

# Dokumentasi Teknis SIMANJA

## Daftar Isi :

[Arsitektur Sistem](#)

[Spesifikasi Teknologi](#)

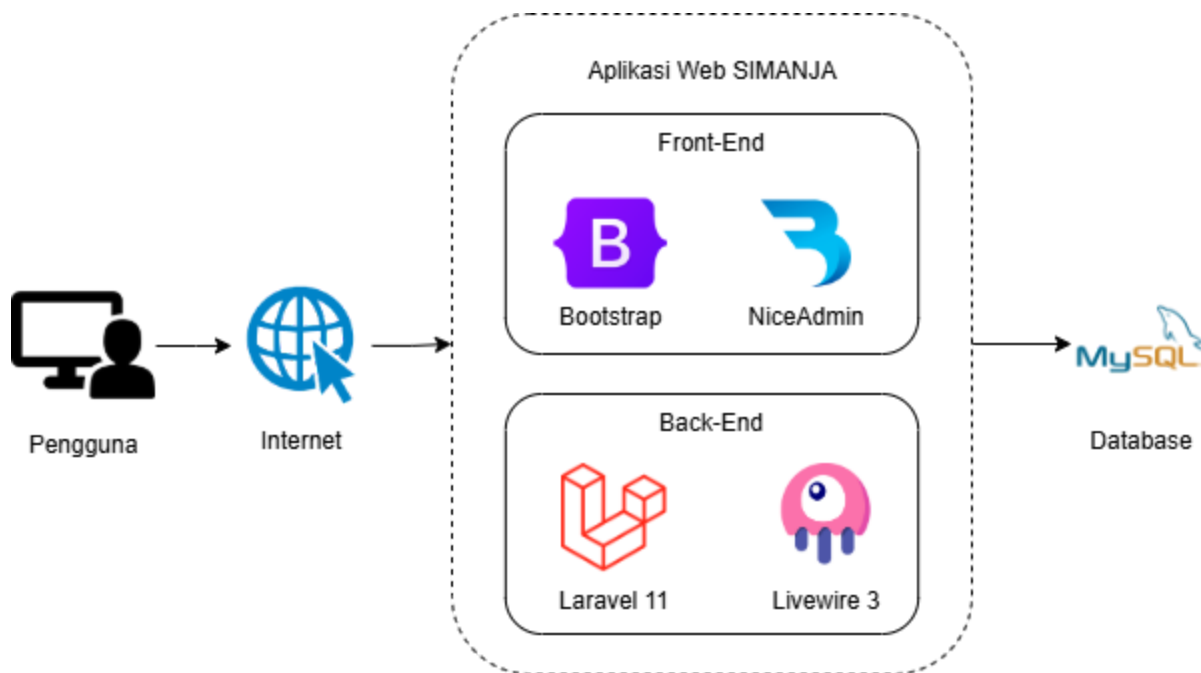
[Struktur Database](#)

[Struktur Folder](#)

[Panduan Deploy](#)

## Arsitektur Sistem

Rancangan arsitektur terdiri dari tiga lapisan utama, yaitu bagian front-end yang terdiri dari HTML dan CSS, back-end yang menggunakan framework PHP, yaitu Laravel dan Livewire, sedangkan data layer menggunakan sistem manajemen basis data MySQL. Rancangan arsitektur sistem dapat dilihat pada Gambar 1.



Rancangan Arsitektur Sistem

## Diagram Rancangan Arsitektur Sistem:

### 1. Frontend (User Interface):

- User interface dibangun berdasarkan template opensource Bootstrapmade yang dapat diakses pada link berikut : <https://bootstrapmade.com/demo/NiceAdmin/>
- Menggunakan Laravel Livewire untuk membuat komponen yang dinamis tanpa perlu memuat ulang halaman (real-time interaksi).
- Komunikasi antara UI dan server dilakukan melalui AJAX yang diatur oleh Livewire.

### 2. Backend (Laravel Application):

- **Route:** Menerima permintaan (request) dari pengguna dan menentukan logika yang akan diproses.
- **Controller:** Mengatur logika bisnis utama, seperti pengolahan data atau pemanggilan layanan.
- **Livewire Components:** Menghubungkan frontend dengan backend melalui event dan properti dinamis.
- **Service Layer (Optional):** Menyediakan pengelompokan fungsi atau logika bisnis kompleks, yang dapat digunakan oleh controller atau Livewire.

### 3. Database (MySQL):

- Database MySQL menyimpan data aplikasi dalam bentuk tabel yang dirancang sesuai kebutuhan (misalnya, tabel pengguna, produk, transaksi, dll.).
- ORM (Eloquent) mempermudah interaksi antara Laravel dan MySQL.

### 4. Data Flow:

- Permintaan dari pengguna di UI dikirimkan ke backend melalui Livewire.
- Backend memproses permintaan (via Route, Controller, Livewire) dan mengakses database menggunakan Eloquent ORM.
- Respon dikirim kembali ke UI untuk ditampilkan secara dinamis.

---

## Spesifikasi Teknologi

### 1. Bahasa Pemrograman

- **Backend:** PHP 8.2
- **FrontEnd:** HTML5, CSS3, JavaScript (dengan Livewire)

## 2. Framework

- **Backend:** Laravel 11.9 (Link Dokumentasi : [Installation - Laravel 11.x - The PHP Framework For Web Artisans](#) )
- **Frontend:** Laravel Livewire 3.4 (Link Dokumentasi : [Quickstart](#) | [Laravel Livewire](#))

## 3. Database

- **Database Management System:** MySQL
- **ORM:** Eloquent ORM (bawaan Laravel)

## 4. Alat Pendukung

- **Web Server:**
  - Apache
  - Built-in Laravel Server untuk pengembangan
- **Lingkungan Pengembangan:**
  - Laragon
  - Visual Studio Code
- **Pengujian:**
  - PHPUnit (untuk pengujian unit)
  - Laravel Dusk (untuk pengujian browser, opsional)

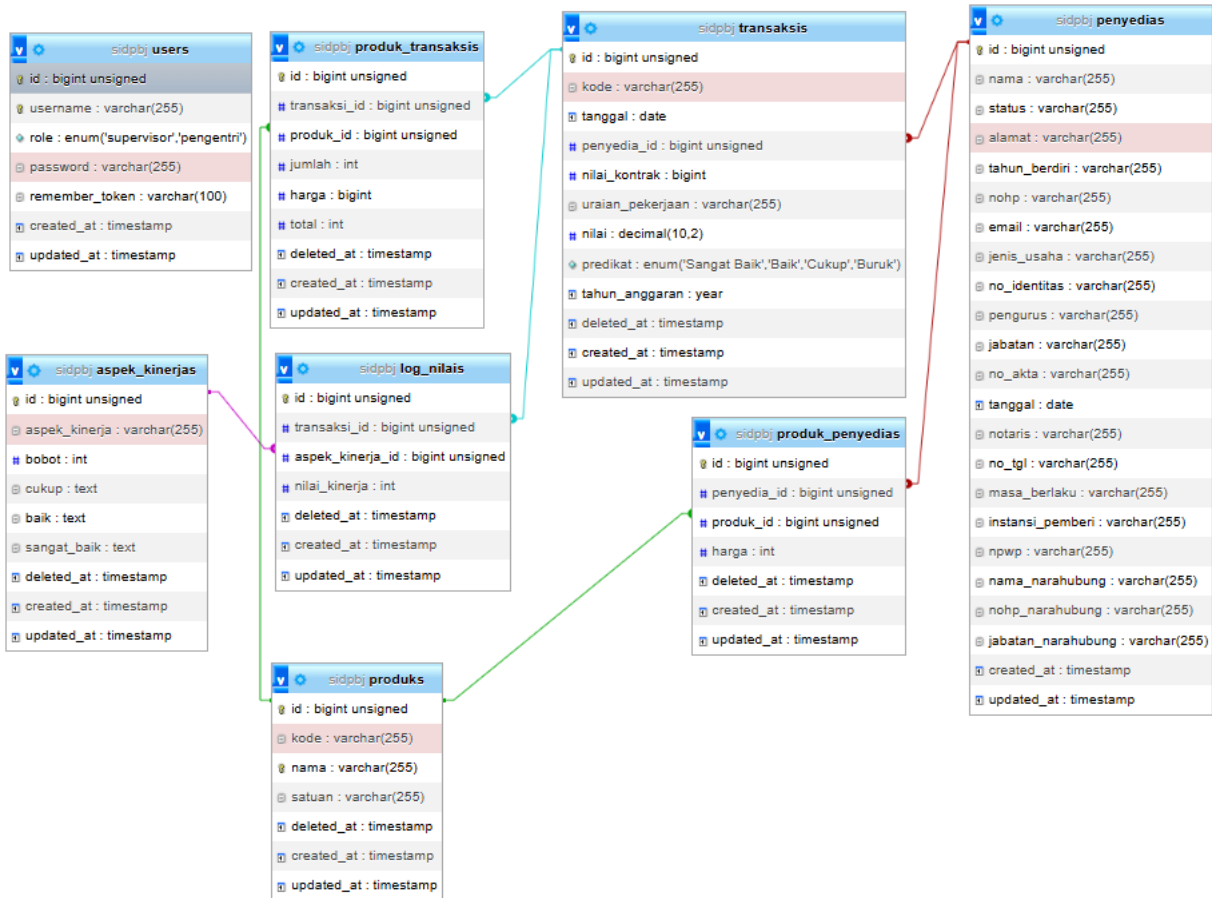
## 7. Sistem Operasi

- **Pengembangan:** Windows 11

---

# Struktur Database

Rancangan basis data mencakup tabel-tabel yang terdiri dari 8 tabel beserta relasi antar tabelnya.



Gambar 2. Entity Relationship Diagram Database

## Struktur Folder

Struktur folder proyek berikut ini menggambarkan bagaimana kode sumber sistem diorganisasikan:

```

/project-root
|
├── app/
|   ├── Http/
|   │   ├── Controllers/
|   │   ├── Livewire/
|   │   └── Middleware/
|   └── Models/

```

# Direktori utama untuk kode sumber

# Controller untuk logika aplikasi

# Komponen Livewire untuk interaksi

# Middleware untuk mengelola permintaan

# Model untuk interaksi dengan database

Providers/	# Service providers untuk men
View/	# View terkait dengan antarmul
config/	# Konfigurasi <b>aplikasi</b> (data
database/	# Migrasi dan seeding untuk l
migrations/	# File migrasi untuk membuat
factories/	# Data dummy untuk pengujian
seeders/	# Seeder untuk memasukkan da
public/	# Berkas <b>publik</b> ( <b>CSS</b> , <b>JS</b> , gar
assets/	# File <b>CSS</b> dan <b>JS</b> , termasuk
storage/	# File upload atau penyimpanan
index.php	# Titik masuk aplikasi
resources/	# Template dan view Blade
views/	# File Blade untuk tampilan
lang/	# File bahasa untuk internasi
routes/	# File untuk mendefinisikan
web.php	# Rute untuk aplikasi web
api.php	# Rute <b>API</b> jika ada
storage/	# Direktori untuk log dan fi
tests/	# Tes otomatis untuk memast
Feature/	# Tes fitur
.env	# File lingkungan untuk <b>kon</b>

### Deskripsi Struktur Folder

- **app/**: Menyimpan logika inti aplikasi seperti controller, model, dan komponen Livewire.
- **config/**: Menyimpan konfigurasi untuk berbagai layanan dan integrasi.
- **database/**: Berisi file migrasi untuk membuat atau mengubah struktur database, beserta data dummy dan seeding.

- **public/**: Berkas yang diakses oleh pengguna langsung, termasuk file template yang digunakan (Bootstrap NiceAdmin).
  - **resources/views/**: Menyimpan file Blade yang digunakan untuk rendering tampilan aplikasi. Termasuk di dalamnya adalah template Bootstrap NiceAdmin.
  - **routes/**: Menyimpan file rute yang digunakan untuk mendefinisikan endpoint aplikasi.
- 

## Panduan Deploy

### Langkah 1: Akses cPanel

#### Login ke cPanel

Masukkan username dan password cPanel yang diberikan oleh penyedia hosting Anda.

### Langkah 2: Akses Terminal (SSH) di cPanel

#### 1. Aktifkan SSH

- Pastikan SSH telah diaktifkan oleh penyedia hosting Anda. Jika belum, hubungi tim support hosting untuk mengaktifkan akses SSH.
- Di cPanel, cari dan pilih **Terminal** (pada bagian "Advanced").
- Jika Terminal tidak tersedia, Anda perlu menggunakan aplikasi SSH seperti **PuTTY** atau terminal Mac/Linux untuk mengakses server.

#### 2. Login ke Terminal

- Akses terminal melalui cPanel atau menggunakan SSH client (misalnya PuTTY).
- Masukkan perintah berikut untuk login ke server:

```
ssh username@yourdomain.com
```

- Gantilah `username` dengan username cPanel dan `yourdomain.com` dengan nama domain atau IP server.

### Langkah 3: Install PHP dan Composer (Jika Belum Terpasang)

#### 1. Periksa Versi PHP

- Pastikan PHP sudah terpasang di server. Jalankan perintah:

```
php -v
```

- Pastikan versi PHP yang terinstal adalah 8.1 atau lebih tinggi (untuk Laravel 11).

## 2. Install Composer (Jika Belum Terpasang)

- Jika Composer belum terinstal, Anda bisa menginstalnya dengan menjalankan perintah berikut:

```
curl -sS https://getcomposer.org/installer | php  
mv composer.phar /usr/local/bin/composer
```

## Langkah 4: Clone Repositori Git ke Server

### 1. Akses Direktori Web

- Arahkan terminal ke direktori di mana Anda ingin menempatkan proyek (misalnya `public_html` untuk akses domain utama).

```
cd public_html
```

### 2. Clone Repositori Git

- Gunakan perintah `git clone` untuk menarik file dari repositori Git Anda:

```
git clone https://github.com/mhdaliffarhan/simanja.git
```

- 

## Langkah 5: Konfigurasi Aplikasi

### 1. Masuk ke Direktori Proyek

- Pindah ke direktori proyek yang baru saja di-clone:

```
cd simanja/kode
```

### 2. Salin File `.env`

- Salin file `.env.example` menjadi `.env` untuk konfigurasi aplikasi:

```
cp .env.example .env
```

### 3. Sesuaikan Konfigurasi Database dan Lainnya

- Edit file `.env` untuk mengonfigurasi koneksi database dan variabel aplikasi lainnya:

```
nano .env
```

- Atur parameter-parameter berikut:
  - `DB_CONNECTION=mysql`
  - `DB_HOST=localhost` atau alamat database Anda
  - `DB_PORT=3306`
  - `DB_DATABASE=nama_database`
  - `DB_USERNAME=nama_user`
  - `DB_PASSWORD=password_user`

## Langkah 6: Install Dependensi dengan Composer

### 1. Install Dependensi Aplikasi

- Jalankan perintah berikut untuk menginstal semua dependensi yang diperlukan oleh Laravel:

```
composer install --optimize-autoloader --no-dev
```

### 2. Generate Application Key

- Laravel membutuhkan key aplikasi untuk enkripsi data. Jalankan perintah:

```
php artisan key:generate
```

## Langkah 7: Migrasi dan Seed Database

- **Migrasi Database**



```
php artisan migrate --seed
```