

# PharmaSys

---

Aplikasi ini adalah proyek yang dibuat untuk membantu salah satu mitra kami dalam mengelola data obat-obatan di apotek mereka. Aplikasi ini dapat digunakan untuk menginput data obat-obatan, mengupdate data obat-obatan, dan menghapus data. Pada aplikasi ini terdapat dua aktor yaitu **admin** dan **kasir** dimana masing-masing aktor memiliki hak akses yang berbeda.

- **Admin** dapat menginput, mengupdate, dan menghapus data.
- **Kasir** hanya dapat menginput data barang yang dibeli.

Aplikasi ini juga dapat menampilkan data obat yang kehabisan stok, paling laris, tanggal expired obat, dan grafik penjualan pada dashboard admin.

---

## Fitur-Fitur

### Sistem Kasir:

- Login/Logout
- Pencatatan pembelian barang
- Pencatatan penjualan barang
- Melihat stok barang
- Melihat produk yang dijual

### Sistem Admin:

- Login/Logout
- Pengelolaan stok barang
- Pengelolaan produk
- Pengelolaan user
- Menambahkan informasi produk
- Menambahkan harga produk
- Melihat rekap penjualan
- Mencetak laporan penjualan
- Mengubah informasi produk (deskripsi, stok, tanggal kadaluarsa, harga, dan lain sebagainya)

### Fitur Lain:

- Menampilkan data obat yang kehabisan stok
  - Menampilkan produk paling laris
  - Menampilkan tanggal expired obat
  - Menampilkan grafik penjualan pada dashboard admin
- 

## Teknologi

- **Laravel** (Backend)

- **Laravel Mix** (Frontend)
- **PostgreSQL** (Database)

## Struktur Proyek

Berikut adalah struktur utama proyek PharmaSys:

```
Tugass-Besar-Kelompok1/
├── app/
│   ├── Console/
│   ├── Events/
│   ├── Exceptions/
│   ├── Helpers/
│   ├── Http/
│   │   ├── Controllers/
│   │   │   ├── Controller.php
│   │   │   └── Admin/
│   │   │       ├── DashboardController.php
│   │   │       ├── ProductController.php
│   │   │       ├── SaleController.php
│   │   │       ├── PurchaseController.php
│   │   │       ├── UserController.php
│   │   │       ├── SupplierController.php
│   │   │       ├── RoleController.php
│   │   │       ├── PermissionController.php
│   │   │       ├── CategoryController.php
│   │   │       ├── BackupController.php
│   │   │       ├── NotificationController.php
│   │   │       ├── SettingController.php
│   │   │       └── Auth/
│   │   └── Middleware/
│   ├── Listeners/
│   ├── Models/
│   │   ├── Category.php
│   │   ├── Product.php
│   │   ├── Purchase.php
│   │   ├── Sale.php
│   │   ├── Supplier.php
│   │   └── User.php
│   ├── Notifications/
│   └── Providers/
├── bootstrap/
├── config/
├── database/
│   ├── factories/
│   ├── migrations/
│   └── seeders/
├── public/
│   ├── css/
│   └── js/
```

```

├── storage/
├── assets/
├── favicon.ico
├── index.php
├── resources/
│   ├── views/
│   │   ├── admin/
│   │   │   ├── dashboard.blade.php
│   │   │   ├── backup.blade.php
│   │   │   ├── settings.blade.php
│   │   │   ├── products/
│   │   │   ├── purchases/
│   │   │   ├── sales/
│   │   │   ├── users/
│   │   │   ├── roles/
│   │   │   ├── suppliers/
│   │   │   ├── auth/
│   │   │   ├── includes/
│   │   │   └── layouts/
│   │   └── welcome.blade.php
│   ├── css/
│   ├── js/
│   └── lang/
├── routes/
│   ├── web.php
│   ├── api.php
│   ├── channels.php
│   └── console.php
├── storage/
├── tests/
│   ├── Feature/
│   ├── Unit/
│   ├── CreatesApplication.php
│   └── TestCase.php
├── .env.example
├── artisan
├── composer.json
├── composer.lock
├── package.json
├── package-lock.json
├── phpunit.xml
├── webpack.mix.js
└── README.md

```

#### Penjelasan singkat:

- **app/** : Berisi logic aplikasi, controller, model, middleware, dsb.
- **app/Http/Controllers/Admin/** : Semua controller utama (produk, user, transaksi, dsb).
- **app/Models/** : Model Eloquent untuk database.
- **database/migrations/** : File migrasi struktur tabel database.
- **resources/views/** : File tampilan (Blade) untuk admin, kasir, dsb.

- **routes/** : Definisi routing aplikasi (web, api, dsb).
- **public/** : File entry point dan asset publik.
- **config/** : File konfigurasi aplikasi.
- **tests/** : Pengujian unit dan fitur.

Struktur di atas dapat berkembang sesuai kebutuhan pengembangan fitur baru.

---

## Instalasi & Menjalankan Aplikasi

### Prasyarat

- PHP >= 7.4
- Composer
- Node.js & npm
- PostgreSQL

### Langkah-langkah

#### 1. Clone repositori ini

```
git clone <repository-url>
cd PharmacyMS-Laravel-3
```

#### 2. Install dependencies

```
composer install
npm install && npm run dev
```

#### 3. Copy file environment

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

#### 4. Konfigurasi database di file **.env**

```
DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432
DB_DATABASE=pharmasys_db
DB_USERNAME=postgres
DB_PASSWORD=postgres
```

#### 5. Generate key

```
php artisan key:generate
```

## 6. Migrasi database

```
php artisan migrate
```

## 7. Jalankan server

```
php artisan serve
```

## 8. Akses aplikasi

Buka browser ke <http://127.0.0.1:8000>

---

# Implementasi Multi-Komputer (Client-Server) dengan Kabel LAN

PharmaSys dapat digunakan secara bersamaan pada dua komputer atau lebih (misal: satu sebagai server, satu sebagai client) dengan koneksi kabel LAN dan satu database PostgreSQL terpusat. Berikut langkah-langkah implementasinya:

### 1. Siapkan Jaringan LAN

- Hubungkan kedua komputer menggunakan kabel LAN (langsung atau melalui switch/router).
- Pastikan kedua komputer berada dalam satu jaringan (misal: IP 192.168.1.2 dan 192.168.1.3).

### 2. Instalasi Database di Server

- Pilih salah satu komputer sebagai server database (PostgreSQL).
- Install PostgreSQL dan buat database **pharmasys\_db**.
- Edit file konfigurasi PostgreSQL (**postgresql.conf** dan **pg\_hba.conf**) agar menerima koneksi dari IP client.
- Pastikan port 5432 terbuka di firewall server.

### 3. Konfigurasi Aplikasi di Setiap Komputer

- Pada file **.env** di masing-masing komputer, atur variabel berikut:

```
DB_CONNECTION=pgsql
DB_HOST=IP_SERVER_DATABASE # contoh: 192.168.1.2
DB_PORT=5432
DB_DATABASE=pharmasys_db
```

```
DB_USERNAME=postgres
DB_PASSWORD=postgres
```

- Ganti **IP\_SERVER\_DATABASE** dengan IP komputer server database.

#### 4. Jalankan Aplikasi

- Lakukan instalasi dan migrasi database seperti biasa di kedua komputer.
- Jalankan aplikasi dengan **php artisan serve** di masing-masing komputer.
- Akses aplikasi melalui browser di masing-masing komputer, misal:
  - Server: <http://192.168.1.2:8000>
  - Client: <http://192.168.1.3:8000>

#### 5. Penggunaan Bersama

- Semua data (produk, transaksi, user, dll) akan tersimpan di satu database pusat dan dapat diakses secara real-time dari kedua komputer.
- Pastikan koneksi jaringan stabil agar aplikasi berjalan lancar.

#### Catatan:

- Jika ingin lebih dari dua komputer, cukup tambahkan komputer ke jaringan LAN dan lakukan konfigurasi **.env** seperti di atas.
- Untuk keamanan, batasi akses database hanya dari IP yang dipercaya.



## Penggunaan

Silakan login sesuai role (admin/kasir) untuk mengakses fitur yang tersedia.

---



## API & Web Endpoints

### Autentikasi & User

- **POST /login**  
Login user (admin/kasir)
- **POST /admin/logout**  
Logout user
- **GET /admin/profile**  
Lihat profil user
- **POST /admin/profile/{user}**  
Update profil user
- **PUT /admin/profile/update-password/{user}**  
Update password user

### Dashboard & Notifikasi

- **GET /admin/dashboard**  
Dashboard ringkasan data
- **GET /admin/notification**  
Tandai notifikasi sebagai dibaca
- **GET /admin/notification-read**  
Lihat notifikasi yang sudah dibaca

## Manajemen User, Role, Permission

- **GET /admin/users**  
List user
- **POST /admin/users**  
Tambah user
- **GET /admin/users/{id}**  
Detail user
- **PUT/PATCH /admin/users/{id}**  
Update user
- **DELETE /admin/users/{id}**  
Hapus user
- **GET /admin/roles**  
List role
- **POST /admin/roles**  
Tambah role
- **GET /admin/roles/{id}**  
Detail role
- **PUT/PATCH /admin/roles/{id}**  
Update role
- **DELETE /admin/roles/{id}**  
Hapus role
- **GET /admin/permissions**  
List permission
- **POST /admin/permissions**  
Tambah permission
- **DELETE /admin/permissions/{id}**  
Hapus permission
- **PUT /admin/permission**  
Update permission

## Manajemen Supplier

- **GET /admin/suppliers**  
List supplier
- **POST /admin/suppliers**  
Tambah supplier
- **GET /admin/suppliers/{id}**  
Detail supplier
- **PUT/PATCH /admin/suppliers/{id}**  
Update supplier
- **DELETE /admin/suppliers/{id}**  
Hapus supplier

## Manajemen Kategori

- **GET /admin/categories**  
List kategori
- **POST /admin/categories**  
Tambah kategori
- **DELETE /admin/categories/{id}**  
Hapus kategori
- **PUT /admin/categories**  
Update kategori

## Manajemen Produk

- **GET /admin/products**  
List produk
- **POST /admin/products**  
Tambah produk
- **PUT/PATCH /admin/products/{id}**  
Update produk
- **DELETE /admin/products/{id}**  
Hapus produk
- **GET /admin/products/outstock**  
Produk yang habis stok
- **GET /admin/products/expired**  
Produk yang kadaluarsa

## Pembelian (Purchase)

- **GET /admin/purchases**  
List pembelian
- **POST /admin/purchases**  
Tambah pembelian
- **PUT/PATCH /admin/purchases/{id}**  
Update pembelian



- **DELETE /admin/purchases/{id}**  
Hapus pembelian
- **GET /admin/purchases/reports**  
Lihat laporan pembelian
- **POST /admin/purchases/reports**  
Generate laporan pembelian

## Penjualan (Sale)

- **GET /admin/sales**  
List penjualan
- **POST /admin/sales**  
Tambah penjualan
- **PUT/PATCH /admin/sales/{id}**  
Update penjualan
- **DELETE /admin/sales/{id}**  
Hapus penjualan
- **GET /admin/sales/reports**  
Lihat laporan penjualan
- **POST /admin/sales/reports**  
Generate laporan penjualan

## Pengaturan & Lain-lain

- **GET /admin/settings**  
Lihat pengaturan
- **POST /admin/settings**  
Simpan pengaturan

---

## Endpoint API (Sanctum)

- **GET /api/user**  
Mendapatkan data user yang sedang login (butuh token Sanctum)

---

### Catatan:

- Semua endpoint **/admin/\*** membutuhkan autentikasi (login).
  - Untuk akses API, gunakan token dari Laravel Sanctum.
  - Endpoint CRUD mengikuti standar Laravel Resource Controller (**index, store, show, update, destroy**).
  - Untuk integrasi mobile/frontend, gunakan endpoint di atas sesuai kebutuhan.
- 

## Tim Pengembang

- **Adam Ibnu Ramadhan** – Dev Ops
- **Muhammad Bagus Setiawan** – Backend Developer

- **Putu Ngurah Semara** – Frontend Developer
- **Raisha Alike Irwandira** – UI/UX Designer
- **Rendy Rifandi Kurnia** – Quality Assurance

---

PROFESSEUR : M.DA ROSBTS SIO BORDEAUX - LYCÉE GUSTAVE EIFFEL