

NAMA : RIVAI
NIM : 20230801290

Dokumentasi Teknis dan Sistem: TeamAnuBot

Nama Proyek	TeamAnuBot: Platform Manajemen & Penyewaan Bot LINE
Versi Dokumen	2.0
Status Dokumen	Final
Tanggal Pembaruan	22 Juli 2025
Disusun Oleh	RIVAI

Riwayat Revisi Dokumen

Versi	Tanggal	Penulis	Deskripsi Perubahan
1.0	Juli 2025	RIVAI	Draf awal dokumen bisnis dan teknis.
1.1	Juli 2025	RIVAI	Penyesuaian dokumen dengan struktur migrasi dan rute aplikasi.
2.0	22 Juli 2025	Gemini AI	Ekspansi dokumen menjadi versi komprehensif dengan detail implementasi, alur kerja, dan spesifikasi teknis mendalam.

DAFTAR ISI

1. Pendahuluan.....	2
1.1. Latar Belakang Proyek.....	2
1.2. Ruang Lingkup Proyek	3
1.2.1. Dalam Ruang Lingkup (In-Scope).....	3
1.2.2. Luar Ruang Lingkup (Out-of-Scope).....	3
1.3. Tujuan & Sasaran Proyek.....	3
1.4. Definisi, Istilah, dan Akronim.....	4
2. Spesifikasi Kebutuhan Perangkat Lunak (SRS).....	4
2.1. Kebutuhan Fungsional	4
2.1.1. Modul Registrasi dan Autentikasi Pengguna (akuns)	4
2.1.2. Modul Katalog dan Pembelian Produk	5
2.1.3. Modul Notifikasi dan Webhook	5
2.1.4. Modul Panel Admin (Filament)	5
2.2. Kebutuhan Non-Fungsional	6
3. Desain Sistem dan Arsitektur.....	6
3.1. Tinjauan Arsitektur.....	6
3.2. Desain Database (Skema Rinci).....	7
3.3. Alur Kerja Kunci (Key Workflows).....	8
4. Spesifikasi API.....	9
4.1. API Internal (Administratif).....	9
4.2. Interaksi dengan API Eksternal.....	10
5. Lingkungan dan Deployment.....	11
5.1. Variabel Lingkungan (.env).....	11
5.2. Proses Deployment (Rekomendasi).....	12

1. Pendahuluan

1.1. Latar Belakang Proyek

Seiring dengan meningkatnya penggunaan aplikasi LINE sebagai platform komunikasi utama bagi bisnis, komunitas, dan individu di Indonesia, kebutuhan akan otomasi layanan pelanggan menjadi krusial. Banyak entitas mengandalkan *bot* untuk menjawab pertanyaan, memberikan informasi, atau bahkan melakukan transaksi. Namun, proses pembuatan, pengelolaan, dan pemeliharaan bot seringkali memerlukan pengetahuan teknis yang mendalam dan memakan waktu. Proses manual seperti aktivasi langganan, pengecekan pembayaran, dan manajemen pengguna menjadi tidak efisien seiring bertambahnya jumlah pelanggan.

TeamAnuBot hadir sebagai solusi untuk menjembatani kesenjangan ini. Proyek ini bertujuan untuk menyediakan sebuah platform berbasis web yang memungkinkan siapa saja, baik dengan maupun tanpa latar belakang teknis, untuk dapat menyewa dan mengelola *self-bot* atau *official bot* LINE dengan mudah. Dengan mengintegrasikan sistem pembayaran modern (Midtrans) dan sistem verifikasi yang familier (WhatsApp OTP), TeamAnuBot menyederhanakan seluruh siklus hidup layanan bot, mulai dari pendaftaran pengguna hingga perpanjangan langganan.

1.2. Ruang Lingkup Proyek

Dokumen ini menguraikan fungsionalitas dan spesifikasi teknis untuk platform TeamAnuBot.

1.2.1. Dalam Ruang Lingkup (In-Scope)

- Pengembangan platform web yang mencakup halaman publik, dasbor pengguna, dan panel admin.
- Sistem registrasi dan autentikasi pengguna akhir (akuns) menggunakan Email, Password, dan verifikasi WhatsApp OTP.
- Sistem autentikasi administrator (users) untuk akses ke Panel Admin Filament.
- Katalog produk untuk layanan penyewaan bot.
- Integrasi penuh dengan **Midtrans** sebagai gerbang pembayaran, termasuk pembuatan transaksi (Snap Token) dan penanganan notifikasi (*webhook*).
- Integrasi dengan **API WhatsApp** eksternal untuk pengiriman pesan OTP dan invoice.
- Pengembangan **API Internal** yang diamankan dengan API Key untuk keperluan administratif.
- Manajemen status langganan (aktif, tidak aktif, kedaluwarsa).
- Fitur ekspor data penyewa bot dari panel admin.

1.2.2. Luar Ruang Lingkup (Out-of-Scope)

- Pengembangan atau kustomisasi fungsionalitas dari bot LINE itu sendiri. Platform hanya menyediakan layanan penyewaan dan manajemen.
- Penyediaan infrastruktur server dan manajemennya (diasumsikan sudah ditangani).
- Layanan dukungan pelanggan langsung melalui chat. Platform bersifat *self-service*.
- Pengembangan aplikasi mobile native. Platform ini berbasis web responsif.

1.3. Tujuan & Sasaran Proyek

- **Tujuan Utama:** Menciptakan platform *self-service* yang andal untuk otomasi penyewaan dan pengelolaan bot LINE.
- **Sasaran Kuantitatif:**
 - Mengurangi waktu pendaftaran dan aktivasi pengguna menjadi di bawah 3 menit.
 - Mencapai tingkat keberhasilan pemrosesan pembayaran otomatis melalui *webhook* sebesar 99.9%.
 - Menyediakan API Internal dengan waktu respons rata-rata di bawah 300ms.
 - Mengurangi intervensi manual admin untuk aktivasi layanan hingga 95%.

1.4. Definisi, Istilah, dan Akronim

Istilah	Definisi
Bot LINE	Program komputer yang beroperasi di dalam aplikasi LINE untuk mengotomatiskan tugas.
Self-bot	Jenis bot yang beroperasi menggunakan akun personal pengguna.
Official Bot	Bot yang terdaftar secara resmi melalui LINE for Business.
OTP	One-Time Password. Kode sandi sekali pakai yang digunakan untuk verifikasi.
Midtrans	Perusahaan gerbang pembayaran (payment gateway) di Indonesia.
Webhook	Mekanisme notifikasi otomatis dari satu sistem ke sistem lain berbasis HTTP POST.
API	Application Programming Interface. Antarmuka yang memungkinkan dua sistem berkomunikasi.
Filament	Perangkat TALL stack untuk membangun panel admin di Laravel.
Livewire	Framework full-stack untuk Laravel yang memungkinkan pembuatan antarmuka dinamis.
CRUD	Create, Read, Update, Delete. Operasi dasar dalam manajemen data.

2. Spesifikasi Kebutuhan Perangkat Lunak (SRS)

2.1. Kebutuhan Fungsional

2.1.1. Modul Registrasi dan Autentikasi Pengguna (akuns)

- **Deskripsi:** Modul ini menangani semua proses yang terkait dengan akun pengguna akhir.
- **Alur Proses Registrasi:**
 1. Pengguna mengakses halaman /register.
 2. Pengguna mengisi nama, email, nomor WhatsApp, dan password.
 3. Sistem memvalidasi input (misalnya, email harus unik).
 4. Sistem memanggil OtpController untuk mengirim 4-6 digit OTP ke nomor WhatsApp yang didaftarkan.
 5. Pengguna diarahkan ke halaman verifikasi OTP.
 6. Pengguna memasukkan OTP. Jika valid, akun di tabel akuns dibuat dan diaktifkan.
- **Alur Proses Login:**
 1. Pengguna mengakses halaman /login.
 2. Pengguna memasukkan email dan password.
 3. Sistem melakukan autentikasi menggunakan *guard* akun.
 4. Jika berhasil, pengguna diarahkan ke dasbor/profil mereka.

- **Aturan Bisnis:**
 - Satu nomor WhatsApp hanya bisa digunakan untuk satu akun.
 - OTP berlaku selama 5 menit.
 - Gagal memasukkan OTP 3 kali akan memblokir sementara permintaan OTP berikutnya.

2.1.2. Modul Katalog dan Pembelian Produk

- **Deskripsi:** Menampilkan produk layanan dan memfasilitasi proses transaksi.
- **Alur Proses:**
 1. Pengguna (baik sudah login maupun belum) dapat melihat daftar produk dari tabel `products`.
 2. Pengguna menekan tombol "Beli" pada produk yang diinginkan.
 3. Sistem mengarahkan ke halaman `/order` atau formulir pemesanan.
 4. Setelah konfirmasi, sistem memanggil `MidtransController` untuk membuat transaksi.
 5. Sebuah *record* baru dibuat di tabel `payment_transactions` dengan status pending.
 6. Backend meminta *Snap Token* ke Midtrans.
 7. Token tersebut dikirim ke frontend, yang kemudian membuka *popup* pembayaran Midtrans.
 8. Pengguna menyelesaikan pembayaran melalui antarmuka Midtrans.

2.1.3. Modul Notifikasi dan Webhook

- **Deskripsi:** Menangani komunikasi otomatis dari sistem eksternal.
- **Alur Proses Webhook Midtrans:**
 1. Midtrans mengirim notifikasi HTTP POST ke *endpoint* `/midtrans/webhook`.
 2. `MidtransController` menerima dan memvalidasi notifikasi tersebut.
 3. Sistem mencari transaksi di `payment_transactions` berdasarkan `midtrans_order_id`.
 4. Status transaksi diperbarui sesuai dengan payload dari Midtrans (misalnya, `settlement`, `expire`, `cancel`).
 5. **Jika `transaction_status` adalah `settlement`:** Sistem akan memicu logika aktivasi layanan. Ini melibatkan pembuatan atau pembaruan *record* di tabel `statuses` dan `data_penyewa_bots`.

2.1.4. Modul Panel Admin (Filament)

- **Deskripsi:** Dasbor terpusat untuk Administrator mengelola seluruh aspek platform.
- **Fungsionalitas:**
 - **Manajemen Pengguna Admin (users):** CRUD untuk akun-akun yang dapat mengakses panel admin.
 - **Manajemen Akun Pelanggan (akuns):** Melihat, mengedit, atau menonaktifkan akun pelanggan.
 - **Manajemen Produk (products):** Menambah, mengubah, atau menghapus layanan bot yang ditawarkan.
 - **Manajemen Transaksi:** Melihat seluruh riwayat `payment_transactions` dan statusnya.
 - **Manajemen Langganan:** Mengelola data di `statuses` dan `data_penyewa_bots`, termasuk penyesuaian manual waktu `habis` jika diperlukan.
 - **Fitur Ekspor:** Terdapat tombol pada manajemen "Data Penyewa Bot" untuk mengekspor data dalam format XLSX.

2.2. Kebutuhan Non-Fungsional

- **Kinerja (Performance):**
 - Waktu muat halaman utama dan katalog produk harus di bawah 2 detik.
 - Waktu respons API untuk proses internal (misalnya, membuat token Midtrans) harus di bawah 500ms.
 - Proses penanganan *webhook* harus selesai dalam waktu 1 detik untuk memastikan respons cepat ke Midtrans.
 - **Keamanan (Security):**
 - Password pengguna (baik users maupun akuns) di-hash menggunakan Bcrypt.
 - Aplikasi dilindungi dari serangan umum seperti SQL Injection dan Cross-Site Scripting (XSS) melalui mekanisme bawaan Laravel (ORM Eloquent dan templating Blade).
 - Akses ke Panel Admin (/admin) dilindungi oleh *middleware* auth dengan *guard* web.
 - Akses ke dasbor pengguna dilindungi oleh *middleware* auth dengan *guard* akun.
 - Akses ke API Internal (/api/everything) diamankan menggunakan API Key statis yang divalidasi via *header* HTTP.
 - **Skalabilitas (Scalability):**
 - Aplikasi dirancang untuk berjalan dalam kontainer Docker, memungkinkan replikasi mudah jika terjadi lonjakan trafik.
 - Sesi dan data *cache* (seperti OTP) disarankan untuk disimpan di Redis agar aplikasi tetap *stateless* dan dapat di-load balance.
 - **Ketersediaan (Availability):**
 - Target ketersediaan layanan adalah 99.8%.
 - Database harus di-backup secara rutin (harian) untuk pemulihan bencana.
-

3. Desain Sistem dan Arsitektur

3.1. Tinjauan Arsitektur

Sistem TeamAnuBot mengadopsi arsitektur monolit modular yang berjalan di atas TALL Stack, dikemas dalam kontainer Docker.

- **Containerization:** Seluruh aplikasi (web server, PHP, database) didefinisikan dalam `docker-compose.yml` untuk konsistensi lingkungan development dan production.
- **Web Server:** Nginx bertindak sebagai *reverse proxy* yang meneruskan permintaan ke layanan PHP-FPM.
- **Backend:** Framework Laravel 12 menjadi inti dari aplikasi, menangani *routing*, logika bisnis, dan interaksi database.
- **Frontend:** Dibuat menggunakan Blade dan Livewire, menghasilkan antarmuka yang dinamis tanpa perlu membangun SPA (Single Page Application) terpisah.
- **Database:** Menggunakan MySQL (atau MariaDB) sebagai sistem manajemen database relasional.
- **Cache:** Redis digunakan untuk menyimpan data yang sering diakses seperti sesi, OTP, dan *caching query*.

3.2. Desain Database (Skema Rinci)

Berikut adalah deskripsi rinci untuk setiap tabel utama berdasarkan file migrasi.

Tabel: users

Tabel ini khusus untuk akun Administrator yang mengakses Panel Admin Filament.

Nama Kolom	Tipe Data	Keterangan
id	bigint, unsigned	Primary Key, Auto-increment
avatar_url	varchar(255)	Nullable. URL ke foto profil admin.
name	varchar(255)	Nama lengkap admin.
email	varchar(255)	Unik. Alamat email untuk login.
password	varchar(255)	Password yang sudah di-hash.
remember_token	varchar(100)	Nullable. Untuk fitur "Remember Me".
timestamps	timestamp	created_at dan updated_at.

Tabel: akuns

Tabel ini untuk akun pengguna akhir (pelanggan).

Nama Kolom	Tipe Data	Keterangan
id	bigint, unsigned	Primary Key, Auto-increment
name	varchar(255)	Nama lengkap pelanggan.
whatsapp_number	varchar(20)	Nomor WhatsApp aktif untuk OTP & notifikasi.
email	varchar(255)	Unik. Alamat email untuk login.
password	varchar(255)	Password yang sudah di-hash.
timestamps	timestamp	created_at dan updated_at.

Tabel: products

Menyimpan daftar layanan yang dapat dibeli.

Nama Kolom	Tipe Data	Keterangan
id	bigint, unsigned	Primary Key, Auto-increment
name	varchar(255)	Nama produk/layanan (e.g., "Sewa SelfBot Bulanan").
description	varchar(255)	Deskripsi singkat produk.
price	decimal(10, 2)	Harga produk dalam IDR.
order	integer	Default 0. Untuk pengurutan tampilan produk.

Tabel: payment_transactions

Mencatat setiap upaya transaksi pembayaran.

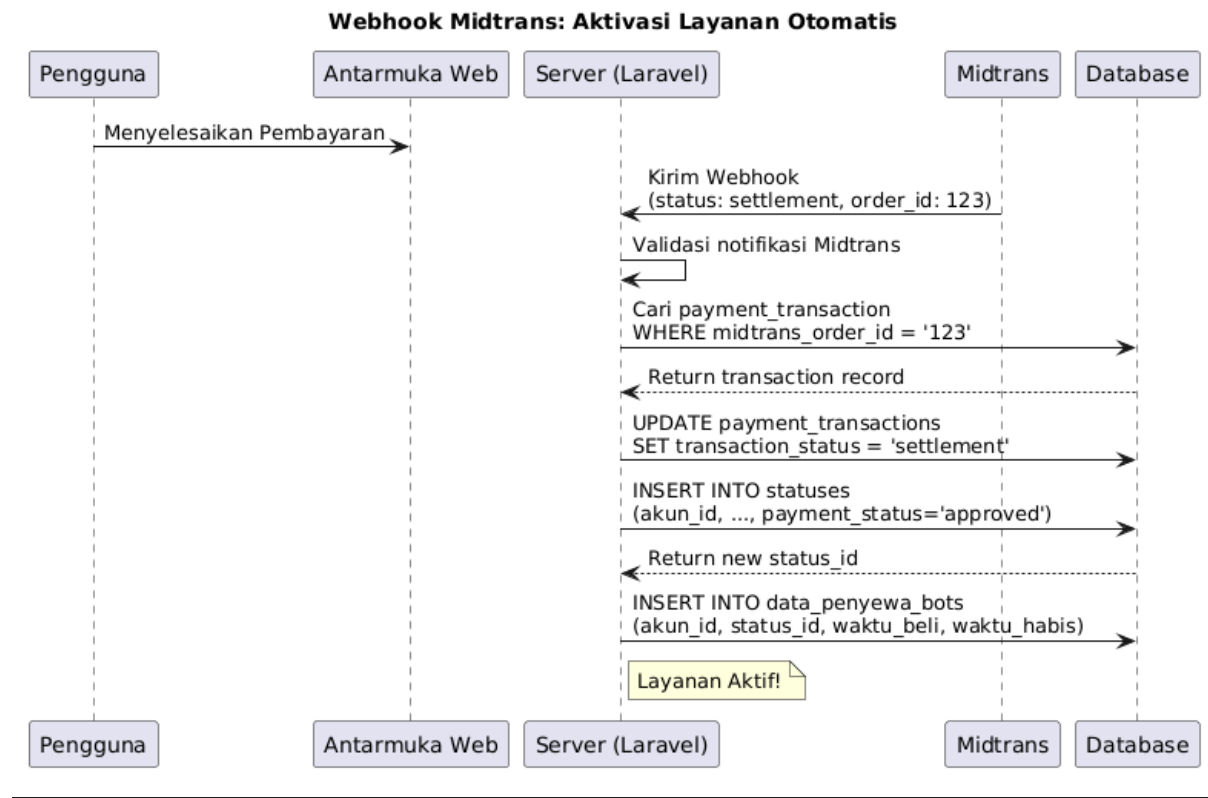
Nama Kolom	Tipe Data	Keterangan
id	bigint, unsigned	Primary Key, Auto-increment
akun_id	bigint, unsigned	Foreign Key ke akuns.id.
product_id	bigint, unsigned	Foreign Key ke products.id.
midtrans_order_id	varchar(255)	ID Pesanan unik yang kita kirim ke Midtrans.
midtrans_transaction_id	varchar(255)	Nullable. ID Transaksi dari Midtrans.
amount	decimal(10, 2)	Jumlah yang harus dibayar.
transaction_status	varchar(255)	Default pending. Status dari Midtrans.
raw_response	text	Nullable. Menyimpan respons JSON lengkap dari webhook Midtrans untuk audit.

Tabel: statuses dan data_penyewa_bots

Kedua tabel ini bekerja sama untuk mengelola status dan durasi langganan aktif. statuses sebagai pemicu awal, dan data_penyewa_bots sebagai catatan historis dan aktif.

3.3. Alur Kerja Kunci (Key Workflows)

Alur Aktivasi Layanan Otomatis



4. Spesifikasi API

4.1. API Internal (Administratif)

API ini bertujuan untuk keperluan internal atau integrasi sistem-ke-sistem di masa depan, bukan untuk dikonsumsi oleh publik.

- **Base URL:** `https://yourdomain.com/api/everything`
- **Autentikasi:** Setiap permintaan harus menyertakan HTTP Header:
 - `Accept: application/json`
 - `x-api-token: rivai123`
- **Struktur Respons Sukses (HTTP 200 OK):**

JSON

```
{
  "success": true,
  "data": [
    { ... }
  ],
  "message": "Data retrieved successfully."
}
```

- **Struktur Respons Gagal (HTTP 4xx/5xx):**

JSON

```
{
  "success": false,
  "message": "Error description.",
  "errors": {
    "field": [ "Error message for this field." ]
  }
}
```

Resource: akuns

- **Endpoint:** `GET /akuns`
 - **Deskripsi:** Mengambil daftar semua akun pelanggan.
 - **Respons Sukses (200):**

JSON

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "name": "Pelanggan Satu",
      "whatsapp_number": "628123456789",
      "email": "pelanggan1@example.com",
      "created_at": "..."
    }
  ],
  "message": "Akuns retrieved successfully."
}
```

- **Endpoint:** `POST /akuns`
 - **Deskripsi:** Membuat akun pelanggan baru.
 - **Request Body:**

JSON

```
{
  "name": "Pelanggan Baru",
  "whatsapp_number": "628987654321",
  "email": "baru@example.com",
}
```

```
"password": "password123",  
"password_confirmation": "password123"  
}
```

○ **Respons Sukses (201 Created):** Objek data dari akun yang baru dibuat.
(Struktur serupa berlaku untuk endpoint *GET /akuns/{id}*, *PUT /akuns/{id}*, dan *DELETE /akuns/{id}*, serta untuk semua resource lainnya: *users*, *products*, *data-penyewa-bots*, *statuses*).

4.2. Interaksi dengan API Eksternal

API WhatsApp

- **Tujuan:** Mengirim pesan OTP dan Invoice.
- **URL Endpoint:** <http://152.42.185.116:3000/send> (dari `WHATSAPP_API_URL`)
- **Metode:** POST
- **Headers:**
 - Content-Type: application/json
 - x-whatsapp-token: rivai123 (dari `API_KEY`)
- **Contoh Body Permintaan (dari backend Laravel):**

JSON

```
{  
  "number": "6281234567890",  
  "message": "Kode OTP Anda adalah: 123456. Jangan berikan kode ini kepada siapapun."  
}
```

API Midtrans

- **Tujuan:** Membuat sesi pembayaran dan menangani notifikasi.
- **Endpoint Pembuatan Token:** <https://app.midtrans.com/snap/v1/transactions> (untuk production)
- **Endpoint Webhook:** <https://yourdomain.com/midtrans/webhook> (URL yang didaftarkan di portal Midtrans).
- **Contoh Payload Webhook yang Ditangani:**

JSON

```
{  
  "transaction_time": "2025-07-22 10:00:00",  
  "transaction_status": "settlement",  
  "transaction_id": "xxx-xxx-xxx",  
  "status_message": "midtrans payment notification",  
  "status_code": "200",  
  "signature_key": "...",  
  "payment_type": "gopay",  
  "order_id": "ORDER-12345",  
  "merchant_id": "...",  
  "gross_amount": "150000.00",  
  "fraud_status": "accept",  
  "currency": "IDR"  
}
```

5. Lingkungan dan Deployment

5.1. Variabel Lingkungan (.env)

File .env adalah pusat konfigurasi aplikasi. Berikut adalah variabel kunci yang harus diatur.

Variabel	Contoh Nilai	Deskripsi
APP_NAME	TeamAnuBot	Nama aplikasi.
APP_ENV	production	Lingkungan aplikasi (development/production).
APP_KEY	base64:...	Kunci enkripsi Laravel (Wajib di-generate).
APP_DEBUG	false	Matikan mode debug di production.
APP_URL	https://yourdomain.com	URL utama aplikasi.
DB_CONNECTION	mysql	Tipe koneksi database.
DB_HOST	127.0.0.1	Host database.
DB_DATABASE	teamanubot	Nama database.
DB_USERNAME	root	Username database.
DB_PASSWORD	password	Password database.
MIDTRANS_SERVER_KEY	SB-Mid-server-...	Kunci server dari portal Midtrans.
MIDTRANS_CLIENT_KEY	SB-Mid-client-...	Kunci klien dari portal Midtrans.
WHATSAPP_API_URL	http://.../send	URL endpoint API WhatsApp.
API_KEY	rivai123	Nilai token untuk header x-whatsapp-token.
API_KEY2	rivai123	Nilai token untuk header x-api-token (Internal API).

5.2. Proses Deployment (Rekomendasi)

1. **Push ke Git:** Pengembang melakukan push kode ke repositori Git (e.g., GitHub, GitLab).
 2. **CI/CD Pipeline:** Sebuah pipeline (e.g., GitHub Actions) terpicu.
 3. **Testing:** Menjalankan tes otomatis (unit, feature test).
 4. **Build Asset:** Menjalankan `npm install` & `npm run build` untuk mengkompilasi aset CSS/JS.
 5. **Build Docker Image:** Membangun image Docker baru dari Dockerfile.
 6. **Push ke Registry:** Mengunggah image Docker ke sebuah *container registry* (e.g., Docker Hub, GHCR).
 7. **Deploy:** Di server production, tarik image terbaru dan restart kontainer aplikasi menggunakan `docker-compose up -d --build`.
 8. **Post-Deploy:** Menjalankan `php artisan migrate --force` untuk menerapkan migrasi database jika ada.
-