

Pembuatan Database dan Tabel

Pembuatan Database

- Struktur Query:

```
CREATE DATABASE db_sekolah;
```

Pertama-tama, buatlah database baru dengan nama yang sesuai, db_sekolah .

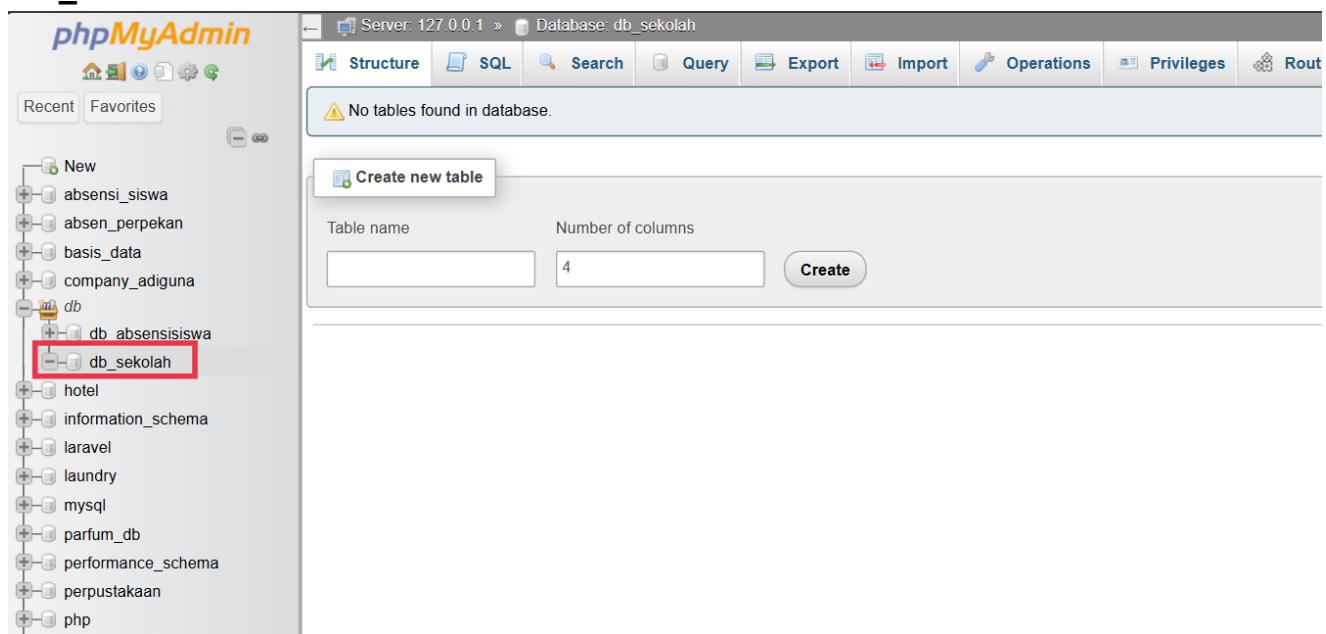
Buka phpMyAdmin dan pilih new pada menu sidebar

The screenshot shows the phpMyAdmin interface. On the left, there's a sidebar with a 'New' button highlighted by a red box. The main area is titled 'Create new table'. It has fields for 'Table name' (containing '4') and 'Number of columns'. A 'Create' button is at the bottom right of the form.

Setelah muncul tampilan seperti ini, masukkan nama database "db_sekolah" pada fitur create database lalu tekan tombol create

The screenshot shows the phpMyAdmin 'Databases' interface. On the left, the sidebar has a 'New' button highlighted by a red box. The main area shows a 'Create database' form with 'db_sekolah' in the input field and a 'Create' button highlighted by a red box. Below the form is a table with database information, and at the bottom is a toolbar with various icons.

Pada menu sidebar, akan muncul database yang telah dibuat yaitu dengan nama "**db_sekolah**"



The screenshot shows the phpMyAdmin interface. In the top navigation bar, 'Server: 127.0.0.1' and 'Database: db_sekolah' are selected. Below the navigation bar are tabs: Structure, SQL, Search, Query, Export, Import, Operations, Privileges, and Rout. On the left, a sidebar lists various databases, with 'db_sekolah' highlighted by a red box. The main area displays a 'Create new table' form with 'Table name' set to an empty field and 'Number of columns' set to 4, with a 'Create' button.

Penjelasan:

digunakan untuk membuat sebuah database baru dengan nama `db_sekolah`. Setelah query ini dieksekusi, sistem database akan mengalokasikan ruang untuk menyimpan data yang nantinya akan dimasukkan ke dalam database `db_sekolah`.

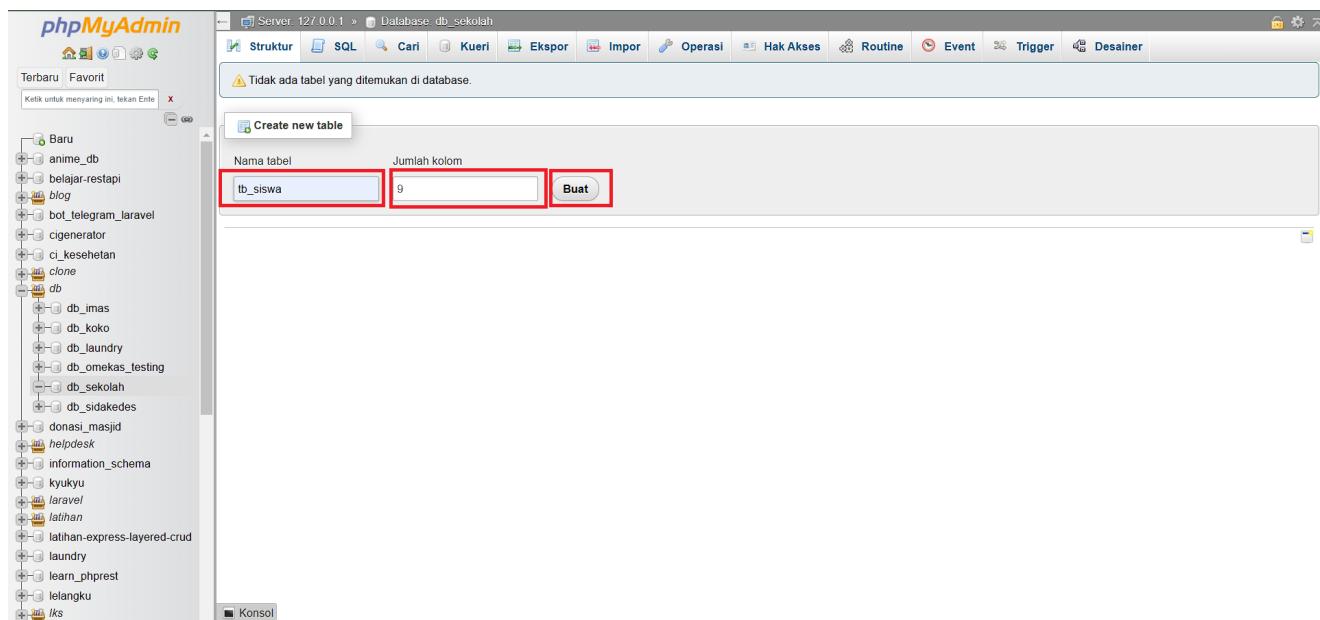
Pembuatan Tabel `tb_siswa`

- **Struktur Query:**

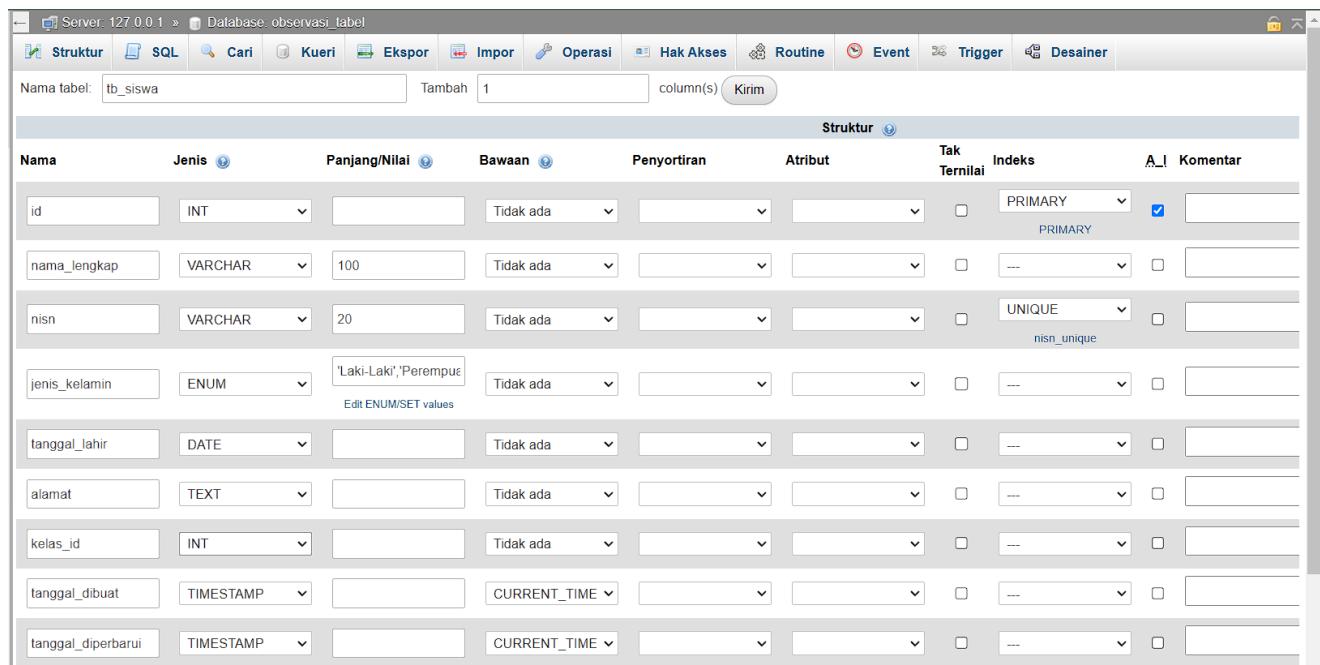
```
CREATE TABLE tb_siswa (
    id INT AUTO_INCREMENT PRIMARY KEY, -- ID unik untuk setiap siswa
    nama_lengkap VARCHAR(100) NOT NULL, -- Nama lengkap siswa
    nisn VARCHAR(20) NOT NULL UNIQUE, -- (NISN) siswa
    jenis_kelamin ENUM('Laki-Laki', 'Perempuan') NOT NULL, -- Jenis kelamin
    tanggal_lahir DATE NOT NULL, -- Tanggal lahir siswa
    alamat TEXT NOT NULL, -- Alamat lengkap siswa
    kelas_id INT NOT NULL, -- ID kelas siswa (relasi dengan tabel kelas)
    tanggal_dibuat TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- Waktu pembuatan
    data
    tanggal_diperbarui TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP -- Waktu pembaruan data terakhir
);
```

Pertama masuk ke dalam database yang sudah dibuat sebelumnya, setelah itu pilih menu **Create new table**, disini kita membuat tabel `tb_siswa` dengan 9 kolom. selanjutnya klik

tombol create.!



Setelah itu, masukkan rincian kolom seperti nama kolom, tipe data, panjang data. Pada tabel `tb_siswa` terdapat 9 kolom yaitu `id`, `nama_lengkap`, `nisn`, `jenis_kelamin`, `tanggal_lahir`, `alamat`, `kelas_id`, `tanggal_dibuat`, dan `tanggal_diperbarui`. Pastikan untuk menandai kolom `id` sebagai atribut Primary Key. Jangan lupa untuk menekan tombol **Save** setelah selesai mengkonfigurasi tabel.



Penjelasan:

ini digunakan untuk membuat tabel baru bernama `tb_siswa` dengan struktur kolom berikut:

- `id` : Kolom integer yang berfungsi sebagai **primary key** dan akan diisi secara otomatis dengan ID unik untuk setiap siswa (**AUTO_INCREMENT**).
- `nama_lengkap` : Kolom **VARCHAR(100)** yang wajib diisi (**NOT NULL**) untuk menyimpan nama lengkap siswa.

- `nisn` : Kolom **VARCHAR(20)** yang wajib diisi (**NOT NULL**) dan memiliki nilai unik (**UNIQUE**) untuk Nomor Induk Siswa Nasional (NISN) siswa.
 - `jenis_kelamin` : Kolom **ENUM** yang membatasi nilai untuk jenis kelamin siswa antara '**Laki-Laki**' dan '**Perempuan**', dan harus diisi (**NOT NULL**).
 - `tanggal_lahir` : Kolom **DATE** untuk mencatat tanggal lahir siswa yang harus diisi (**NOT NULL**).
 - `alamat` : Kolom **TEXT** untuk menyimpan alamat lengkap siswa yang wajib diisi (**NOT NULL**).
 - `kelas_id` : Kolom integer yang digunakan untuk menyimpan ID kelas siswa yang terkait. Kolom ini wajib diisi (**NOT NULL**) dan akan menjadi **foreign key** yang merujuk ke tabel kelas.
 - `tanggal_dibuat` : Kolom **TIMESTAMP** untuk mencatat waktu pembuatan data secara otomatis (**DEFAULT CURRENT_TIMESTAMP**).
 - `tanggal_diperbarui` : Kolom **TIMESTAMP** untuk mencatat waktu terakhir data diperbarui. Kolom ini akan diupdate otomatis setiap kali data diubah (**ON UPDATE CURRENT_TIMESTAMP**).
-

Pembuatan Tabel `tb_kelas`

- **Struktur:**

```
CREATE TABLE tb_kelas (
    id_kelas INT AUTO_INCREMENT PRIMARY KEY, -- ID unik untuk setiap kelas
    nama_kelas VARCHAR(50) NOT NULL, -- Nama kelas (contoh: X-A, XI-B)
    walikelas_id INT, -- ID wali kelas yang terhubung
);
```

- **Langkah - langkah:**

Pilih menu **Create new table**, disini kita membuat tabel `tb_kelas` dengan 3 kolom.

selanjutnya klik tombol **create**.

The screenshot shows the phpMyAdmin interface for a database named 'db_sekolah'. In the left sidebar, there's a list of tables including 'absensi_siswa', 'absen_perkepan', 'basis_data', 'company_adiguna', 'tb_siswa', 'db_absensisiswa', 'db_sekolah', 'tb_kelas', 'nopol', 'information_schema', 'aravel', 'audry', 'mysql', 'xarjum_db', 'performance_schema', 'perpusatakaan', 'xhp', 'xphmyadmin', 'xaktikum_7', 'xroject', and 'school_management'. The main area shows a table structure for 'tb_kelas' with 3 columns. A red box highlights the 'Create' button.

Setelah itu, masukkan rincian kolom seperti nama kolom, tipe data, panjang data. pada tabel **tb_kelas** terdapat 3 kolom yaitu **id_kelas** ,**nama_kelas** dan **walikelas_id** , Pastikan untuk menandai kolom **id_kelas** sebagai atribut Primary key . dan jangan lupa untuk menekan tombol Save setelah selesai mengkonfigurasi tabel.

The screenshot shows the 'Structure' tab for the 'tb_kelas' table. It lists three columns: 'id_kelas' (INT, 20), 'nama_kelas' (VARCHAR, 50), and 'walikelas_id' (INT, 20). The 'id_kelas' column has its 'Null' checkbox unchecked and its 'Index' dropdown set to 'PRIMARY'. The 'Save' button at the bottom is highlighted with a red box.

Penjelasan:

digunakan untuk membuat tabel baru bernama **tb_kelas** dengan struktur kolom berikut:

- **id_kelas** : Kolom integer yang otomatis bertambah nilainya (**AUTO_INCREMENT**) dan berfungsi sebagai **primary key** atau pengidentifikasi unik untuk setiap kelas.
- **nama_kelas** : Kolom string (VARCHAR) dengan panjang maksimum 50 karakter untuk menyimpan nama kelas, misalnya **X-A**, **XI-B**, dll. Kolom ini wajib diisi (**NOT NULL**).
- **walikelas_id** : Kolom integer yang akan menyimpan **ID wali kelas**. Kolom ini dapat digunakan untuk membuat relasi dengan tabel lain, misalnya tabel **wali kelas**.

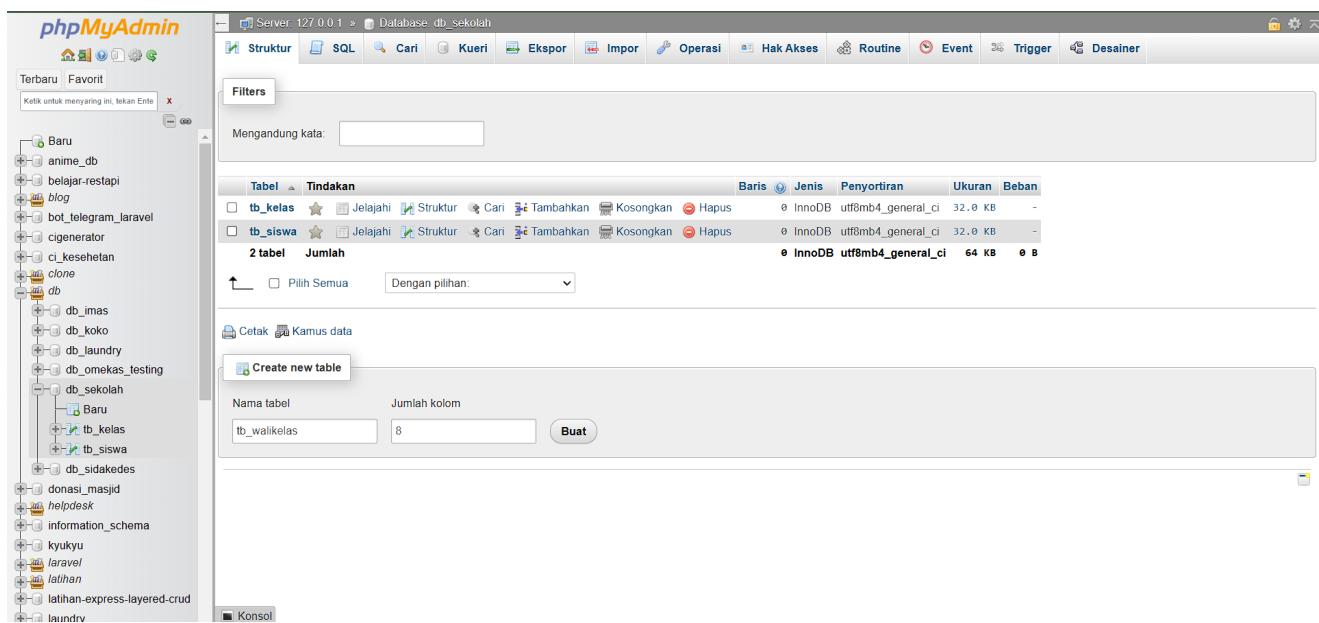
Pembuatan Tabel tb_walikelas

- Struktur:

```
CREATE TABLE tb_walikelas (
    id_walikelas INT AUTO_INCREMENT PRIMARY KEY,
    kelas VARCHAR(100) NOT NULL,
    nama_walikelas VARCHAR(100) NOT NULL,
    nip VARCHAR(20) UNIQUE NULL UNIQUE,
    jenis_kelamin ENUM('Laki-laki', 'Perempuan') NOT NULL,
    alamat TEXT,
    tanggal_dibuat TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    tanggal_diperbarui TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

- Langkah - langkah:

Pilih menu **Create new table**, disini kita membuat tabel `tb_walikelas` dengan 2 kolom. selanjutnya klik tombol **create**.



Setelah itu, masukkan rincian kolom seperti nama kolom, tipe data, panjang data. pada tabel `tb_kelas` terdapat 2 kolom yaitu `id_walikelas` dan `nama_walikelas`, Pastikan untuk menandai kolom `id_walikelas` sebagai atribut Primary key . dan jangan lupa untuk menekan tombol Save setelah selesai mengkonfigurasi tabel.

The screenshot shows the MySQL Workbench interface with the 'Struktur' (Structure) tab selected. A table named 'tb_walikelas' is being defined. The columns are as follows:

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Ternilai	Indeks	A_I	Komentar
id_walikelas	INT		Tidak ada			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
nama_walikelas	VARCHAR	100	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>	
nip	VARCHAR	20	Tidak ada			<input type="checkbox"/>	UNIQUE	<input type="checkbox"/>	nip_unique
jenis_kelamin	ENUM	'Laki-Laki','Perempuan'	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>	
alamat	TEXT		Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>	
tanggal_dibuat	TIMESTAMP		CURRENT_TIME			<input type="checkbox"/>	---	<input type="checkbox"/>	
tanggal_diperbarui	TIMESTAMP		CURRENT_TIMESTAMP			<input type="checkbox"/>	---	<input type="checkbox"/>	

Penjelasan:

ini membuat tabel baru bernama `tb_walikelas` dengan struktur kolom berikut:

- `id_walikelas` : Kolom integer sebagai **primary key**, berfungsi untuk memberikan ID unik bagi setiap wali kelas. Nilai akan bertambah otomatis (**AUTO_INCREMENT**).
- `nama_walikelas` : Kolom **VARCHAR(100)** yang wajib diisi (**NOT NULL**) untuk menyimpan nama lengkap wali kelas.
- `nip` : Kolom **VARCHAR(20)** untuk menyimpan NIP (Nomor Induk Pegawai) wali kelas. Bersifat unik (**UNIQUE**) dan opsional (**NULL**).
- `jenis_kelamin` : Kolom **ENUM** dengan nilai terbatas antara '**Laki-laki**' atau '**Perempuan**', digunakan untuk mencatat jenis kelamin wali kelas.
- `alamat` : Kolom **TEXT** untuk menyimpan alamat lengkap wali kelas.
- `tanggal_dibuat` : Kolom **TIMESTAMP** untuk mencatat waktu pembuatan data. Defaultnya adalah waktu saat data dibuat (**DEFAULT CURRENT_TIMESTAMP**).
- `tanggal_diperbarui` : Kolom **TIMESTAMP** untuk mencatat waktu terakhir data diperbarui. Secara otomatis diperbarui saat ada perubahan pada data (**ON UPDATE CURRENT_TIMESTAMP**).

Pembuatan Tabel `tb_kasus`

- **Struktur:**
- ```
CREATE TABLE tb_kasus (
 id_kasus INT AUTO_INCREMENT PRIMARY KEY,
 siswa_id INT,
 deskripsi_kasus TEXT,
 tanggal_kasus DATE
);
```

- Langkah - langkah:**

Pilih menu **Create new table**, disini kita membuat tabel `tb_kasus` dengan 4 kolom. selanjutnya klik tombol **create**.

The screenshot shows the phpMyAdmin interface for a database named 'db\_sekolah'. In the left sidebar, there's a tree view of databases and tables. The main area shows a table of existing tables: tb\_kelas, tb\_siswa, and tb\_walikelas. Below this, a 'Create new table' form is open. It has fields for 'Table name' (tb\_kasus) and 'Number of columns' (4). A red box highlights the 'Create' button, which is also highlighted with a red border.

Setelah itu, masukkan rincian kolom seperti nama kolom, tipe data, panjang data. pada tabel `tb_kelas` terdapat 4 kolom yaitu `id_kasus`, `siswa_id`, `deskripsi_kasus` dan `tanggal_kasus`, Pastikan untuk menandai kolom `id_kasus` sebagai atribut Primary key . dan jangan lupa untuk menekan tombol Save setelah selesai mengkonfigurasi tabel.

This screenshot shows the 'Struktur' (Structure) tab for the 'tb\_kasus' table. It lists four columns: 'id\_kasus' (INT, primary key, auto-increment), 'siswa\_id' (INT), 'deskripsi\_kasus' (TEXT), and 'tanggal\_kasus' (DATE). The 'id\_kasus' column is marked as 'PRIMARY' and has a red box around it. Other columns have checkboxes for 'Tak Ter nilai' and 'Indeks'. At the bottom, there are buttons for 'Pratinjau SQL' and 'Simpan'.

### Penjelasan:

ini akan membuat tabel `tb_kasus` dengan struktur sebagai berikut:

- `id_kasus` : Kolom integer yang berfungsi sebagai **primary key** dengan nilai unik untuk setiap kasus. Nilainya akan bertambah otomatis (**AUTO\_INCREMENT**).
- `siswa_id` : Kolom integer untuk menyimpan ID siswa yang terkait dengan kasus tersebut.

- **deskripsi\_kasus** : Kolom teks (**TEXT**) untuk menyimpan deskripsi atau penjelasan mengenai kasus.
  - **tanggal\_kasus** : Kolom **DATE** untuk menyimpan tanggal kejadian kasus.
- 

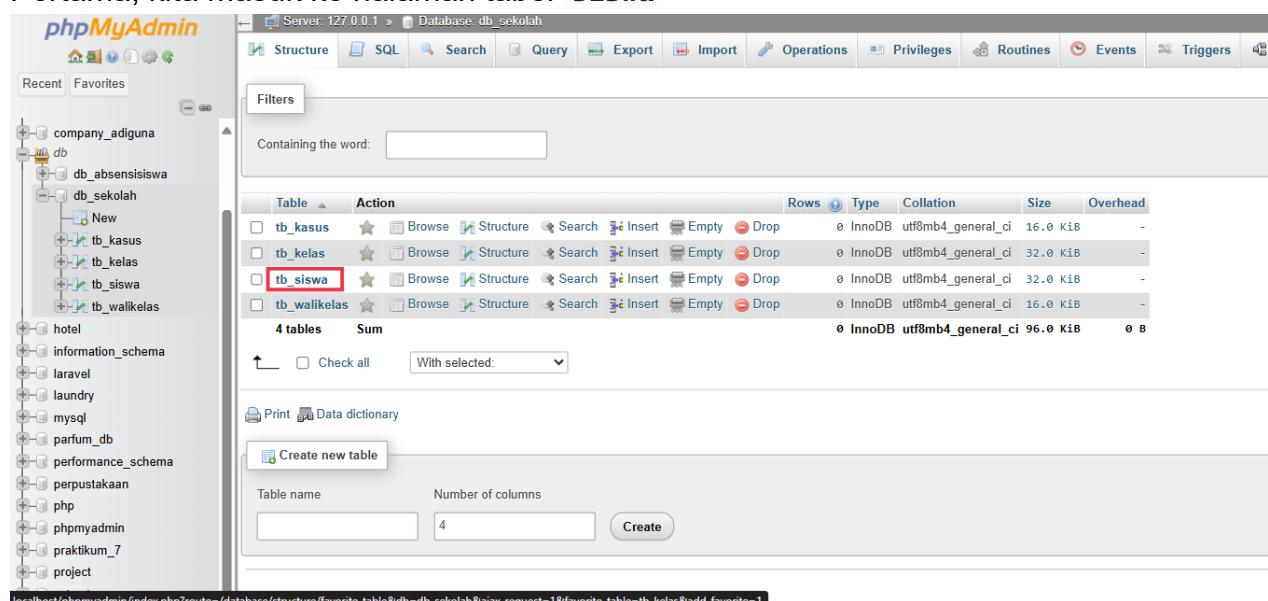
## Relasi Tabel

### Relasi Tabel Siswa-Kelas

Seperti yang kita ketahui, kita telah membuat tabel siswa yang mempunyai 3 kolom. Sekarang, kita ingin menghubungkan atau meng-relasikan kolom-kolom dari tabel siswa dengan ketiga tabel lainnya.

- **Langkah - langkah :**

Pertama, kita masuk ke halaman tabel `siswa`



| Table                     | Action                                                                                                                                    | Rows | Type   | Collation          | Size     | Overhead |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|------|--------|--------------------|----------|----------|
| <code>tb_kasus</code>     | <a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a> | 0    | InnoDB | utf8mb4_general_ci | 16.0 KiB | -        |
| <code>tb_kelas</code>     | <a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a> | 0    | InnoDB | utf8mb4_general_ci | 32.0 KiB | -        |
| <code>tb_siswa</code>     | <a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a> | 0    | InnoDB | utf8mb4_general_ci | 32.0 KiB | -        |
| <code>tb_walikelas</code> | <a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a> | 0    | InnoDB | utf8mb4_general_ci | 16.0 KiB | -        |

4 tables Sum

localhost:8080/admin/index.php?route=database/structure/favorite\_table&db=db\_sekolah&ajax\_request=1&favorite\_table=tb\_kelas&add\_favorite=1

Setelah itu, klik pada menu bar yaitu **Structure**. Selanjutnya, pada halaman **Structure** di tabel `tb_Siswa` terdapat tombol **Relation view / Tampilan Hubungan**.

The screenshot shows the phpMyAdmin interface for the db\_sekolah database. The left sidebar lists various databases and tables. The main area displays the structure of the tb\_siswa table. The table has 9 columns:

| # | Nama                      | Jenis                         | Penyortiran        | Atribut | Tak Ternilai | Bawaan              | Komentar       | Ekstra | Tindakan             |
|---|---------------------------|-------------------------------|--------------------|---------|--------------|---------------------|----------------|--------|----------------------|
| 1 | <b>id</b>                 | int(11)                       |                    |         | Tidak        | Tidak ada           | AUTO_INCREMENT |        | Ubah  Hapus  Lainnya |
| 2 | <b>nama_lengkap</b>       | varchar(100)                  | utf8mb4_general_ci |         | Tidak        | Tidak ada           |                |        | Ubah  Hapus  Lainnya |
| 3 | <b>nisn</b>               | varchar(20)                   | utf8mb4_general_ci |         | Tidak        | Tidak ada           |                |        | Ubah  Hapus  Lainnya |
| 4 | <b>jenis_kelamin</b>      | enum('Laki-Laki','Perempuan') | utf8mb4_general_ci |         | Tidak        | Tidak ada           |                |        | Ubah  Hapus  Lainnya |
| 5 | <b>tanggal_lahir</b>      | date                          |                    |         | Tidak        | Tidak ada           |                |        | Ubah  Hapus  Lainnya |
| 6 | <b>alamat</b>             | text                          | utf8mb4_general_ci |         | Tidak        | Tidak ada           |                |        | Ubah  Hapus  Lainnya |
| 7 | <b>kelas_id</b>           | int(11)                       |                    |         | Tidak        | Tidak ada           |                |        | Ubah  Hapus  Lainnya |
| 8 | <b>tanggal_dibuat</b>     | timestamp                     |                    |         | Tidak        | current_timestamp() |                |        | Ubah  Hapus  Lainnya |
| 9 | <b>tanggal_diperbarui</b> | timestamp                     |                    |         | Tidak        | current_timestamp() |                |        | Ubah  Hapus  Lainnya |

Below the table structure, there are buttons for Pilih Semua, Dengan pilihan:, Jelajahi, Ubah, Hapus, Utama, Unik, Indeks, Spasial, and Teks penuh. The 'Indeks' section shows two primary key constraints: PRIMARY on id and relasi\_kelas on kelas\_id.

Setelah di klik kita akan diarahkan ke halaman untuk membuat constraint. Pada constraint pertama yang akan kita buat adalah constraint antara kolom `kelas_id` dari tabel `siswa` ke kolom `id_kelas` dari tabel `tb_kelas`. Kita akan menamakan constraint ini "fkkelas\_siswa" dan mengubah `ON DELETE` dan `UPDATE constraint ini menjadi CASCADE` (yang berarti, jika data dari salah satu kolom yang terhubung di constraint ini terubah/terhapus maka datanya di kolom lain yang terhubung dengannya akan terubah/terhapus juga).

The screenshot shows the configuration of a foreign key constraint in the tb\_siswa table. The constraint is named `fk_kelas_siswa`. It maps the `kelas_id` column to the `id_kelas` column in the `tb_kelas` table. The `ON DELETE` and `ON UPDATE` actions are both set to `CASCADE`.

Pastikan untuk menekan tombol 'Save' setelah selesai mengkonfigurasi tabel.

## Relasi Tabel Kelas-Walikelas

Pertama, kita masuk ke halaman tabel tb\_kelas

The screenshot shows the phpMyAdmin interface for the 'db\_sekolah' database. On the left, there's a tree view of tables under the 'db\_sekolah' schema. In the main area, a table named 'tb\_kelas' is selected and highlighted with a red box. The table structure is shown with columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. Below the table, there are buttons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Routines, Events, and Triggers. A 'Filters' section at the top has a search bar. A 'Create new table' form is also visible.

Setelah itu, klik pada menu bar yaitu Structure . Selanjutnya, pada halaman Structure di tabel tb\_Kelas terdapat tombol Relation view .

This screenshot shows the 'Structure' tab for the 'tb\_kelas' table. The 'Relation view' button is highlighted with a red box. The table structure is displayed with columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. The 'Indexes' and 'Partitions' sections are also visible.

Setelah di klik kita akan diarahkan ke halaman untuk membuat constraint .

Pada constraint pertama yang akan kita buat adalah constraint antara kolom `walikelas_id` dari tabel `tb_kelas` ke kolom `id_walikelas` dari tabel `tb_walikelas` . Kita akan menamakan constraint ini “`fkwalikelas_kelas`” dan mengubah `ON DELETE` dan `UPDATE` constraint ini menjadi `CASCADE` (yang berarti, jika data dari salah satu kolom yang terhubung di constraint ini terubah/terhapus maka datanya di

kolom lain yang terhubung dengannya akan terubah/terhapus juga)\_.

The screenshot shows the 'Foreign key constraints' section of the phpMyAdmin interface for the 'tb\_kelas' table. A constraint named 'fk\_walikelas\_kelas' is being configured. The 'ON DELETE' and 'ON UPDATE' dropdowns are both set to 'CASCADE'. The 'Column' dropdown is set to 'walikelas\_id'. The 'Database' dropdown is set to 'db\_sekolah', the 'Table' dropdown is set to 'tb\_walikelas', and the 'Column' dropdown is set to 'id\_walikelas'. At the bottom, there are 'Preview SQL' and 'Save' buttons, with the 'Save' button highlighted by a red box.

Pastikan untuk menekan tombol 'Save' setelah selesai mengkonfigurasi tabel.

## Relasi Tabel Kasus-Siswa

Pertama, kita masuk ke halaman tabel tb\_kasus

The screenshot shows the 'Structure' tab of the phpMyAdmin interface for the 'db\_sekolah' database. The 'tb\_kasus' table is selected and highlighted with a red box. Other tables like 'tb\_kelas', 'tb\_siswa', and 'tb\_walikelas' are also listed. Below the table list, there is a 'Create new table' section where 'Table name' is set to 'tb\_kasus' and 'Number of columns' is set to '4'. The URL at the bottom of the browser window is 'localhost/phpmyadmin/index.php?route=/database/structure/favorite\_table&db=db\_sekolah&ajax\_request=1&favorite\_table=tb\_kelas&add\_favorite=1'.

Setelah itu, klik pada menu bar yaitu **Structure**. Selanjutnya, pada halaman **Structure** di tabel tb\_kasus terdapat tombol **Relation view**.

The screenshot shows the phpMyAdmin interface for the 'tb\_kasus' table. The 'Structure' tab is active. The table structure is as follows:

| # | Name            | Type    | Collation          | Attributes | Null | Default | Comments | Extra          | Action             |
|---|-----------------|---------|--------------------|------------|------|---------|----------|----------------|--------------------|
| 1 | id_kasus        | int(20) |                    |            | No   | None    |          | AUTO_INCREMENT | Change  Drop  More |
| 2 | siswa_id        | int(20) |                    |            | No   | None    |          |                | Change  Drop  More |
| 3 | deskripsi_kasus | text    | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| 4 | tanggal_kasus   | date    |                    |            | No   | None    |          |                | Change  Drop  More |

Below the table structure, there are tabs for 'Indexes' and 'Partitions'. The 'Indexes' tab shows a primary key constraint named 'PRIMARY' on the 'id\_kasus' column.

Setelah di klik kita akan diarahkan ke halaman untuk membuat constraint .

Pada constraint pertama yang akan kita buat adalah constraint antara kolom `siswa_id` dari tabel `tb_kasus` ke kolom `id_siswa` dari tabel `tb_siswa`. Kita akan menamakan constraint ini “`fksiswa_kasus`” dan mengubah `ON DELETE` dan `UPDATE constraint ini menjadi CASCADE` (yang berarti, jika data dari salah satu kolom yang terhubung di constraint ini terubah/terhapus maka datanya di kolom lain yang terhubung dengannya akan terubah/terhapus juga).

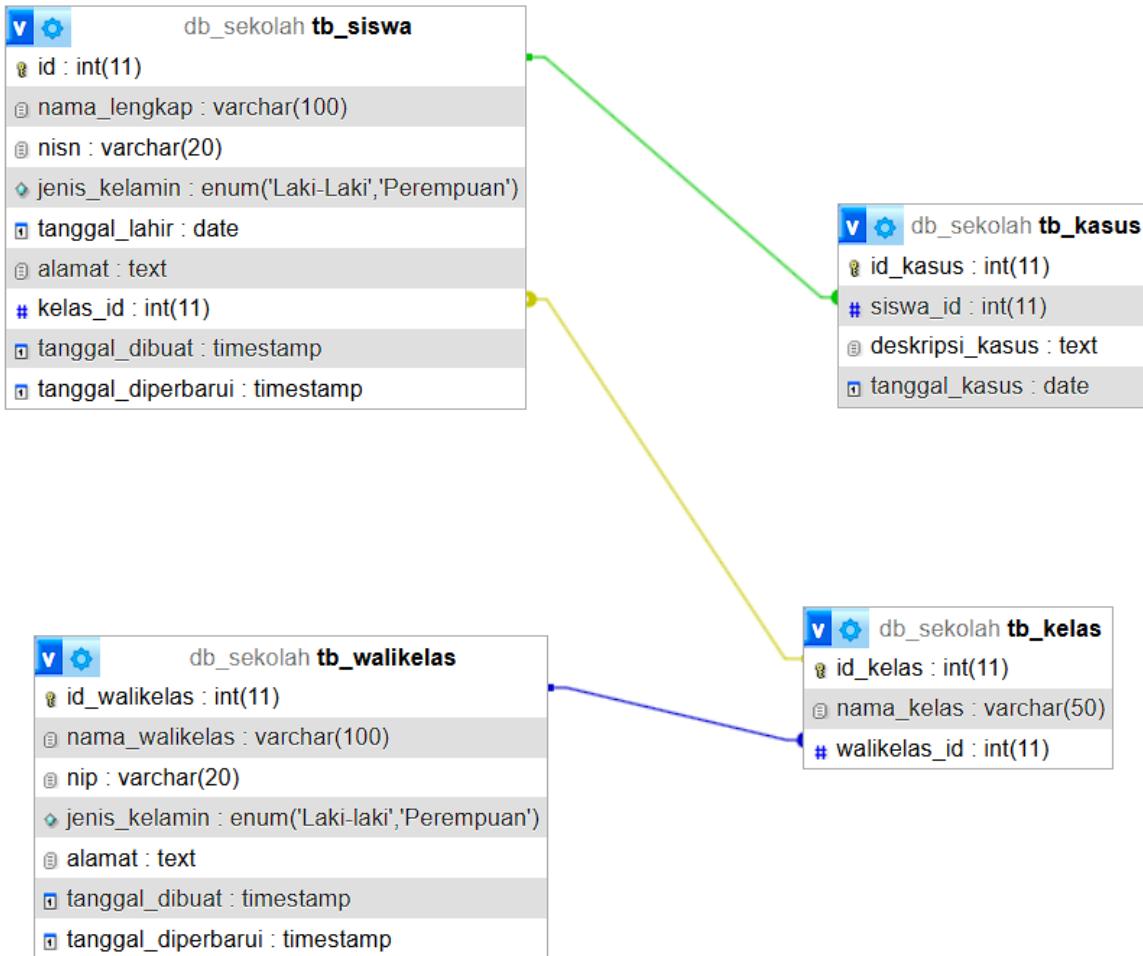
The screenshot shows the configuration of a foreign key constraint named 'fk\_siswa\_kasus'. The configuration is as follows:

- Constraint properties:**
  - Column:** siswa\_id
  - Foreign key constraint (INNODB):**
    - Database:** db\_sekolah
    - Table:** tb\_siswa
    - Column:** id\_siswa
- Actions:**
  - ON DELETE:** CASCADE
  - ON UPDATE:** CASCADE

At the bottom, there are buttons for 'Preview SQL' and 'Save', with 'Save' being highlighted.

Pastikan untuk menekan tombol 'Save' setelah selesai mengkonfigurasi tabel.

## Hasil Relasi



Anda dapat menggunakan data dummy untuk setiap tabel yang tersedia melalui tautan berikut:

[Data Dummy](#).

## Kesimpulan

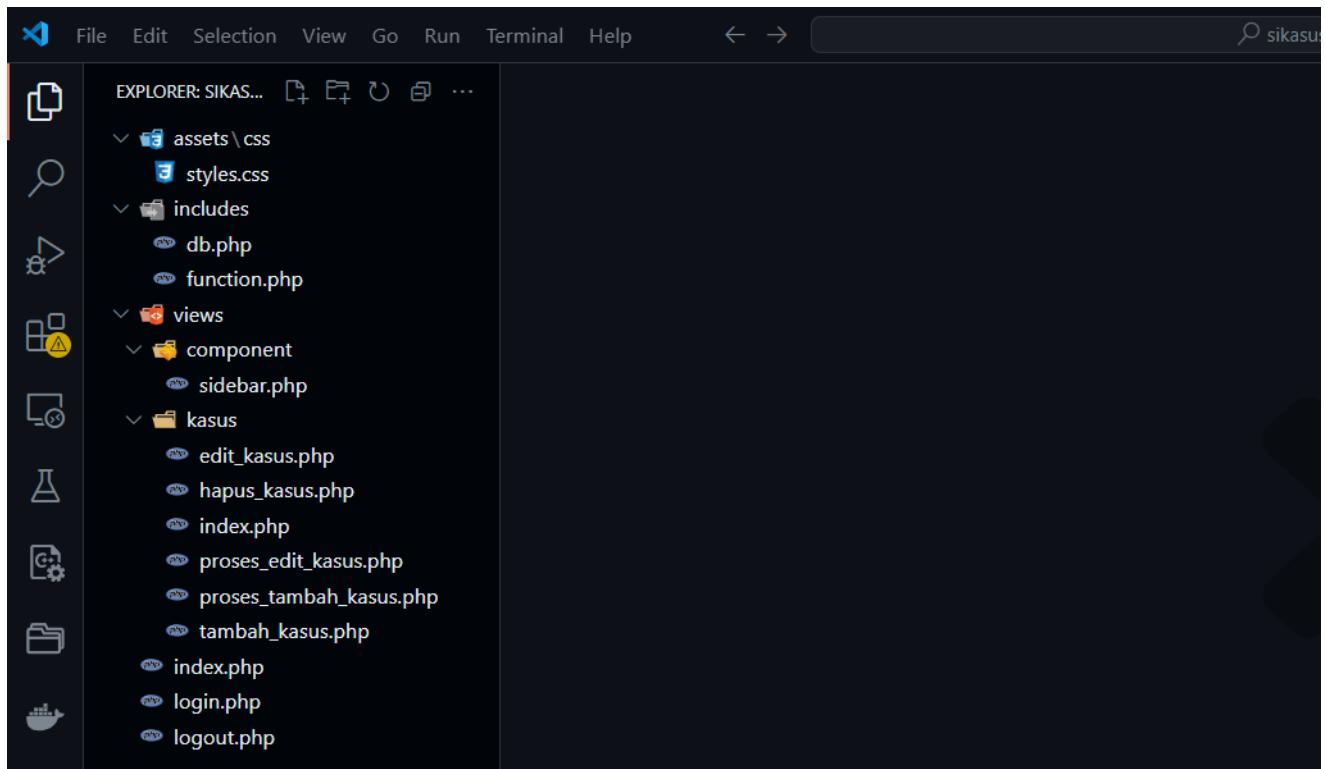
Dengan membuat keempat tabel ini, kita telah membangun fondasi struktur database yang dapat menampung data siswa, kelas, wali kelas, dan catatan kasus. Struktur ini memungkinkan relasi antar-tabel sehingga data dapat dikelola secara lebih terorganisir dan relevan untuk kebutuhan aplikasi berbasis web.

## Pembuatan Aplikasi Web

Pada bagian ini, kita akan membuat fitur-fitur utama untuk aplikasi web yang meliputi:

1. Melihat daftar kasus
2. Menambahkan kasus
3. Mengedit kasus

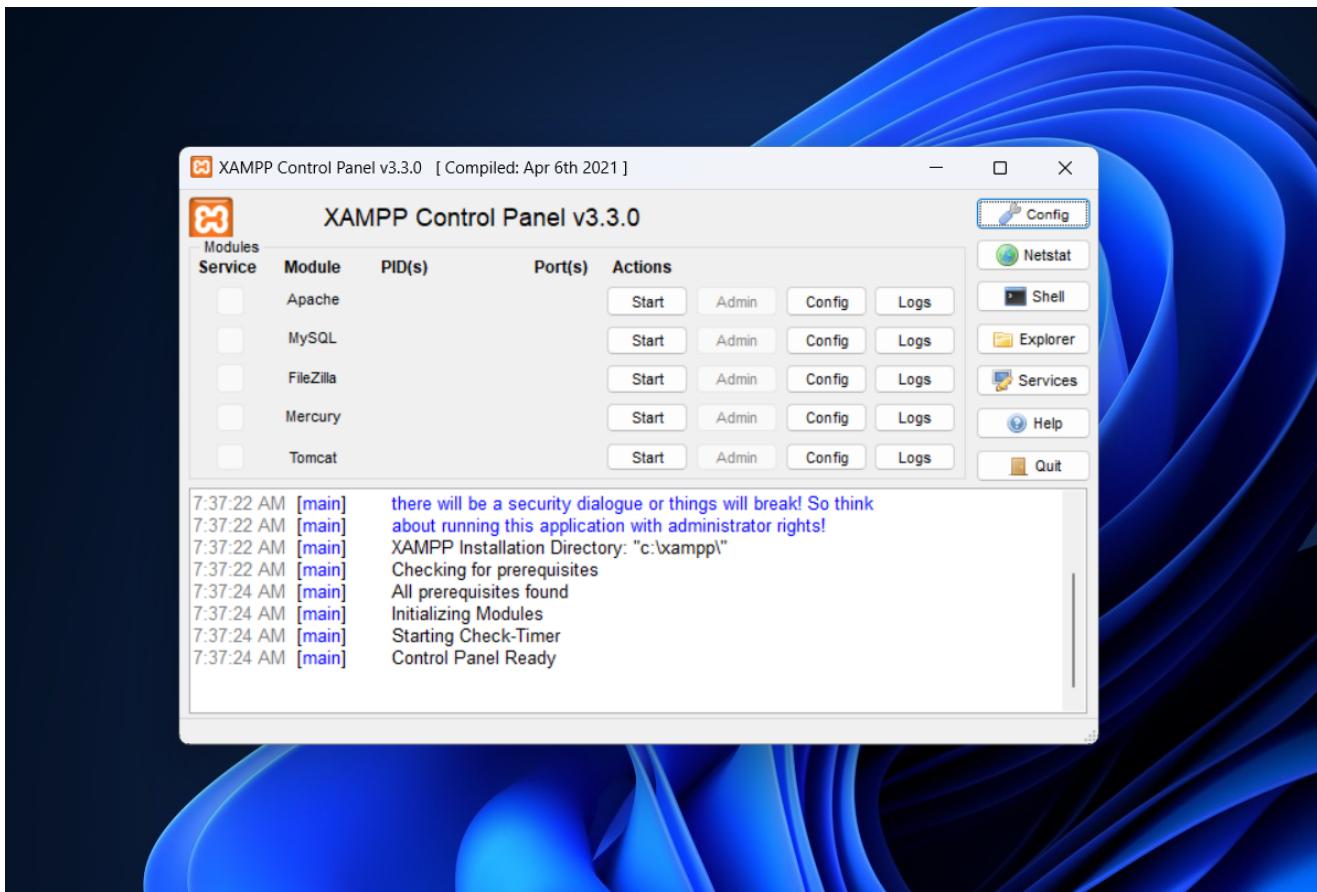
## Struktur Folder



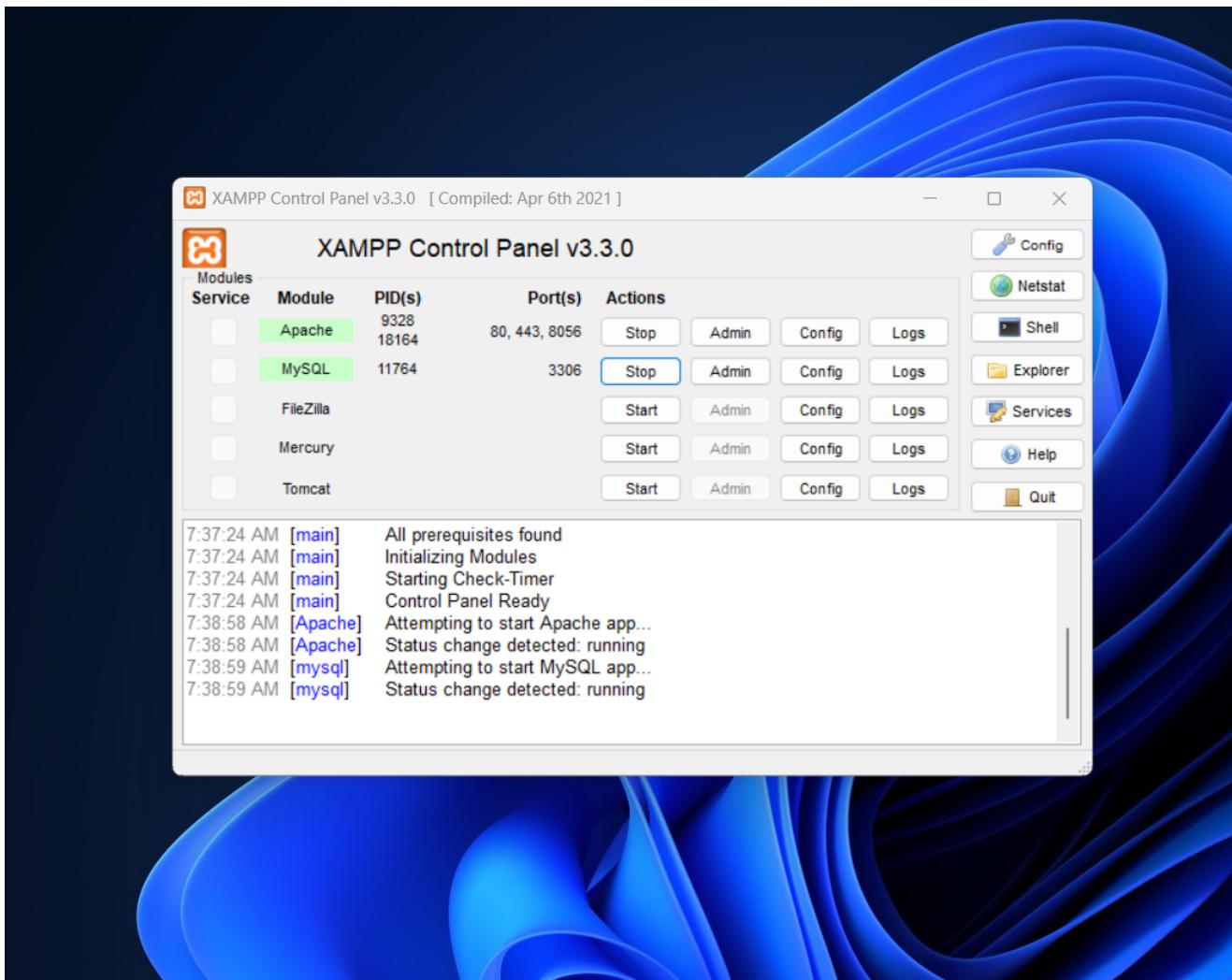
## Menjalankan Server Lokal

Setelah struktur folder berhasil disiapkan, langkah selanjutnya adalah menjalankan server lokal untuk menguji aplikasi Anda. Proses ini dapat dilakukan menggunakan aplikasi **XAMPP**

Pertama, **Buka Control Panel XAMPP di komputer Anda.**

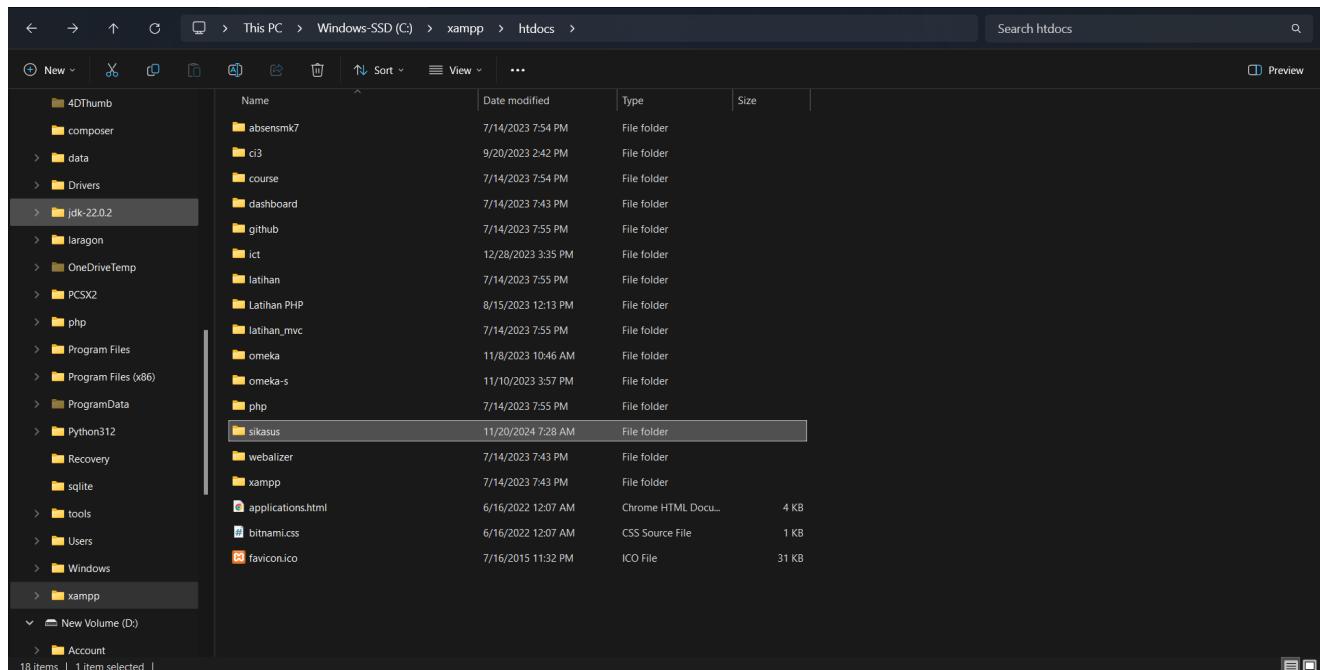


Cari modul **Apache** dan **MySQL** pada daftar layanan. Lalu klik tombol **Start** di sebelah masing-masing modul.



Jika berhasil, status akan berubah menjadi "Running" dengan warna hijau. Jika ada konflik port, pastikan port Apache tidak digunakan oleh aplikasi lain atau ubah portnya melalui konfigurasi.

Selanjutnya, salin folder proyek Anda ke direktori `C:/xampp/htdocs`. disini folder projek yang saya buat bernama `sikasus`



Kemudian, buka browser dan akses aplikasi Anda menggunakan URL seperti berikut:

```
http://localhost/sikasus/
```

## Membuat Koneksi Database

Fitur pertama yang akan kita buat adalah menampilkan daftar kasus.

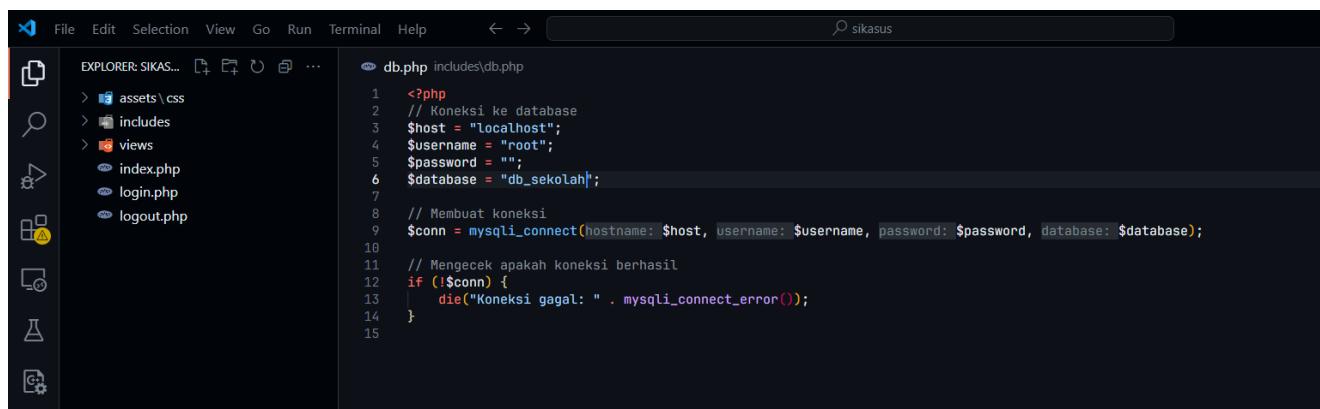
Sebelum kita mulai mengambil dan menampilkan data, kita perlu menghubungkan aplikasi dengan database. Buat file `db.php` dan masukkan kode berikut untuk menghubungkan aplikasi dengan MySQL. Buat file nya didalam folder `includes` sesuai dengan struktur folder yang saya berikan sebelumnya.

```
includes/db.php
```

```
<?php
// Koneksi ke database
$host = "localhost";
$username = "root";
$password = "";
$database = "db_sekolah";

// Membuat koneksi
$conn = mysqli_connect($host, $username, $password, $database);

// Mengecek apakah koneksi berhasil
if (!$conn) {
 die("Koneksi gagal: " . mysqli_connect_error());
}
```



```
db.php includes\db.php
1 <?php
2 // Koneksi ke database
3 $host = "localhost";
4 $username = "root";
5 $password = "";
6 $database = "db_sekolah";
7
8 // Membuat koneksi
9 $conn = mysqli_connect($host, $username, $password, $database);
10
11 // Mengecek apakah koneksi berhasil
12 if (!$conn) {
13 die("Koneksi gagal: " . mysqli_connect_error());
14 }
15
```

### Penjelasan :

- Baris `mysqli_connect()` digunakan untuk menghubungkan aplikasi ke database.

- Variabel `$host` , `$username` , `$password` , dan `$database` adalah konfigurasi koneksi database.
  - Fungsi `die()` akan menghentikan eksekusi dan menampilkan pesan jika koneksi gagal.
- 

## Menampilkan Daftar Kasus

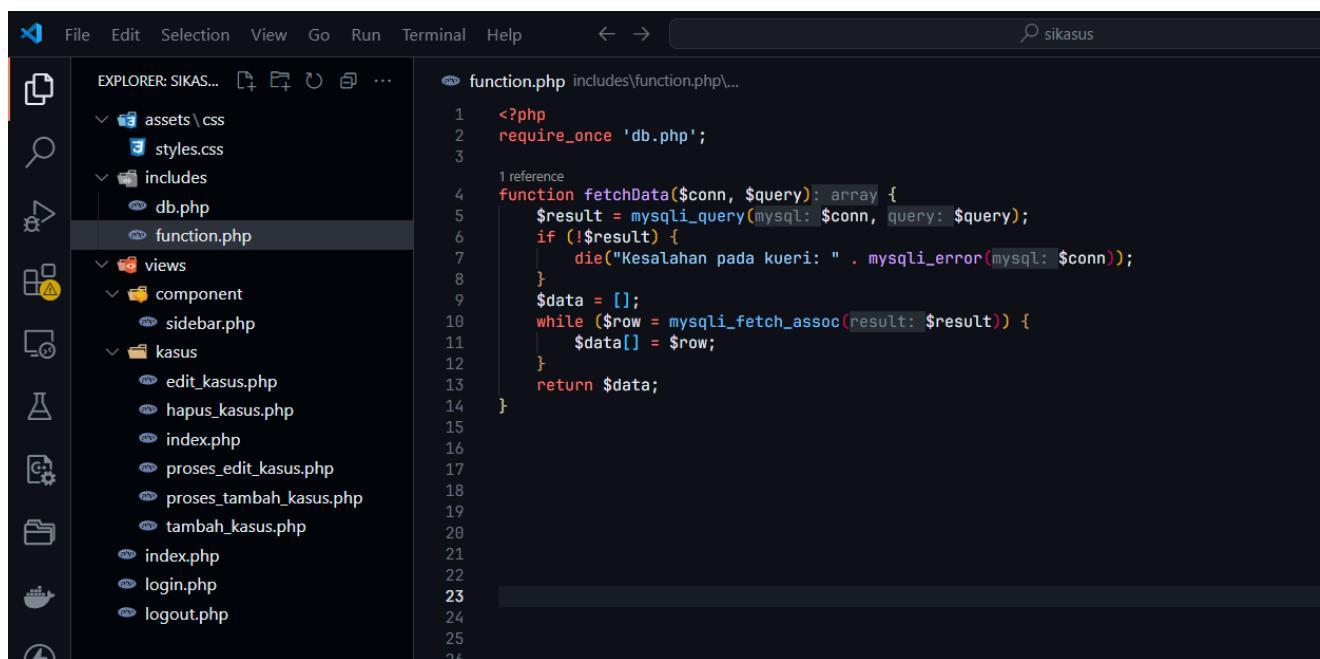
Pertama-tama, buat fungsi yang akan digunakan untuk mengambil data dari database. Simpan fungsi ini di dalam file `function.php`.

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

function fetchData($conn, $query) {
 $result = mysqli_query($conn, $query);

 if (!$result) {
 die("Query gagal: " . mysqli_error($conn));
 }

 $data = [];
 while ($row = mysqli_fetch_assoc($result)) {
 $data[] = $row;
 }
 return $data;
}
```



The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons for file operations like copy, paste, search, and refresh. The main area has two panes. The left pane shows a file tree with the following structure:

```

EXPLORER: SIKAS... includes\function.php\...
├── assets\css
│ └── styles.css
└── includes
 ├── db.php
 └── function.php
```
The 'function.php' file is currently selected in the file tree.

```

The right pane displays the contents of the 'function.php' file:

```

1 <?php
2 require_once 'db.php';
3
4 function fetchData($conn, $query): array {
5     $result = mysqli_query($conn, $query);
6     if (!$result) {
7         die("Kesalahan pada kueri: " . mysqli_error($conn));
8     }
9     $data = [];
10    while ($row = mysqli_fetch_assoc($result)) {
11        $data[] = $row;
12    }
13    return $data;
14
15
16
17
18
19
20
21
22
23
24
25
26
```
The code is color-coded, with syntax highlighting for PHP keywords, functions, and variables.

```

**Penjelasan :**

- `fetchData($conn, $query)` : Fungsi untuk mengeksekusi query dan mengembalikan hasil dalam bentuk array.
  - `mysqli_query()` : Menjalankan query pada database.
  - `mysqli_fetch_assoc()` : Mengambil setiap baris data hasil query sebagai array asosiatif.

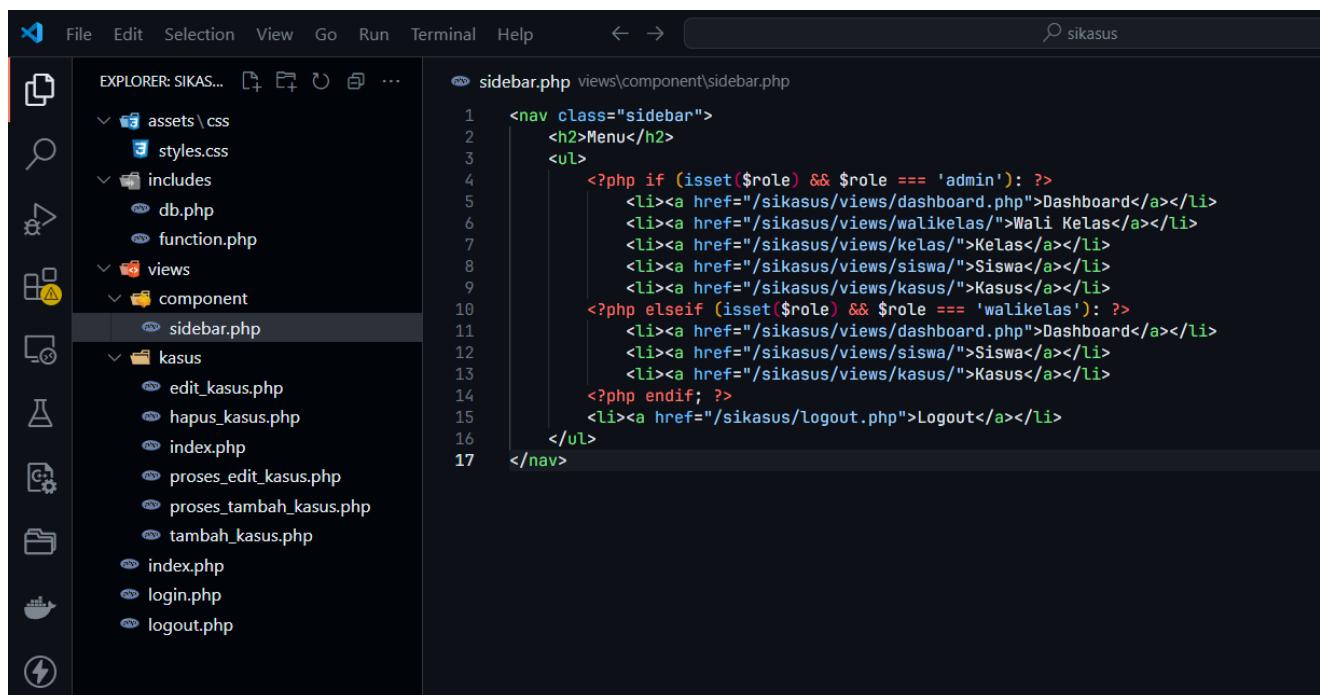
**Kemudian**, buat file bernama sidebar.php di dalam folder views/component . Sidebar ini akan memudahkan navigasi dalam aplikasi, menampilkan menu yang berbeda berdasarkan peran pengguna.

## view/scomponent/sidebar.php

```
<nav class="sidebar">
 <h2>Menu</h2>

 <?php if (isset($role) && $role === 'admin'): ?>
 Dashboard
 Wali Kelas
 Kelas
 Siswa
 Kasus
 <?php elseif (isset($role) && $role === 'walikelas'): ?>
 Dashboard
 Siswa
 Kasus
 <?php endif; ?>
 Logout

</nav>
```



## Penjelasan:

- `$role` : Variabel ini berisi peran pengguna (admin atau walikelas). Berdasarkan peran ini, menu yang berbeda akan ditampilkan.
- `isset()` : Fungsi ini memeriksa apakah variabel `$role` telah didefinisikan.

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

`views/kasus/index.php`

```
<?php
session_start();

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

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

$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
ORDER BY k.tanggal_kasus DESC";

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

The screenshot shows a code editor interface with a sidebar on the left containing a file tree. The tree shows a project structure with folders like 'assets', 'includes', 'views', 'component', and 'kasus'. Inside 'kasus', there are files: 'edit\_kasus.php', 'hapus\_kasus.php', and 'index.php'. The 'index.php' file is selected and its content is displayed in the main editor area. The code is identical to the one shown in the previous text block.

## 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
ORDER BY k.tanggal_kasus DESC;
```

**Penjelasan :** Query ini menampilkan data kasus siswa, termasuk deskripsi kasus dan nama siswa, dengan menggabungkan informasi dari tabel `tb_kasus` dan `tb_siswa`. Hasilnya diurutkan berdasarkan tanggal kasus yang terbaru.

### Penjelasan Kode:

- `$kasus_query` : Ini adalah query SQL yang mengambil semua data kasus dari tabel `tb_kasus` dan menghubungkannya dengan data siswa dari tabel `tb_siswa` menggunakan `LEFT JOIN`. Data diurutkan berdasarkan tanggal kasus.
- `fetchData()` : Fungsi ini digunakan untuk menjalankan query dan mengambil hasilnya. Fungsi ini didefinisikan dalam file `function.php` (yang akan dibahas di langkah berikutnya).

**Setelah itu**, kita akan menampilkan data kasus dalam bentuk tabel di halaman `index.php`

```
<!DOCTYPE html>
<html lang="id">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Daftar Kasus</title>
</head>
<body>
 <div class="wrapper">
 <!-- Sidebar -->
 <?php include($_SERVER['DOCUMENT_ROOT'] .
' /sikasus/views/component/sidebar.php'); ?>

 <!-- Main Content -->
 <div class="main-content">
 <header>
 <h1>Daftar Kasus</h1>
 </header>
 <div class="container">
 Tambah Kasus
 <table>
 <thead>
 <tr>
 <th width="10">#</th>
 <th width="10">Nama Siswa</th>
 <th width="10">Deskripsi Kasus</th>
 <th width="10">Tanggal Kasus</th>
 <th width="10">Aksi</th>
 </tr>
 </thead>
 </table>
 </div>
 </div>
 </div>
</body>
```

```

<tbody>
 <?php foreach ($kasus_result as $number => $kasus):
?>
 <tr>
 <td><?= $number += 1 ?></td>
 <td><?= $kasus['nama_lengkap'] ?></td>
 <td><?= $kasus['deskripsi_kasus'] ?></td>
 <td><?= $kasus['tanggal_kasus'] ?></td>
 <td>
 <a href="edit_kasus.php?id=<?=$kasus['id_kasus'] ?>" class="button">Edit
 <a href="hapus_kasus.php?id=<?=$kasus['id_kasus'] ?>" class="button"
 onclick="return confirm('Apakah Anda yakin ingin menghapus siswa ini?');">Hapus
 </td>
 </tr>
 <?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</div>
</body>
</html>

```

```

File Edit Selection View Go Run Terminal Help ↻ → ⌂ s kasus
EXPLORER: SIKAS... 📁 🗃 🗃 🗃 ...
index.php views\kasus\index.php
19 <html lang="id">
20 <body>
21 <div class="wrapper">
22 <!-- Header -->
23 <div class="main-content">
24 <!-- Main Content -->
25 <div class="container">
26 Tambah Kasus
27 <table>
28 <thead>
29 <tr>
30 <th width="10">#</th>
31 <th width="10">Nama Siswa</th>
32 <th width="10">Deskripsi Kasus</th>
33 <th width="10">Tanggal Kasus</th>
34 <th width="10">Aksi</th>
35 </tr>
36 </thead>
37 <tbody>
38 <?php foreach ($kasus_result as $number => $kasus): ?>
39 <tr>
40 <td><?= $number += 1 ?></td>
41 <td><?= $kasus['nama_lengkap'] ?></td>
42 <td><?= $kasus['deskripsi_kasus'] ?></td>
43 <td><?= $kasus['tanggal_kasus'] ?></td>
44 <td>
45 <a href="edit_kasus.php?id=<?=$kasus['id_kasus'] ?>" class="button">Edit
46 <a href="hapus_kasus.php?id=<?=$kasus['id_kasus'] ?>" class="button"
47 onclick="return confirm('Apakah Anda yakin ingin menghapus siswa ini?');">Hapus
48 </td>
49 </tr>
50 <?php endforeach; ?>
51 </tbody>
52 </table>
53 </div>
54 </body>
55 </html>

```

## Penjelasan:

- **foreach** : Digunakan untuk mengulangi setiap data kasus yang didapatkan dari hasil query dan menampilkannya dalam tabel HTML.

- `$number += 1` : Variabel ini digunakan untuk menampilkan nomor urut pada setiap baris data kasus.
- **Tautan Edit dan Hapus:** Menampilkan tombol untuk mengedit atau menghapus kasus tertentu.

**Selanjutnya**, kita akan menautkan file CSS untuk mempercantik tampilan web dengan CSS.

```

20
21 <head>
22 <meta charset="UTF-8">
23 <meta name="viewport" content="width=device-width, initial-scale=1.0">
24 <link rel="stylesheet" href="/sikasus/assets/css/styles.css">
25 <title>Daftar Kasus</title>
26 </head>
27

```

### Penjelasan:

- Di tahap ini, kita memastikan file CSS yang ada di folder `assets/css/` sudah terhubung dengan benar ke file HTML. Tanpa ini, CSS tidak akan diterapkan pada halaman

**Setelah itu**, kita akan membuat style CSS. berikut kodenya

`assets/css/styles.css`

```

/* Global Styles */
* {
 font-family: Arial, sans-serif;
 margin: 0;
 padding: 0;
 box-sizing: border-box;
}

body {
 background-color: #f4f4f9;
}

/* Wrapper for layout */
.wrapper {
 display: flex;
}

/* Sidebar */
.sidebar {
 width: 250px;
 background-color: #343a40;
 color: white;
}

```

```
height: 100vh;
padding: 1rem;
position: fixed;
}

.sidebar h2 {
 text-align: center;
 margin-bottom: 2rem;
}

.sidebar ul {
 list-style: none;
 padding: 0;
}

.sidebar ul li {
 margin: 0.8rem 0;
}

.sidebar ul li a {
 display: block;
 padding: 0.5rem;
 color: white;
 text-decoration: none;
 border-radius: 5px;
 transition: background-color 0.3s;
}

.sidebar ul li a:hover {
 background-color: #495057;
}

/* Main Content */
.main-content {
 margin-left: 250px;
 width: calc(100% - 250px);
 padding: 1rem;
}

/* Header */
header {
 padding: 1rem;
 text-align: center;
 text-decoration: underline;
 margin-bottom: 1rem;
}

/* Grid Layout for Content */
.content {
 display: grid;
```

```
grid-template-columns: repeat(3, 1fr);
gap: 1rem;
padding: 1rem;
}

/* Cards */
.card, .container, .profile-card, .cases-card {
 padding: 1rem;
 background-color: white;
 border-radius: 5px;
 box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
 margin-bottom: 1rem;
}

.card {
 text-align: center;
}

.container {
 max-width: 100%;
 margin: 1rem;
}

h2 {
 text-align: center;
}

/* Table Styles */
table {
 width: 100%;
 border-collapse: collapse;
 margin-top: 1rem;
}

table th, table td {
 padding: 0.5rem;
 text-align: left;
 border: 1px solid #ddd;
}

/* Buttons */
.button {
 display: inline-block;
 padding: 0.5rem 1rem;
 margin: 0.5rem 0;
 background-color: #007bff;
 color: white;
 text-align: center;
 border: none;
 border-radius: 5px;
}
```

```
 cursor: pointer;
 text-decoration: none;
}

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

/* Profile Section */

.content-siswa {
 padding: 1rem;
 background: #fff;
 border-radius: 5px;
 min-height: 100vh;
}

.profile-header {
 text-align: center;
}

.profile-avatar {
 font-size: 3rem;
 margin: 1rem 0;
 color: #007bff;
}

.profile-info {
 margin-bottom: 1rem;
}

.profile-info h1 {
 margin: 0.5rem 0;
}

/* Case Items */

.case-item {
 padding: 1rem;
 border-bottom: 1px solid #ddd;
}

.case-item:last-child {
 border-bottom: none;
}

/* Empty State */

.empty-state {
 text-align: center;
 padding: 2rem;
 color: #666;
}
```

```
.empty-state p {
 margin: 0.5rem 0;
}
```

```

.content-siswa {
 background: #f0f0f0;
 border-radius: 5px;
 min-height: 100vh;
}

.profile-header {
 text-align: center;
}

.profile-avatar {
 font-size: 3rem;
 margin: 1rem 0;
 color: #007bff;
}

.profile-info {
 margin-bottom: 1rem;
}

.profile-info h1 {
 margin: 0.5rem 0;
}

/* Case Items */
.case-item {
 padding: 1rem;
 border-bottom: 1px solid #ddd;
}

.case-item:last-child {
 border-bottom: none;
}

/* Empty State */
.empty-state {
 text-align: center;
 padding: 2rem;
 color: #666;
}

.empty-state p {
 margin: 0.5rem 0;
}

```

## Penjelasan :

- **Global Styles:**

- `* : Menetapkan aturan untuk semua elemen HTML, termasuk font, margin, padding, dan model box.`
- `body : Mengatur warna latar belakang halaman menjadi abu-abu terang (#f4f4f4).`

- **Layout Wrapper:**

- `.wrapper : Menggunakan flexbox untuk tata letak fleksibel, memungkinkan elemen di dalamnya diatur dengan mudah.`

- **Sidebar:**

- `.sidebar : Sidebar dengan lebar tetap ( 250px ), latar belakang gelap, tinggi penuh ( 100vh ), dan diposisikan tetap di layar.`

- **Sidebar Menu:**

- `.sidebar ul : Menghilangkan gaya daftar standar dan padding.`
- `.sidebar ul li : Menambahkan jarak antar item menu.`
- `.sidebar ul li a : Mengatur tampilan link menu, termasuk padding, warna, dan efek transisi.`

- **Main Content:**

- `.main-content : Area konten utama yang mengatur margin kiri untuk memberi ruang pada sidebar dan menghitung lebar yang tersisa.`

- **Header:**

- **header** : Menambahkan padding, menyelaraskan teks ke tengah, menambahkan garis bawah pada teks header, dan memberi jarak bawah.
- **Content Grid**:
  - **.content** : Menggunakan layout grid untuk membagi konten menjadi 3 kolom dengan jarak antar item.
- **Card**:
  - **.card , .container , .profile-card , .cases-card** : Kartu dengan padding, latar belakang putih, sudut melengkung, bayangan lembut, dan jarak bawah.
- **Table**:
  - **table** : Menetapkan tabel dengan lebar penuh dan pengaturan untuk menggabungkan border sel.
  - **table th , table td** : Menambahkan padding, penyelarasan kiri, dan batas tipis pada setiap sel tabel.
- **Button**:
  - **.button** : Tombol dengan padding, warna latar belakang biru, teks putih, dan efek hover untuk perubahan warna latar belakang.
- **Profile and Case Section**:
  - **.profile-header** : Menyelaraskan header profil ke tengah.
  - **.profile-avatar** : Menentukan ukuran font besar untuk avatar profil.
  - **.case-item** : Mengatur item kasus dengan padding dan garis pemisah bawah.
  - **.empty-state** : Menyusun tampilan status kosong dengan teks berwarna abu-abu dan padding besar.

## Sebelum :

#	Nama Siswa	Deskripsi Kasus	Tanggal Kasus	Aksi
1	Andi Muh. Raihan Alkawzar	Tidak hadir pada hari wajib kerja bakti sekolah.	2024-12-07	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Andi Muh. Raihan Alkawzar	Melanggar aturan ruang kesehatan dengan masuk tanpa izin.	2024-12-06	<a href="#">Edit</a> <a href="#">Hapus</a>
3	Andi Muh. Raihan Alkawzar	Menghilangkan buku perpustakaan.	2024-12-05	<a href="#">Edit</a> <a href="#">Hapus</a>
4	Andi Muh. Raihan Alkawzar	Menggunakan barang milik sekolah tanpa izin.	2024-12-04	<a href="#">Edit</a> <a href="#">Hapus</a>
5	Andi Muh. Raihan Alkawzar	Mengabaikan tugas piket dengan sengaja.	2024-12-03	<a href="#">Edit</a> <a href="#">Hapus</a>
6	Andi Muh. Raihan Alkawzar	Tidak menghadiri rapat OSIS yang wajib diikuti.	2024-12-02	<a href="#">Edit</a> <a href="#">Hapus</a>
	Andi Muh.	Mengganggu teman selama ...	...	...

## Sesudah :

The screenshot shows a web browser window with the URL `localhost/sikasus/views/kasus/index.php`. On the left, there is a dark sidebar with a 'Menu' button and a 'Logout' link. The main content area has a title 'Daftar Kasus' and a 'Tambah Kasus' button. Below is a table listing 8 cases:

#	Nama Siswa	Deskripsi Kasus	Tanggal Kasus	Aksi
1	Andi Muh. Raihan Alkawzar	Tidak hadir pada hari wajib kerja bakti sekolah.	2024-12-07	Edit Hapus
2	Andi Muh. Raihan Alkawzar	Menghilangkan buku perpustakaan.	2024-12-05	Edit Hapus
3	Andi Muh. Raihan Alkawzar	Menggunakan barang milik sekolah tanpa izin.	2024-12-04	Edit Hapus
4	Andi Muh. Raihan Alkawzar	Mengabaikan tugas piket dengan sengaja.	2024-12-03	Edit Hapus
5	Andi Muh. Raihan Alkawzar	Mengganggu teman selama diskusi kelompok berlangsung.	2024-12-01	Edit Hapus
6	Muhammad Farhan Maulana	Berbicara keras di ruang guru.	2024-11-30	Edit Hapus
7	Andi Muh. Raihan Alkawzar	Tidak membawa alat tulis saat ulangan harian.	2024-11-30	Edit Hapus
8	Muhammad Farhan Maulana	Melakukan tindakan bullying terhadap siswa	2024-11-29	Edit Hapus

## Membuat Halaman Tambah Data Kasus

Pertama-tama, buat file baru dengan nama `tambah_kasus.php` di dalam folder `views/kasus/`. File ini akan digunakan untuk menampilkan formulir yang memungkinkan pengguna memilih siswa, memasukkan deskripsi kasus, dan memilih tanggal kasus.

`views/kasus/tambah_kasus.php`

```
<?php
session_start();

// Menghubungkan ke database
require_once __DIR__ . '/../../../../includes/db.php';
```

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

- File Explorer:** Shows the project structure:
  - assets\css
  - includes
    - db.php
    - function.php
  - views
    - component
    - kasus
      - edit\_kasus.php
      - hapus\_kasus.php
      - index.php
      - proses\_edit\_kasus.php
      - proses\_tambah\_kasus.php
      - tambah\_kasus.php
    - index.php
    - login.php
    - logout.php
- Search Bar:** "sikasus"
- Code Editor:** The file "tambah\_kasus.php" is open, showing the following code:

```
<?php
session_start();
// Menghubungkan ke database
require_once __DIR__ . '/../../../../includes/db.php';
```

## Penjelasan:

- `session_start()` digunakan untuk memulai sesi yang memungkinkan Anda menyimpan dan mengakses data sesi, seperti status login pengguna.
- `require_once` digunakan untuk mengimpor file `db.php` yang berfungsi untuk menghubungkan aplikasi ke database, sehingga Anda bisa berinteraksi dengan data.

---

Berikutnya, mendapatkan daftar siswa dari database

```
// Mendapatkan daftar siswa dari database
$query = "SELECT id, nama_lengkap FROM tb_siswa";
$result = mysqli_query($conn, $query);
```

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a tree view of files and folders. The main area is the Code Editor showing a PHP file named 'tambah\_kasus.php'. The code in the editor is:

```
<?php
session_start();

// Menghubungkan ke database
require_once __DIR__ . '../../../../../includes/db.php';

// Mendapatkan daftar siswa dari database
$query = "SELECT id, nama_lengkap FROM tb_siswa";
$result = mysqli_query($conn, $query);
?

12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

## Query yang digunakan

```
SELECT id, nama_lengkap FROM tb_siswa
```

**Penjelasan :** Query ini digunakan untuk mengambil data dari tabel `tb_siswa`, khususnya kolom `id` dan `nama_lengkap`. Hasil query ini akan mengembalikan daftar ID dan nama lengkap siswa yang ada di tabel tersebut.

### Penjelasan:

- **Query SQL** digunakan untuk mengambil data siswa (id dan nama lengkap) dari tabel `tb_siswa`.
- `mysqli_query($conn, $query)` mengeksekusi query tersebut dan menyimpannya dalam variabel `$result`, yang kemudian digunakan untuk menampilkan daftar siswa pada dropdown menu di formulir.

---

**Berikutnya**, buat struktur HTML untuk formulir input data kasus

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

<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Tambah Kasus</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'); ?>

 <!-- Main Content -->
 <div class="main-content">
 <header>
 <h1>Tambah Kasus</h1>
 </header>
 <div class="container">
 <form action="proses_tambah_kasus.php" method="POST">
 <div>
 <label for="siswa_id">Pilih Siswa:</label>
 <select name="siswa_id" id="siswa_id" required>
 <option value="">-- Pilih Siswa --</option>
 <?php while ($row =
mysqli_fetch_assoc($result)): ?>
 <option value="<?=
htmlspecialchars($row['id']) ?>"><?= htmlspecialchars($row['nama_lengkap'])
?></option>
 <?php endwhile; ?>
 </select>
 </div>

 <div>
 <label for="deskripsi">Deskripsi Kasus:</label>
 <textarea name="deskripsi" id="deskripsi" required>
</textarea>
 </div>

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

 <button type="submit">Tambah Kasus</button>
 Kembali
 </form>
 </div>
 </div>
 </div>
</body>

</html>
```

```

14 <html lang="id">
15 <head>
16 <link rel="stylesheet" href="/sikasus/assets/css/styles.css">
17 </head>
18 <body>
19 <!-- Sidebar -->
20 <?php include($_SERVER['DOCUMENT_ROOT'] . '/sikasus/views/component/sidebar.php'); ?>
21
22 <!-- Main Content -->
23 <div class="main-content">
24 <header>
25 <h1>Tambah Kasus</h1>
26 </header>
27 <div class="container">
28 <form action="proses_tambah_kasus.php" method="POST">
29 <div>
30 <label for="siswa_id">Pilih Siswa:</label>
31 <select name="siswa_id" id="siswa_id" required>
32 <option value="">-- Pilih Siswa --</option>
33 <?php while ($row = mysqli_fetch_assoc(result: $result)): ?>
34 <option value='?= htmlspecialchars(string: $row['id']) ?><?= htmlspecialchars(string: $row['nama_lengkap']) ?></option>
35 <?php endwhile; ?>
36 </select>
37 </div>
38
39 <div>
40 <label for="deskripsi">Deskripsi Kasus:</label>
41 <textarea name="deskripsi" id="deskripsi" required></textarea>
42 </div>
43
44 <div>
45 <label for="tanggal">Tanggal Kasus:</label>
46 <input type="date" name="tanggal" id="tanggal" required>
47 </div>
48
49 <button type="submit">Tambah Kasus</button>
50 Kembali
51 </form>
52 </div>
53 </div>
54 </body>
55 </html>

```

## Penjelasan:

- `<form action="proses_tambah_kasus.php" method="POST">` : Menentukan bahwa data dari formulir ini akan dikirim menggunakan metode POST ke file `proses_tambah_kasus.php` .
- `<select> dan <option>` : Digunakan untuk membuat dropdown menu untuk memilih siswa. Data yang ditampilkan berasal dari hasil query yang sebelumnya telah diambil.
- `<textarea>` : Untuk input deskripsi kasus, yang mengharuskan pengguna menulis penjelasan mengenai kasus.
- `<input type="date">` : Untuk memilih tanggal kasus.
- `<input type="submit">` : Tombol untuk mengirim formulir.

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

`views\kasus\proses_tambah_kasus.php`

```

<?php

// Menghubungkan ke database
require_once __DIR__ . '/../../includes/db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 // Mengambil data dari formulir
 $siswa_id = $_POST['siswa_id'];
 $deskripsi = $_POST['deskripsi'];
}

```

```

$tanggal = $_POST['tanggal'];

// Validasi input
if (empty($siswa_id)) {
 header("Location: tambah_kasus.php");
 exit();
}

// Query menggunakan prepared statement untuk mencegah SQL injection
$query = "INSERT INTO tb_kasus (siswa_id, deskripsi_kasus,
tanggal_kasus) VALUES (?, ?, ?)";
$stmt = $conn->prepare($query);
$stmt->bind_param("iss", $siswa_id, $deskripsi, $tanggal);

// Mengeksekusi query dan memberikan feedback kepada pengguna
if ($stmt->execute()) {
 header("Location: ./index.php");
}

// Menutup statement
$stmt->close();
}

header("Location: ./index.php");
// Menutup koneksi
$conn->close();

```

```

<?php
// Menghubungkan ke database
require_once __DIR__ . '/../../includes/db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 // Mengambil data dari formulir
 $siswa_id = $_POST['siswa_id'];
 $deskripsi = $_POST['deskripsi'];
 $tanggal = $_POST['tanggal'];

 // Validasi input
 if (empty($siswa_id)) {
 header(header: "Location: tambah_kasus.php");
 exit();
 }

 // Query menggunakan prepared statement untuk mencegah SQL injection
 $query = "INSERT INTO tb_kasus (siswa_id, deskripsi_kasus, tanggal_kasus) VALUES (?, ?, ?)";
 $stmt = $conn->prepare($query);
 $stmt->bind_param(types: "iss", var: &$siswa_id, vars: &$deskripsi, $tanggal);

 // Mengeksekusi query dan memberikan feedback kepada pengguna
 if ($stmt->execute()) {
 header(header: "Location: ./index.php");
 }
}

// Menutup statement
$stmt->close();

header(header: "Location: ./index.php");
// Menutup koneksi
$conn->close();

```

## Query yang digunakan

`INSERT INTO tb_kasus (siswa_id, deskripsi_kasus, tanggal_kasus) VALUES (?, ?, ?)`

**Penjelasan :** Query ini digunakan untuk menambahkan data baru ke dalam tabel `tb_kasus`. Kolom `siswa_id`, `deskripsi_kasus`, dan `tanggal_kasus` akan diisi dengan nilai yang diberikan melalui parameter (?) . Perintah `INSERT INTO` akan menambahkan baris baru dengan data tersebut ke dalam tabel.

### **Penjelasan :**

- `require_once __DIR__ . '/../../includes/db.php';` : Menghubungkan file untuk konfigurasi database agar dapat digunakan dalam file ini.
- `if ($_SERVER['REQUEST_METHOD'] === 'POST') {` : Memastikan bahwa proses ini hanya dijalankan jika metode request adalah POST.
- `$query = "INSERT INTO tb_kasus (siswa_id, deskripsi_kasus, tanggal_kasus) VALUES (?, ?, ?);` : Menyusun query SQL untuk memasukkan data kasus baru.

---

## **Membuat Halaman Edit Data Kasus**

Pertama-tama, buat file baru bernama `edit_kasus.php` di dalam folder `views/kasus` . File ini digunakan untuk menampilkan formulir pengeditan kasus berdasarkan data yang sudah ada di database.

Selanjutnya, tambahkan kode berikut pada awal file untuk **menghubungkan ke database** dan **mengambil data kasus berdasarkan ID**.

`views/kasus/edit_kasus.php`

```
<?php
session_start();

// Menghubungkan ke database
require_once __DIR__ . '/../../includes/db.php';
require_once __DIR__ . '/../../includes/function.php';
```

The screenshot shows the VS Code interface. On the left is the Explorer sidebar with icons for files, folders, and search. The main area displays the file structure and content of `edit_kasus.php`. The file contains PHP code with line numbers from 1 to 24.

```
1 <?php
2 session_start();
3
4 require_once __DIR__ . '/../../includes/db.php';
5 require_once __DIR__ . '/../../includes/function.php';
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```

## Penjelasan :

- `session_start()` : Fungsi ini digunakan untuk memulai sesi PHP agar bisa menyimpan data sesi seperti login di masa depan.
- `require_once` : Memastikan file database hanya dipanggil sekali untuk menghindari konflik atau duplikasi.

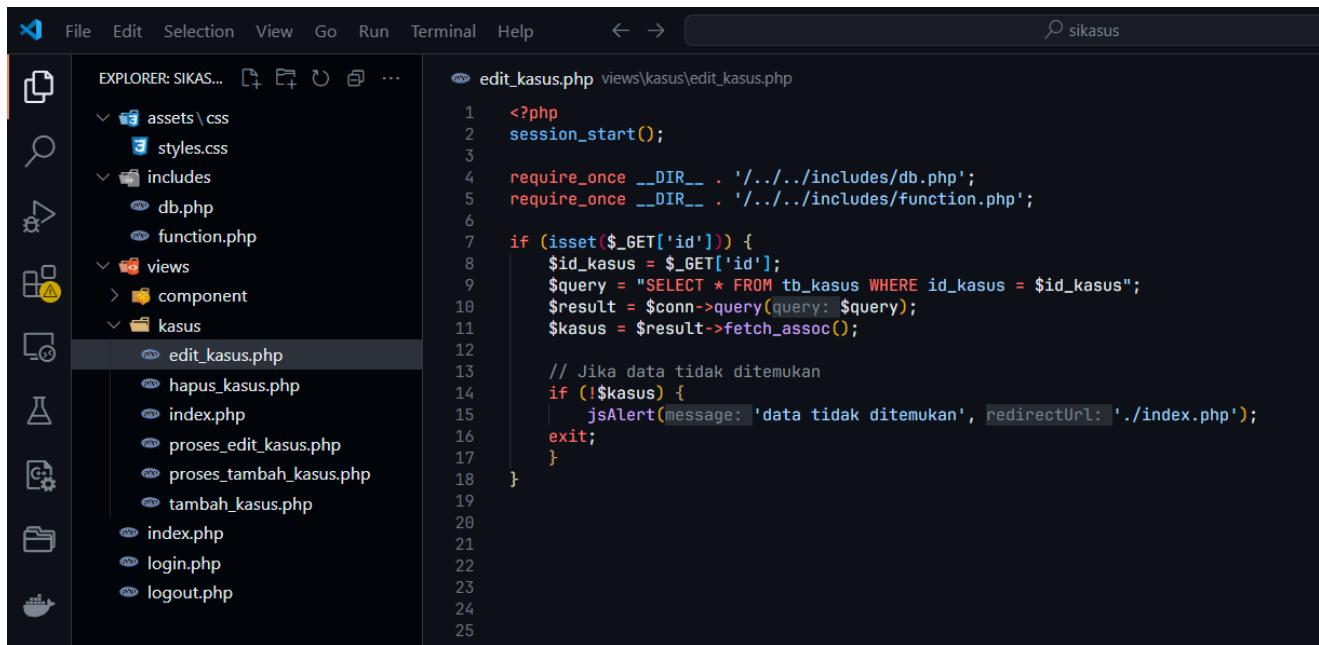
---

Kemudian, ambil data kasus dari database menggunakan parameter `id` yang dikirim melalui URL. Ini penting agar data yang akan diedit sudah muncul di formulir.

`views/kasus/edit_kasus.php`

```
if (isset($_GET['id'])) {
 $id_kasus = $_GET['id'];
 $query = "SELECT * FROM tb_kasus WHERE id_kasus = $id_kasus";
 $result = $conn->query($query);
 $kasus = $result->fetch_assoc();

 // Jika data tidak ditemukan
 if (!$kasus) {
 jsAlert('data tidak ditemukan', './index.php');
 exit;
 }
}
```



```
edit_kasus.php views\kasus\edit_kasus.php
1 <?php
2 session_start();
3
4 require_once __DIR__ . '/../../includes/db.php';
5 require_once __DIR__ . '/../../includes/function.php';
6
7 if (isset($_GET['id'])) {
8 $id_kasus = $_GET['id'];
9 $query = "SELECT * FROM tb_kasus WHERE id_kasus = $id_kasus";
10 $result = $conn->query(query: $query);
11 $kasus = $result->fetch_assoc();
12
13 // Jika data tidak ditemukan
14 if (!$kasus) {
15 jsAlert(message: 'data tidak ditemukan', redirectUrl: './index.php');
16 exit;
17 }
18
19
20
21
22
23
24
25
```

## Query yang digunakan

```
SELECT * FROM tb_kasus WHERE id_kasus = $id_kasus
```

**Penjelasan :** Query ini digunakan untuk mengambil semua data ( \* ) dari tabel tb\_kasus yang memiliki id\_kasus sesuai dengan nilai yang diberikan. Kondisi WHERE id\_kasus = \$id\_kasus memastikan hanya baris dengan id\_kasus yang sesuai yang akan diambil.

### Penjelasan :

- `if (isset($_GET['id'])) {` : Mengecek apakah parameter id ada dalam URL.
- `$query = "SELECT * FROM tb_kasus WHERE id_kasus = $id_kasus";` : Membuat query untuk mengambil data kasus berdasarkan id\_kasus .
- `$kasus = $result->fetch_assoc();` : Mengambil hasil query dan menyimpannya dalam variabel \$kasus .
- `if (!$kasus) { jsAlert('data tidak ditemukan', './index.php'); exit; }` : Jika data tidak ditemukan, menampilkan alert dan menghentikan eksekusi.

---

Selanjutnya, tambahkan query untuk **menampilkan daftar siswa** dari tabel tb\_siswa yang akan ditampilkan di dropdown.

`views/kasus/edit_kasus.php`

```
$query_siswa = "SELECT id, nama_lengkap FROM tb_siswa";
$result_siswa = mysqli_query($conn, $query_siswa);
```

```
<?php
session_start();
require_once __DIR__ . '/../../includes/db.php';
require_once __DIR__ . '/../../includes/function.php';
if (isset($_GET['id'])) {
 $id_kasus = $_GET['id'];
 $query = "SELECT * FROM tb_kasus WHERE id_kasus = $id_kasus";
 $result = $conn->query(query: $query);
 $kasus = $result->fetch_assoc();
}
$query_siswa = "SELECT id, nama_lengkap FROM tb_siswa";
$result_siswa = mysqli_query(mysql: $conn, query: $query_siswa);
?>
```

## Query yang digunakan

```
SELECT id, nama_lengkap FROM tb_siswa
```

**Penjelasan :** Query ini digunakan untuk mengambil data dari tabel `tb_siswa`. Kolom yang diambil adalah `id` dan `nama_lengkap`, yang akan mengembalikan daftar ID dan nama lengkap siswa yang ada dalam tabel.

### Penjelasan :

- `mysqli_query($conn, $query_siswa)` : Menjalankan query SQL untuk mengambil semua data siswa.
- Data ini akan digunakan untuk menampilkan pilihan siswa di formulir edit kasus.

---

Setelah data berhasil diambil, tambahkan struktur HTML untuk menampilkan formulir pengeditan.

`views/kasus/edit_kasus.php`

```
<form action="proses_edit_kasus.php" method="POST">
<input type="hidden" name="id" value="?=<?
htmlspecialchars($kasus['id_kasus']) ?>">

<div>
 <label for="siswa_id">Pilih Siswa:</label>
 <select name="siswa_id" id="siswa_id" required>
 <option value="">-- Pilih Siswa --</option>
 <?php while ($row = mysqli_fetch_assoc($result_siswa)): ?>
 <option value="?=<? htmlspecialchars($row['id']) ?>" ?=>
```

```

$row['id'] == $kasus['siswa_id'] ? 'selected' : '' ?>
 <?= htmlspecialchars($row['nama_lengkap']) ?>
</option>
<?php endwhile; ?>
</select>
</div>

<div>
 <label for="deskripsi">Deskripsi Kasus:</label>
 <textarea name="deskripsi" id="deskripsi"
 required><?= htmlspecialchars($kasus['deskripsi_kasus']) ?>
</textarea>
</div>

<div>
 <label for="tanggal">Tanggal Kasus:</label>
 <input type="date" name="tanggal" id="tanggal"
 value="<?= htmlspecialchars($kasus['tanggal_kasus']) ?>"
 required>
</div>

<button type="submit">Simpan Perubahan</button>
Kembali
</form>

```

```

File Edit Selection View Go Run Terminal Help ← → siklus
File Edit Selection View Go Run Terminal Help ← → siklus
EXPLORER: SIKUS...
assets\css
 styles.css
includes
 db.php
 function.php
views
 component
 kasus
 edit_kasus.php
 hapus_kasus.php
 index.php
 proses.edit_kasus.php
 proses.tambah.kasus.php
 tambah_kasus.php
 index.php
 login.php
 logout.php
edit_kasus.php
25 <html lang="id">
26 <body>
27 <div class="wrapper">
28 <!-- Sidebar -->
29 <?php include($_SERVER['DOCUMENT_ROOT'] . '/siklus/views/component/sidebar.php'); ?>
30
31 <!-- Main Content -->
32 <div class="main-content">
33 <header>
34 <h1>Edit Kasus</h1>
35 </header>
36 <div class="container">
37 <form action="proses_edit_kasus.php" method="POST">
38 <input type="hidden" name="id" value=<?= htmlspecialchars(string: $kasus['id_kasus']) ?>>
39
40 <div>
41 <label for="siswa_id">Pilih Siswa:</label>
42 <select name="siswa_id" id="siswa_id" required>
43 <option value="">-- Pilih Siswa --</option>
44 <?php while ($row = mysqli_fetch_assoc(result: $result_siswa)): ?>
45 <option value=<?= htmlspecialchars(string: $row['id']) ?>> <?= $row['id'] == $kasus['siswa_id'] ? 'selected' : '' ?>
46 </option>
47 </?php endwhile; ?>
48 </select>
49 </div>
50
51 <div>
52 <label for="deskripsi">Deskripsi Kasus:</label>
53 <textarea name="deskripsi" id="deskripsi"
54 required><?= htmlspecialchars(string: $kasus['deskripsi_kasus']) ?></textarea>
55 </div>
56
57 <div>
58 <label for="tanggal">Tanggal Kasus:</label>
59 <input type="date" name="tanggal" id="tanggal"
60 value="<?= htmlspecialchars(string: $kasus['tanggal_kasus']) ?>" required>
61 </div>
62
63 <button type="submit">Simpan Perubahan</button>
64
65 Kembali
66 </form>
67
68 </div>
69 </div>
70 </div>
71 </body>
72</html>

```

## Penjelasan :

- **Field Tersembunyi ( hidden )**: Menyimpan ID kasus agar dapat digunakan untuk query update.
- **Dropdown Siswa**: Menampilkan daftar siswa dari database. Jika siswa sudah dipilih sebelumnya, maka opsi akan otomatis terpilih.

- **Textarea Deskripsi:** Menampilkan deskripsi kasus yang sudah ada.
  - **Input Tanggal:** Menampilkan tanggal kasus.
- 

Setelah formulir dibuat, buat file `proses_edit_kasus.php` di folder yang sama untuk memproses data yang dikirim dari formulir.

`views/kasus/proses_edit_kasus.php`

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

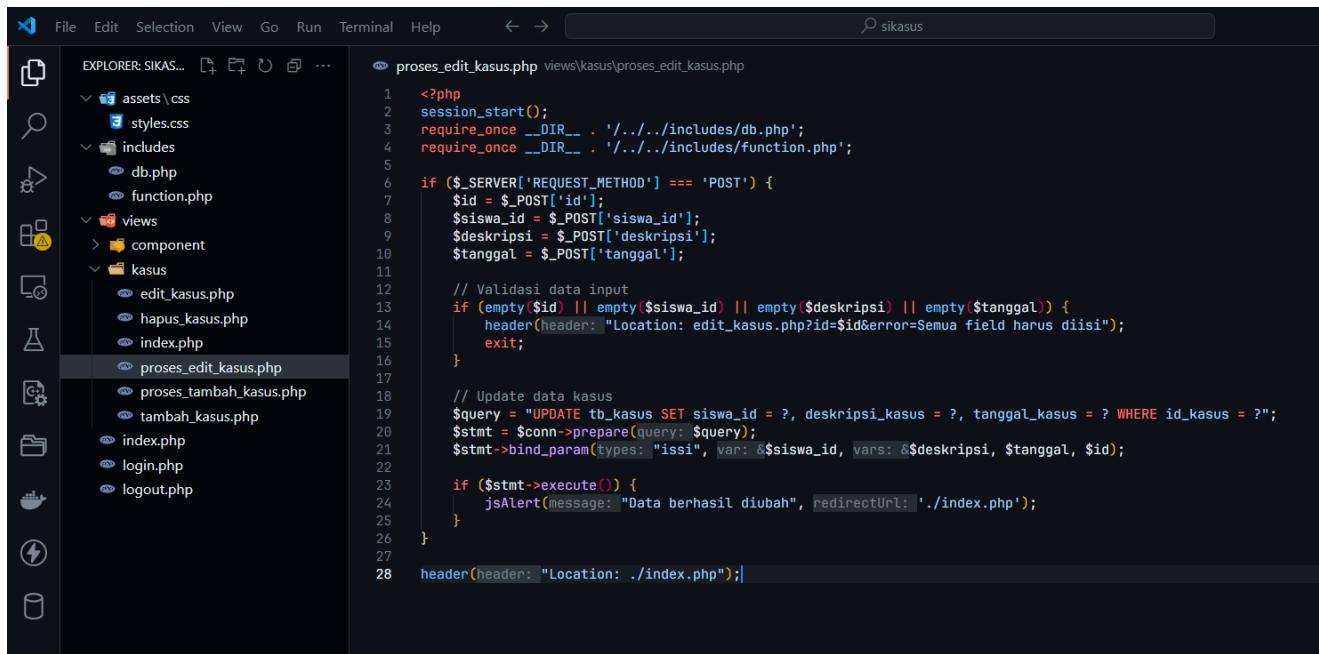
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 $id = $_POST['id'];
 $siswa_id = $_POST['siswa_id'];
 $deskripsi = $_POST['deskripsi'];
 $tanggal = $_POST['tanggal'];

 // Validasi data input
 if (empty($id) || empty($siswa_id) || empty($deskripsi) ||
empty($tanggal)) {
 header("Location: edit_kasus.php?id=$id&error=Semua field harus
diisi");
 exit;
 }

 // Update data kasus
 $query = "UPDATE tb_kasus SET siswa_id = ?, deskripsi_kasus = ?,
tanggal_kasus = ? WHERE id_kasus = ?";
 $stmt = $conn->prepare($query);
 $stmt->bind_param("issi", $siswa_id, $deskripsi, $tanggal, $id);

 if ($stmt->execute()) {
 jsAlert("Data berhasil diubah", './index.php');
 }
}

header("Location: ./index.php");
```



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

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 $id = $_POST['id'];
 $siswa_id = $_POST['siswa_id'];
 $deskripsi = $_POST['deskripsi'];
 $tanggal = $_POST['tanggal'];

 // Validasi data input
 if (empty($id) || empty($siswa_id) || empty($deskripsi) || empty($tanggal)) {
 header(header: "Location: edit_kasus.php?id=$id&error=Semua field harus diisi");
 exit;
 }

 // Update data kasus
 $query = "UPDATE tb_kasus SET siswa_id = ?, deskripsi_kasus = ?, tanggal_kasus = ? WHERE id_kasus = ?";
 $stmt = $conn->prepare(query: $query);
 $stmt->bind_param(types: "issi", var: &$siswa_id, vars: &$deskripsi, $tanggal, $id);

 if ($stmt->execute()) {
 jsAlert(message: "Data berhasil diubah", redirectUrl: './index.php');
 }
}

header(header: "Location: ./index.php");
```

## Query yang digunakan

```
UPDATE tb_kasus SET siswa_id = ?, deskripsi_kasus = ?, tanggal_kasus = ?
WHERE id_kasus = ?
```

**Penjelasan :** Query ini digunakan untuk memperbarui data pada tabel `tb_kasus`. Kolom `siswa_id`, `deskripsi_kasus`, dan `tanggal_kasus` akan diubah sesuai dengan nilai yang diberikan, sementara kondisi `WHERE id_kasus = ?` memastikan pembaruan hanya pada baris dengan `id_kasus` yang sesuai.

### Penjelasan :

- `$_SERVER['REQUEST_METHOD']` : Mengecek apakah request menggunakan metode POST.
- **Validasi:** Memastikan semua input diisi.
- **Query Update:** Menggunakan `prepare` dan `bind_param` untuk memperbarui data kasus.

## Membuat Halaman Hapus Data Kasus

Pertama-tama, buat file bernama `hapus_kasus.php` di dalam folder `views/kasus`. File ini bertugas untuk menghapus data kasus berdasarkan ID yang diterima melalui parameter URL.

Selanjutnya, tambahkan kode berikut untuk memulai sesi PHP dan menghubungkan file database serta fungsi tambahan:

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

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

The screenshot shows the Visual Studio Code interface. The left sidebar displays a tree view of the project structure under 'EXPLORER: SIKAS...'. The 'views\kasus' folder contains several PHP files: 'edit\_kasus.php', 'hapus\_kasus.php' (which is currently selected), 'index.php', 'proses\_edit\_kasus.php', 'proses\_tambah\_kasus.php', 'tambah\_kasus.php', 'index.php', 'login.php', and 'logout.php'. The right pane shows the content of 'hapus\_kasus.php'. The code is as follows:

```
1 <?php
2 session_start();
3
4 require_once __DIR__ . '/../../includes/db.php';
5 require_once __DIR__ . '/../../includes/function.php';
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```

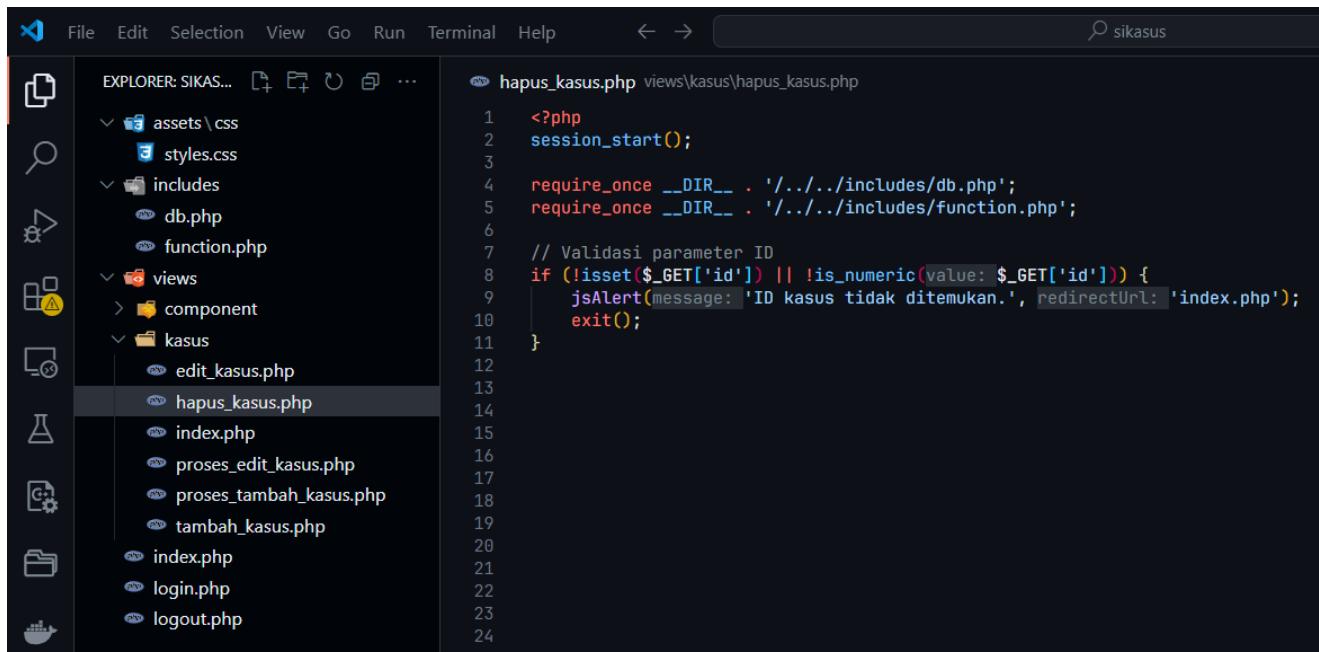
### Penjelasan:

- `session_start()` : Memulai sesi untuk memastikan semua fitur berbasis sesi dapat digunakan, seperti autentikasi pengguna.
- `require_once` : Mengimpor file `db.php` untuk koneksi database dan `function.php` untuk menggunakan fungsi tambahan seperti `alert`.

---

Berikutnya, tambahkan validasi untuk memeriksa keberadaan dan validitas parameter `id` yang diterima melalui URL. Jika parameter tidak ada atau tidak valid, maka pengguna akan diberi notifikasi.

```
// Validasi parameter ID
if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
 jsAlert('ID kasus tidak ditemukan.', 'index.php');
 exit();
}
```



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

// Validasi parameter ID
if (!isset($_GET['id']) || !is_numeric(value: $_GET['id'])) {
 jsAlert(message: 'ID kasus tidak ditemukan.', redirectUrl: 'index.php');
 exit();
}

$id_kasus = intval($_GET['id']);
```

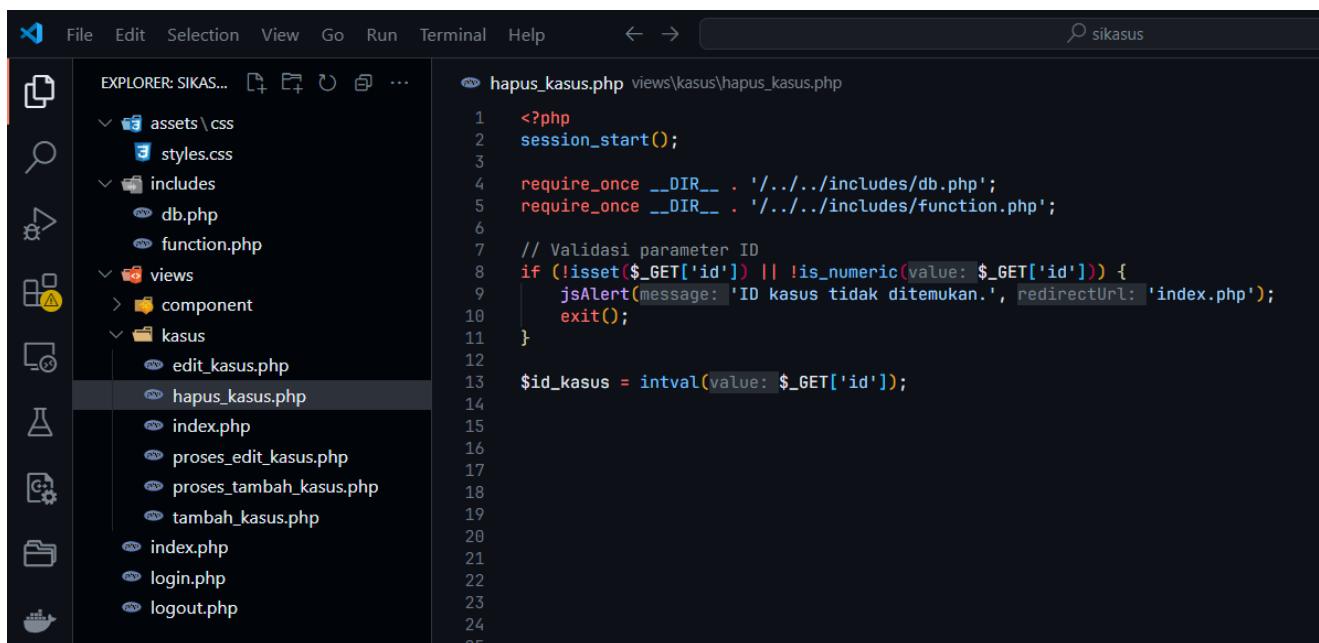
## Penjelasan:

- `isset($_GET['id'])` : Memastikan bahwa parameter `id` dikirim melalui URL.
- `is_numeric($_GET['id'])` : Mengecek apakah nilai `id` adalah angka.
- `jsAlert()` : Menampilkan pesan kesalahan dan mengarahkan pengguna kembali ke halaman utama jika validasi gagal.

---

Setelah validasi, konversi parameter `id` menjadi integer untuk keamanan tambahan dan kemudahan penggunaan.

```
$id_kasus = intval($_GET['id']);
```



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

// Validasi parameter ID
if (!isset($_GET['id']) || !is_numeric(value: $_GET['id'])) {
 jsAlert(message: 'ID kasus tidak ditemukan.', redirectUrl: 'index.php');
 exit();
}

$id_kasus = intval(value: $_GET['id']);
```

## Penjelasan:

- `intval()` : Mengubah nilai parameter `id` menjadi tipe data integer untuk menghindari kemungkinan serangan injeksi.

Lakukan query untuk menghapus data kasus dari tabel `tb_kasus` berdasarkan `id_kasus` yang diterima.

```
// Query untuk menghapus kasus
$delete_query = "DELETE FROM tb_kasus WHERE id_kasus = $id_kasus";
```

The screenshot shows a code editor interface with a sidebar containing project files and a main panel displaying a PHP script. The sidebar shows a folder structure under 'EXPLORER: SIKAS...' with 'assets\css' (containing 'styles.css'), 'includes' (containing 'db.php' and 'function.php'), and 'views' (containing 'component' and 'kasus'). The 'kasus' folder contains several files: 'edit\_kasus.php', 'hapus\_kasus.php' (which is selected), 'index.php', 'proses\_edit\_kasus.php', 'proses\_tambah\_kasus.php', 'tambah\_kasus.php', 'index.php', 'login.php', and 'logout.php'. The main panel displays the following PHP code:

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

// Validasi parameter ID
if (!isset($_GET['id']) || !is_numeric(value: $_GET['id'])) {
 jsAlert(message: 'ID kasus tidak ditemukan.', redirectUrl: 'index.php');
 exit();
}

$id_kasus = intval(value: $_GET['id']);

// Query untuk menghapus kasus
$delete_query = "DELETE FROM tb_kasus WHERE id_kasus = $id_kasus";
```

## Query yang digunakan

```
DELETE FROM tb_kasus WHERE id_kasus = $id_kasus
```

**Penjelasan :** Query ini digunakan untuk menghapus data dari tabel `tb_kasus` berdasarkan `id_kasus`. Kondisi `WHERE id_kasus = $id_kasus` memastikan hanya baris dengan `id_kasus` yang sesuai dengan nilai yang diberikan yang akan dihapus.

### Penjelasan:

- **Query SQL:** Perintah `DELETE` digunakan untuk menghapus data pada tabel `tb_kasus` yang memiliki nilai `id_kasus` tertentu.

Setelah query dibuat, eksekusi query tersebut. Jika berhasil, tampilkan pesan berhasil, jika tidak, tampilkan pesan kesalahan.

```
// Eksekusi query dan redirect dengan notifikasi
if (mysqli_query($conn, $delete_query)) {
 jsAlert('Kasus berhasil dihapus.', 'index.php');
} else {
 jsAlert('Kesalahan saat menghapus kasus.', 'index.php');
}
exit();
```

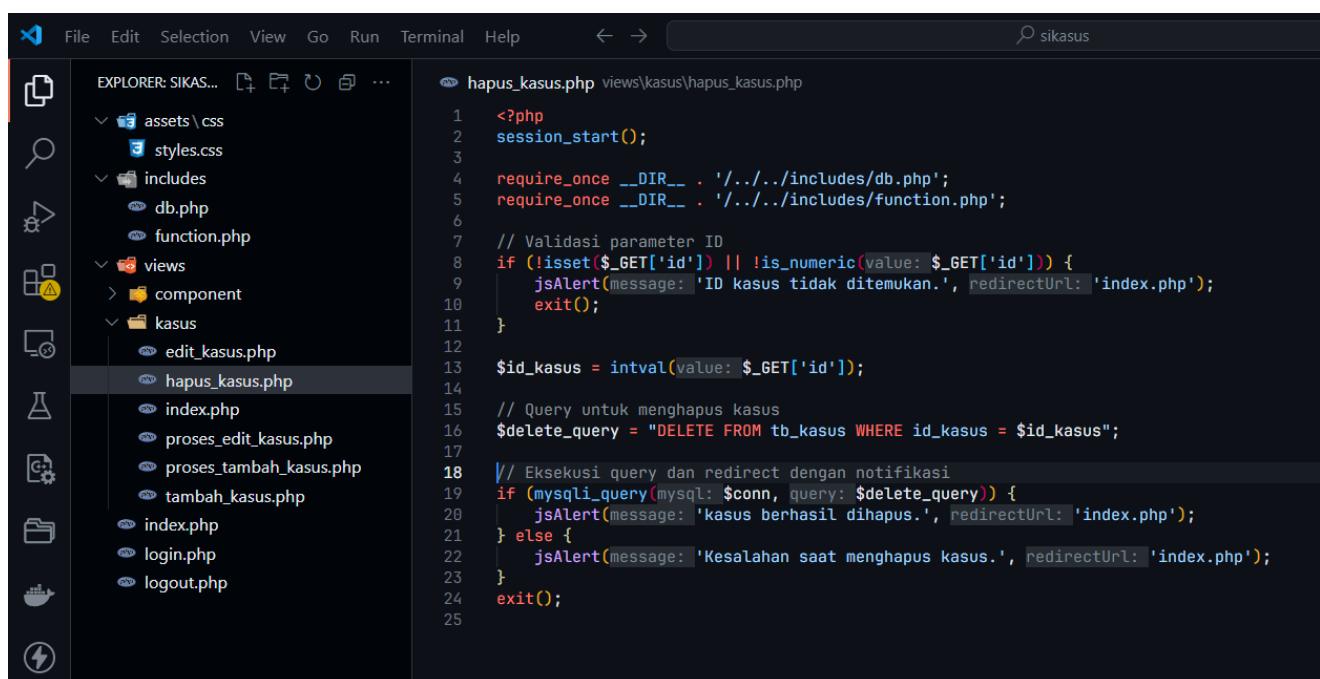
## Penjelasan:

- `mysqli_query()` : Menjalankan query SQL untuk menghapus data.
  - `jsAlert()` : Menampilkan notifikasi kepada pengguna, baik ketika penghapusan berhasil maupun gagal.

Fungsi ini telah didefinisikan di file `function.php` dan digunakan untuk menampilkan alert JavaScript sekaligus melakukan pengalihan (redirect).

Berikut adalah implementasi dari fungsi jsAlert :

```
function jsAlert($message, $redirectUrl = null) {
 echo "<script>";
 echo "alert('" . addslashes($message) . "');";
 if ($redirectUrl) {
 echo "window.location.href = '" . addslashes($redirectUrl) . "';";
 }
 echo "</script>";
 exit();
}
```



## Penjelasan:

- `addslashes()` : Mengamankan teks pesan agar tidak terjadi error pada JavaScript.
- **Redirect**: Jika parameter `$redirectUrl` diberikan, pengguna akan diarahkan ke URL yang ditentukan setelah menutup alert.

## Kesimpulan

Pada tahap awal pengembangan aplikasi web, fitur-fitur yang diintegrasikan meliputi kemampuan untuk melihat daftar kasus di halaman beranda yang dapat diakses oleh semua pengguna, menambahkan kasus yang hanya dapat diakses oleh wali kelas, serta mengedit kasus yang juga terbatas untuk wali kelas. Karena sistem login belum diterapkan, seluruh pengguna dapat mengakses fitur wali kelas, meskipun nantinya fitur tersebut akan dibatasi hanya untuk wali kelas setelah login diterapkan.