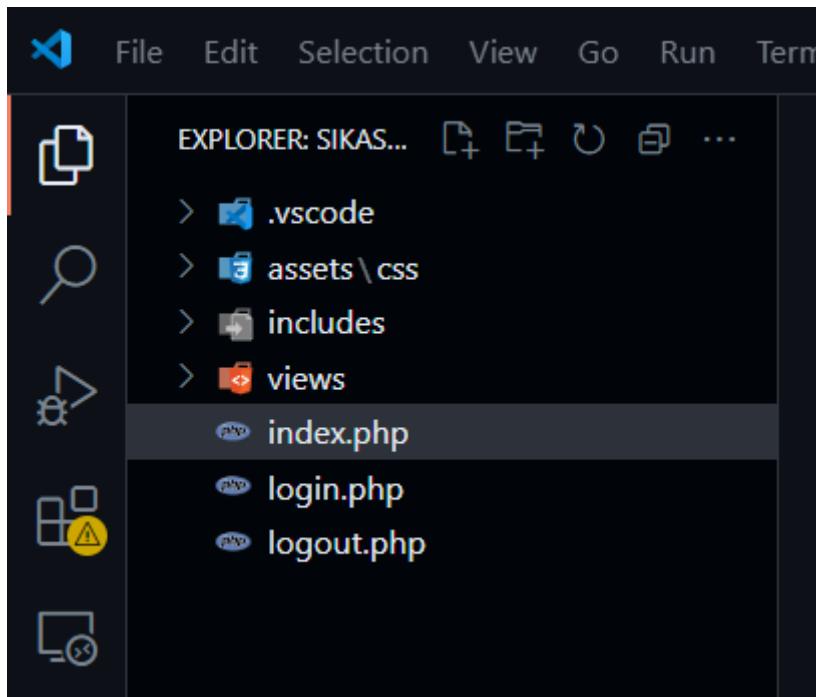
- Fungsi **mysqli_query** digunakan untuk mengeksekusi query yang telah kita buat.
- **\$result** berisi hasil query yang nantinya akan kita tampilkan dalam dropdown di form.

Kesimpulan

Wali kelas diberikan hak untuk melakukan operasi **CRUD** (Create, Read, Update, Delete) terhadap data siswa dan kasus yang ada di kelasnya. Ini memberikan wali kelas wewenang untuk mengelola data dengan lebih mudah dan efisien sesuai dengan tanggung jawabnya sebagai pengelola kelas.

Tampilan Beranda Kasus Siswa

Struktur Folder

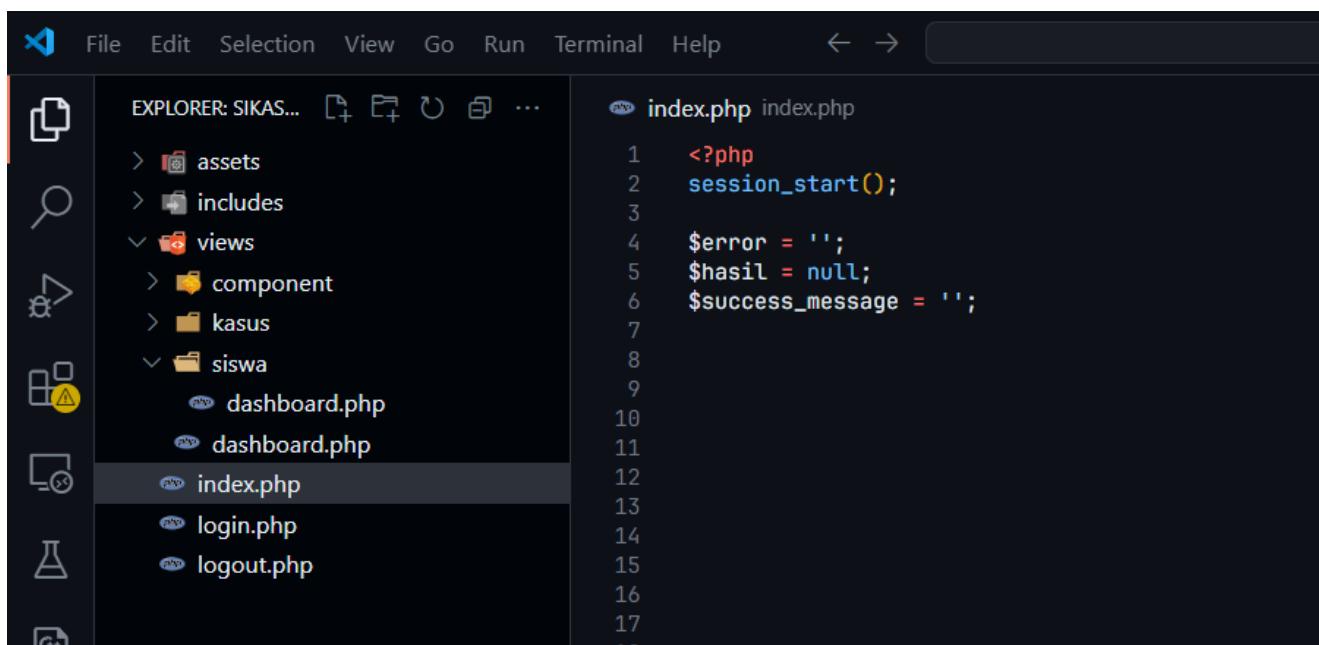


Pertama-tama, buat file bernama `index.php` di root proyek Anda jika belum ada. File ini akan menjadi halaman utama (*homepage*) yang digunakan untuk menampilkan data kasus siswa.

Selanjutnya, tuliskan kode berikut pada bagian awal file `index.php` untuk memulai sesi pengguna dan menginisialisasi variabel penting seperti `$error` , `$hasil` , dan `$success_message` .

```
<?php
session_start();

$error = '';
$hasil = null;
$success_message = '';
```



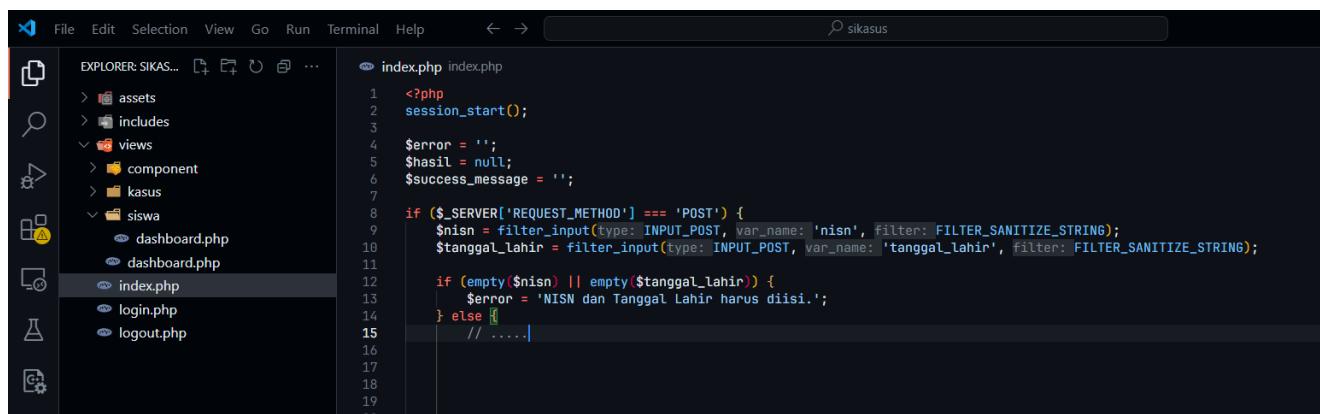
Penjelasan Kode:

- `session_start();` digunakan untuk memulai sesi PHP agar dapat melacak data pengguna selama aplikasi berjalan.
- Variabel `$error` digunakan untuk menyimpan pesan kesalahan.
- Variabel `$hasil` digunakan untuk menyimpan hasil query dari database.
- Variabel `$success_message` digunakan untuk menyimpan pesan sukses jika data berhasil ditemukan.

Setelah itu, tambahkan logika untuk menangani pengiriman data dari formulir dengan metode `POST`. Pada bagian ini, data *NISN* dan *tanggal lahir* akan diterima dan divalidasi.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $nisn = htmlspecialchars($nisn, ENT_QUOTES, 'UTF-8');
    $tanggal_lahir = htmlspecialchars($tanggal_lahir, ENT_QUOTES, 'UTF-8');

    if (empty($nisn) || empty($tanggal_lahir)) {
        $error = 'NISN dan Tanggal Lahir harus diisi.';
    } else {
        // ....
```



The screenshot shows a code editor interface with a sidebar labeled "EXPLORER: SIKAS..." containing files like assets, includes, views, component, kasus, siswa, dashboard.php, and index.php. The main area displays the following PHP code:

```
<?php
session_start();
$error = '';
$hasil = null;
$success_message = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $nisn = filter_input(type: INPUT_POST, var_name: 'nisn', filter: FILTER_SANITIZE_STRING);
    $tanggal_lahir = filter_input(type: INPUT_POST, var_name: 'tanggal_lahir', filter: FILTER_SANITIZE_STRING);

    if (empty($nisn) || empty($tanggal_lahir)) {
        $error = 'NISN dan Tanggal Lahir harus diisi.';
    } else {
        // ....|
```

Penjelasan Kode:

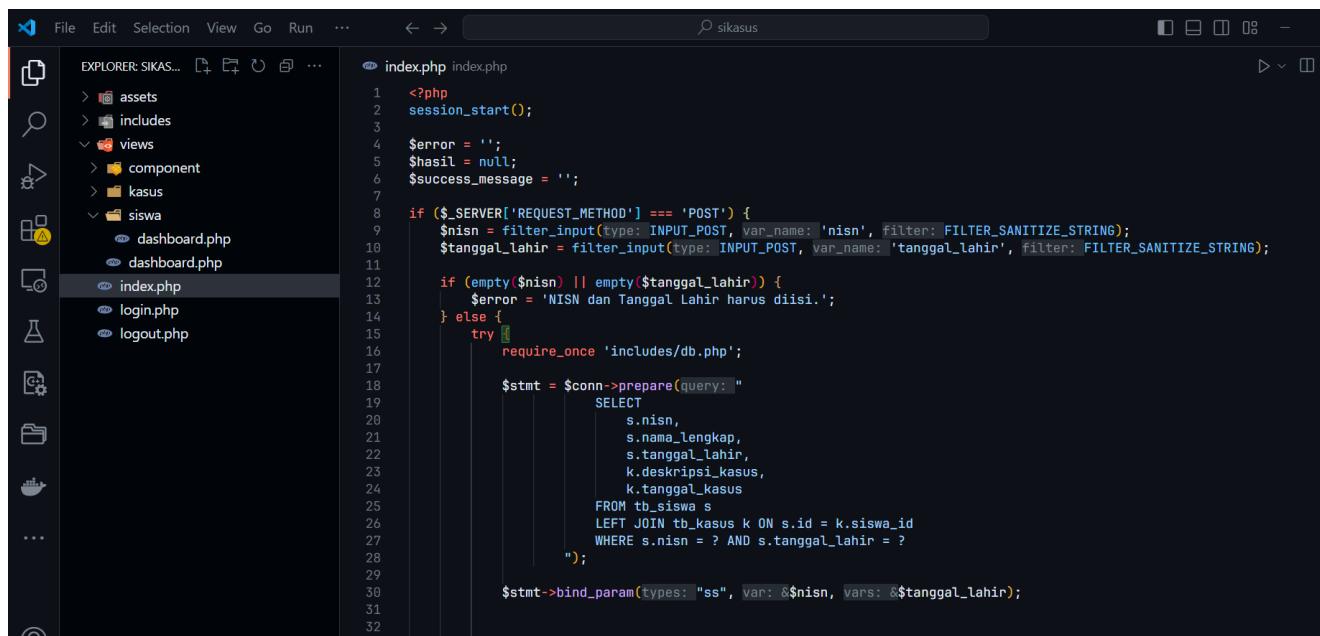
- `$_SERVER['REQUEST_METHOD'] === 'POST'` memastikan logika hanya dijalankan saat formulir dikirimkan.
- `filter_input()` digunakan untuk menyaring data input agar aman dari karakter berbahaya.
- Kondisi `if (empty(...))` mengecek apakah *NISN* atau *tanggal lahir* kosong. Jika iya, maka pesan kesalahan disimpan di `$error`.

Selanjutnya, jika data tidak kosong, lakukan proses koneksi ke database dan jalankan query untuk mencari data siswa dan kasusnya. Jangan lupa, pastikan file koneksi database (db.php) sudah ada.

```
try {
    require_once 'includes/db.php';

    $stmt = $conn->prepare("
        SELECT
            s.nisn,
            s.nama_lengkap,
            s.tanggal_lahir,
            k.deskripsi_kasus,
            k.tanggal_kasus
        FROM tb_siswa s
        LEFT JOIN tb_kasus k ON s.id = k.siswa_id
        WHERE s.nisn = ? AND s.tanggal_lahir = ?
    ");
}

$stmt->bind_param("ss", $nisn, $tanggal_lahir);
```



The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows the project structure with files like index.php, dashboard.php, login.php, and logout.php.
- Code Editor:** Displays the PHP code for index.php. The SQL query is highlighted in yellow.
- Search Bar:** Contains the text "siklus".
- Toolbar:** Includes standard file operations like File, Edit, Selection, View, Go, Run, and a search icon.

Query yang digunakan

```
SELECT
    s.nisn,
    s.nama_lengkap,
    s.tanggal_lahir,
    k.deskripsi_kasus,
    k.tanggal_kasus
FROM tb_siswa s
```

```
LEFT JOIN tb_kasus k ON s.id = k.siswa_id  
WHERE s.nisn = ? AND s.tanggal_lahir = ?
```

Penjelasan: Query ini digunakan untuk mengambil data siswa dan kasus yang terkait dengan siswa berdasarkan `nisn` dan `tanggal_lahir`. Tabel `tb_siswa` digabungkan dengan `tb_kasus` menggunakan `LEFT JOIN` berdasarkan `siswa_id`. Kondisi `WHERE s.nisn = ? AND s.tanggal_lahir = ?` memastikan bahwa hanya data siswa yang memiliki kecocokan `nisn` dan `tanggal_lahir` yang diberikan yang akan diambil. Parameter `nisn` dan `tanggal_lahir` diikat menggunakan `bind_param` dengan tipe data string (`ss`).

Penjelasan Kode:

- `require_once 'includes/db.php';` memuat file koneksi database.
- Query SQL di atas menggunakan `LEFT JOIN` untuk mengambil data siswa beserta kasusnya.
- `bind_param("ss", $nisn, $tanggal_lahir)` digunakan untuk mengikat parameter *NISN* dan *tanggal lahir* ke dalam query dengan format string (`s`).

Lalu, jalankan query dengan metode `execute()` dan periksa apakah data ditemukan.

```
if ($stmt->execute()) {  
    $result = $stmt->get_result();  
    if ($result->num_rows > 0) {  
        $hasil = $result->fetch_all(MYSQLI_ASSOC);  
        $success_message = 'Data ditemukan!';  
    } else {  
        $error = 'Data siswa tidak ditemukan.';  
    }  
} else {  
    throw new Exception('Error executing query');  
}  
} catch (Exception $e) {  
    $error .= "Error : $e";  
}  
}
```

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons for file operations like Open, Save, Find, and Refresh. The main area shows a file named 'index.php' with the following content:

```
8 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
14 } else {
15     try {
16         $nisn = $_POST['nisn'];
17         $tanggal_lahir = $_POST['tanggal_lahir'];
18
19         $query = "SELECT k.nisn, k.nama_lengkap, k.tanggal_lahir, k.deskripsi_kasus, k.tanggal_kasus
20         FROM tb_siswa s
21         LEFT JOIN tb_kasus k ON s.id = k.siswa_id
22         WHERE s.nisn = ? AND s.tanggal_lahir = ?";
23
24         $stmt = $pdo->prepare($query);
25         $stmt->bind_param("ss", $nisn, $tanggal_lahir);
26
27         if ($stmt->execute()) {
28             $result = $stmt->get_result();
29             if ($result->num_rows > 0) {
30                 $hasil = $result->fetch_all(mode: MYSQLI_ASSOC);
31                 $success_message = 'Data ditemukan!';
32             } else {
33                 $error = 'Data siswa tidak ditemukan.';
34             }
35         } else {
36             throw new Exception(message: 'Error executing query');
37         }
38     } catch (Exception $e) {
39         $error .= "Error : $e";
40     }
41 }
42
43 ?>
44
45 }
```

Penjelasan Kode:

- `execute()` menjalankan query yang sudah disiapkan.
- `get_result()` mengambil hasil dari query.
- `fetch_all(MYSQLI_ASSOC)` mengembalikan data dalam bentuk array asosiatif.
- Jika tidak ada hasil, pesan kesalahan akan ditampilkan.

Selanjutnya, tuliskan struktur HTML dasar sebagai berikut.

```
<!DOCTYPE html>
<html lang="id">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Kasus Siswa</title>
    </head>
</html>
```

Penjelasan:

- `<!DOCTYPE html>` digunakan untuk menentukan bahwa dokumen ini adalah dokumen HTML5.
- Tag `<html>` dimulai dengan atribut `lang="id"`, yang menunjukkan bahasa dokumen adalah bahasa Indonesia.
- `<meta charset="UTF-8">` memastikan karakter yang digunakan dalam halaman mendukung berbagai bahasa.

- <title> memberikan judul halaman, yang akan terlihat di tab browser.

Selanjutnya, tambahkan *style internal* untuk mempercantik tampilan.

```
<style>
    body {
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
        background-color: #f9f9f9;
        color: #333;
    }

    .container {
        max-width: 600px;
        margin: 20px auto;
        padding: 20px;
        background: #fff;
        border: 1px solid #ddd;
        border-radius: 4px;
    }

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

    .header h1 {
        font-size: 20px;
        margin-bottom: 10px;
    }

    .header p {
        font-size: 14px;
        color: #555;
    }

    .input-group {
        margin-bottom: 15px;
    }

    .input-group label {
        display: block;
        font-size: 14px;
        margin-bottom: 5px;
    }

    .input-group input {
        width: 100%;
```

```
padding: 8px;
border: 1px solid #ccc;
border-radius: 4px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 4px;
    font-size: 14px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

button:hover {
    background-color: #0056b3;
}

.message {
    padding: 10px;
    margin: 10px 0 20px;
    border: 1px solid;
    border-radius: 4px;
    font-size: 14px;
}

.message.error {
    background: #f8d7da;
    border-color: #f5c6cb;
    color: #721c24;
}

.message.success {
    background: #d4edda;
    border-color: #c3e6cb;
    color: #155724;
}

.result-card {
    border: 1px solid #ddd;
    padding: 15px;
    border-radius: 4px;
}

.result-card h2 {
    margin: 0 0 10px;
}
```

```

.result-card p {
    margin: 5px 0;
}

.auth-buttons {
    margin-top: 20px;
    text-align: center;
}

.auth-buttons a {
    text-decoration: none;
    color: #007bff;
    font-size: 14px;
    transition: text-decoration 0.3s ease;
}

.auth-buttons a:hover {
    text-decoration: underline;
}

.item-kasus {
    padding: 0.5rem 0;
}

</style>
</head>

```

The screenshot shows a code editor interface with the following details:

- File Explorer (Left):** Shows the project structure under "EXPLORER: SIKASUS". It includes:
 - assets
 - includes
 - views
 - component
 - kasus
 - siswa
 - dashboard.php
 - index.php
 - index.php
 - login.php
 - logout.php
- Code Editor (Right):** Displays the content of index.php. The code includes basic HTML structure, meta tags, and CSS styles for body, container, header, and input-group.
- Toolbar:** Standard file operations like File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar labeled "sikasus".
- Bottom Status Bar:** Shows icons for file operations like save, close, and zoom.

Penjelasan:

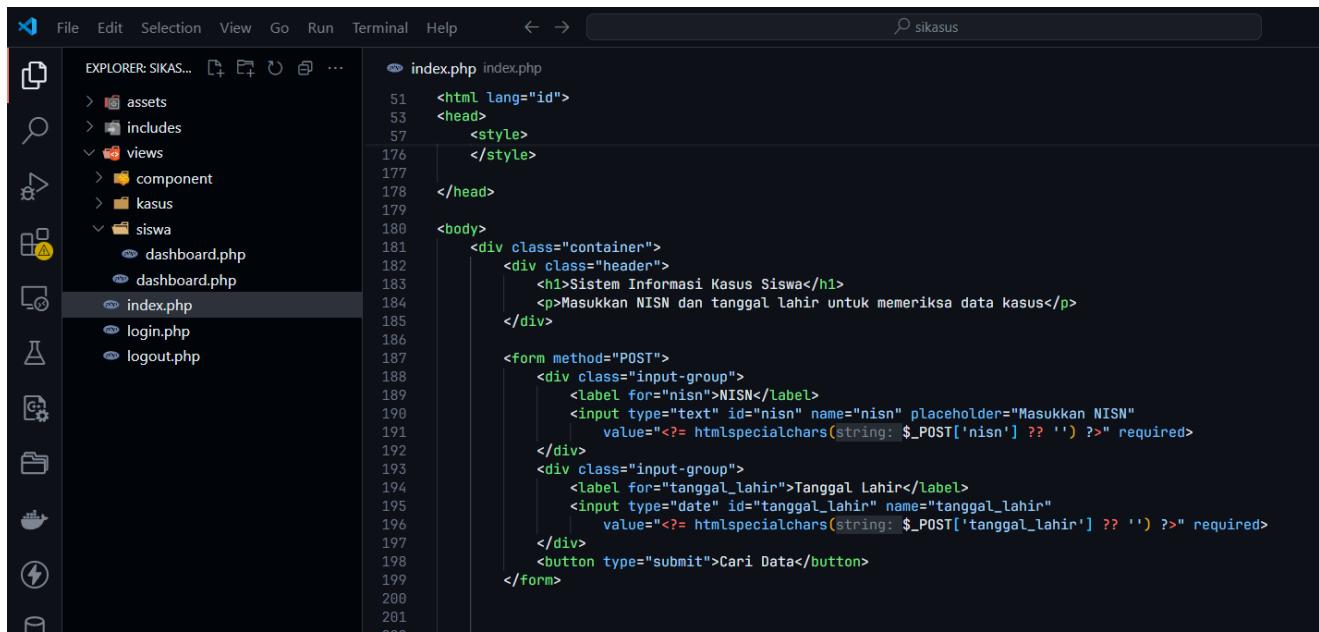
- body memberikan gaya dasar seperti font, margin, padding, dan warna latar belakang.

- `.container` digunakan untuk membatasi lebar konten dan memberikan efek `box shadow` agar tampilan lebih rapi.
- `.input-group`, `label`, dan `input` digunakan untuk mendefinisikan gaya formulir input.
- `.error` dan `.success` digunakan untuk menampilkan pesan kesalahan atau sukses dengan warna berbeda.

Setelah itu, buat struktur HTML untuk formulir input NISN dan tanggal lahir.

```
<body>
  <div class="container">
    <div class="header">
      <h1>Sistem Informasi Kasus Siswa</h1>
      <p>Masukkan NISN dan tanggal lahir untuk memeriksa data kasus</p>
    </div>

    <form method="POST">
      <div class="input-group">
        <label for="nisn">NISN</label>
        <input
          type="text"
          id="nisn"
          name="nisn"
          placeholder="Masukkan NISN"
          value=<?= htmlspecialchars($_POST['nisn'] ?? '') ?>
          required
        />
      </div>
      <div class="input-group">
        <label for="tanggal_lahir">Tanggal Lahir</label>
        <input
          type="date"
          id="tanggal_lahir"
          name="tanggal_lahir"
          value=<?= htmlspecialchars($_POST['tanggal_lahir'] ?? '') ?>
          required
        />
      </div>
      <button type="submit">Cari Data</button>
    </form>
  </div>
</body>
```



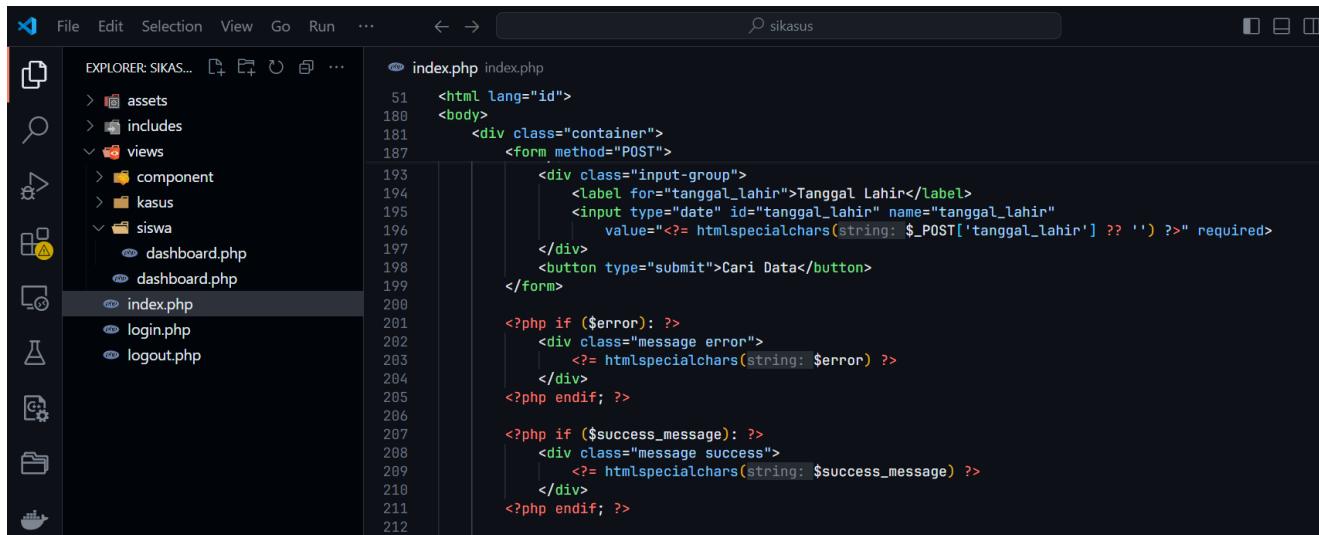
```
<?php if ($error): ?>
<div class="message error">
<?= htmlspecialchars($error) ?>
</div>
<?php endif; ?>

<?php if ($success_message): ?>
<div class="message success">
<?= htmlspecialchars($success_message) ?>
</div>
<?php endif; ?>
```

Selanjutnya, tambahkan kode untuk menampilkan pesan kesalahan (error) dan pesan berhasil.

```
<?php if ($error): ?>
<div class="message error">
<?= htmlspecialchars($error) ?>
</div>
<?php endif; ?>

<?php if ($success_message): ?>
<div class="message success">
<?= htmlspecialchars($success_message) ?>
</div>
<?php endif; ?>
```



```
<?php if ($error): ?>
<div class="message error">
| <?= htmlspecialchars(string: $error) ?>
</div>
<?php endif; ?>

<?php if ($success_message): ?>
<div class="message success">
| <?= htmlspecialchars(string: $success_message) ?>
</div>
<?php endif; ?>
```

Lalu, tampilkan hasil pencarian (jika ada) atau pesan kesalahan (jika tidak ada data ditemukan).

```
<?php if ($hasil): ?>
<div class="result-card">
    <h2>Informasi Kasus Siswa</h2>
    <p><strong>NISN:</strong> <?= htmlspecialchars($hasil[0]['nisn']) ?>
</p>
    <p><strong>Nama Lengkap:</strong> <?= htmlspecialchars($hasil[0]
['nama_lengkap']) ?></p>
    <p><strong>Tanggal Lahir:</strong> <?= date('d F Y',
strtotime($hasil[0]['tanggal_lahir'])) ?></p>

    <?php if (!empty(array_filter($hasil, fn($item) =>
$item['deskripsi_kasus']))): ?>
        <h3>Daftar Kasus:</h3>
        <ul>
            <?php foreach ($hasil as $kasus): ?>
                <?php if ($kasus['deskripsi_kasus']): ?>
                    <li class="item-kasus">
                        <strong>Kasus:</strong> <?=
htmlspecialchars($kasus['deskripsi_kasus']) ?><br>
                        <strong>Tanggal Kejadian:</strong> <?= date('d F
Y', strtotime($kasus['tanggal_kasus'])) ?>
                    </li>
                <?php endif; ?>
            <?php endforeach; ?>
        </ul>
        <?php else: ?>
            <p><strong>Status:</strong> Tidak ada kasus tercatat</p>
        <?php endif; ?>
    </div>
<?php endif; ?>
```

“aset/aset/2.1.png” could not be found.

Penjelasan Kode:

- Hasil data siswa dan daftar kasus ditampilkan dalam elemen `<div>` dan ``.
- Fungsi `htmlspecialchars()` digunakan untuk mencegah XSS dengan mengamankan data sebelum ditampilkan.

Terakhir, tambahkan bagian untuk tombol login/logout.

```
<div class="auth-buttons">
    <?php if (isset($_SESSION['user'])): ?>
        <a href="logout.php">Logout</a>
```

```

<?php else: ?>
    <a href="login.php">Login</a>
<?php endif; ?>
</div>

```

```

EXPLORER: SIKAS...  ⌂ ⌂ ⌂ ... index.php index.php
51   <html lang="id">
180  <body>
181    <div class="container">
213      <?php if ($hasil): ?>
236      <?php endif; ?>
237
238      <div class="auth-buttons">
239        <?php if (isset($_SESSION['user'])): ?>
240          <a href="logout.php">Logout</a>
241        <?php else: ?>
242          <a href="login.php">Login</a>
243        <?php endif; ?>
244      </div>
245    </div>
246  </body>
247
248  </html>

```

Penjelasan:

- <?php if (isset(\$_SESSION['user'])): ?> adalah pernyataan kondisi yang memeriksa apakah ada pengguna yang sedang login (dalam hal ini melalui sesi PHP).
- Jika sesi pengguna aktif (isset(\$_SESSION['user'])), maka tautan "Logout" akan ditampilkan.
- Sebaliknya, jika sesi pengguna tidak aktif, maka tautan "Login" akan ditampilkan.
- Ini memungkinkan pengguna untuk melihat tombol yang sesuai dengan status login mereka.

Jangan lupa untuk menutup tag </body> dan </html> di akhir file.

```

</body>
</html>

```

Penjelasan:

- Tag </body> menandakan bahwa konten halaman selesai dan bagian body telah selesai.
- Tag </html> menandakan bahwa dokumen HTML ini telah selesai.

Hasil Tampilan Homepage

Sistem Informasi Kasus Siswa
Masukkan NISN dan tanggal lahir untuk memeriksa data kasus

NISN
Masukkan NISN

Tanggal Lahir
mm / dd / yyyy

Cari Data

Login

Menguji Tampilan Home Page

Pertama-tama, masukkan **NISN** dan **tanggal lahir** yang valid ke dalam formulir yang disediakan. Langkah ini bertujuan untuk memastikan bahwa data yang diinput sesuai dengan informasi yang ada di database.

Sistem Informasi Kasus Siswa
Masukkan NISN dan tanggal lahir untuk memeriksa data kasus

NISN
0077865020

Tanggal Lahir
01 / 08 / 2007

Cari Data

Login

Hasil yang diharapkan:

Setelah data dimasukkan dengan benar, daftar kasus akan ditampilkan lengkap dengan deskripsi dan tanggal kejadian.

Sistem Informasi Kasus Siswa
Masukkan NISN dan tanggal lahir untuk memeriksa data kasus

NISN
0077865020

Tanggal Lahir
08/01/2007

Cari Data

Data ditemukan!

Informasi Kasus Siswa

NISN: 0077865020
Nama Lengkap: Andi Muh. Raihan Alkawsar
Tanggal Lahir: 01 August 2007

Daftar Kasus:

- Kasus: Meninggalkan kelas tanpa izin dari guru piket.
Tanggal Kejadian: 22 September 2024
- Kasus: Melanggar aturan perpustakaan dengan merobek halaman buku.
Tanggal Kejadian: 30 Oktober 2024
- Kasus: Berperilaku tidak sopan terhadap guru pengawas ujian.
Tanggal Kejadian: 31 Oktober 2024

Menguji Pencarian Data dengan NISN dan Tanggal Lahir yang Salah

Selanjutnya, masukkan **NISN** dan **tanggal lahir** yang tidak sesuai dengan data yang tersimpan di database. Hal ini dilakukan untuk menguji validasi dan penanganan kesalahan pada sistem.

Sistem Informasi Kasus Siswa
Masukkan NISN dan tanggal lahir untuk memeriksa data kasus

NISN
00778650201

Tanggal Lahir
02/12/109090

Cari Data

Login

Hasil yang diharapkan:

Dalam kondisi ini, sistem akan menampilkan pesan kesalahan berikut:
"Data siswa tidak ditemukan."

Sistem Informasi Kasus Siswa

Masukkan NISN dan tanggal lahir untuk memeriksa data kasus

NISN

00778650201

Tanggal Lahir

02/12/109090



Cari Data

Data siswa tidak ditemukan.

Login

Kesimpulan

Orangtua atau pengguna umum dapat mengakses data kasus siswa tanpa perlu login, cukup dengan memasukkan NISN dan tanggal lahir siswa. Hal ini memudahkan akses informasi bagi pihak yang membutuhkan, seperti orangtua siswa, tanpa harus terdaftar atau login terlebih dahulu.