

# **Software Requirement Specification (SRS)**



Disusun Oleh:

Nama : Eko Nurcahyo

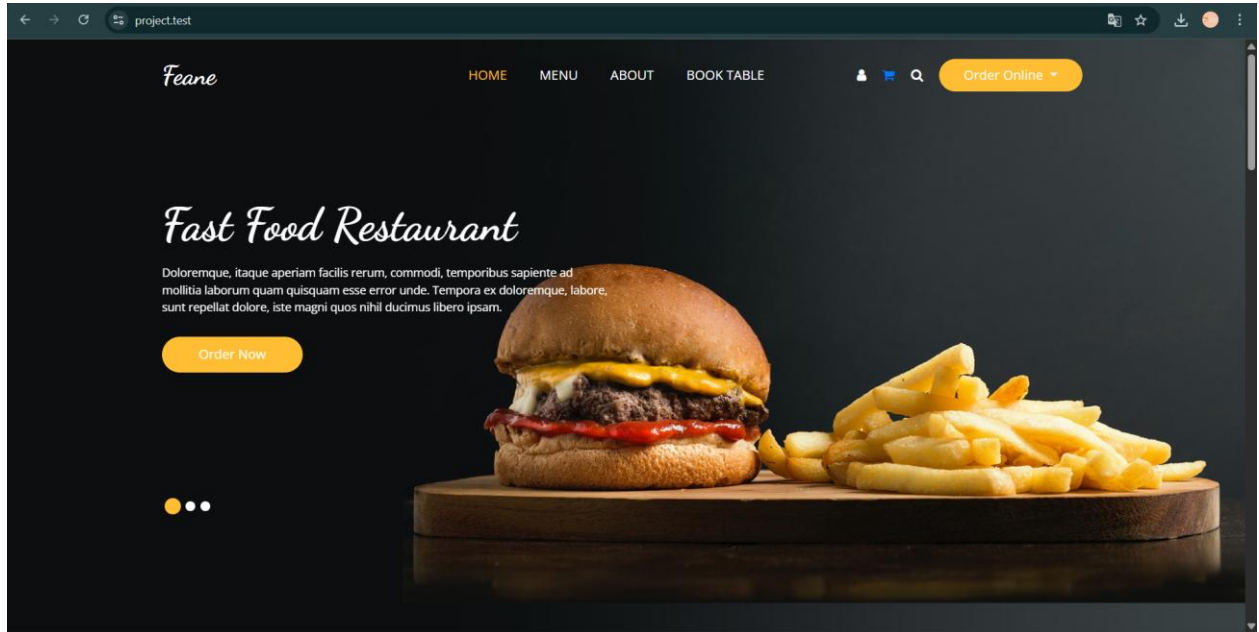
Nim : 20230801169

Prodi : Teknik Informatika

**UNIVERSITAS ESA UNGGUL TANGERANG**

**2025**

# Dokumentasi Implementasi Sistem Kebab Ngajubi



## 1. Struktur Data

Sistem backend Kebab Ngajubi memanfaatkan dua entitas utama dalam database, yaitu Menu dan Order.

### Menu

Digunakan untuk menyimpan informasi produk makanan yang tersedia. Atribut:

- id: Kunci unik tiap menu
- name: Nama menu kebab
- price: Harga satuan
- description: Keterangan tambahan (opsional)
- image: Path gambar menu

## Order

Menyimpan setiap transaksi pemesanan pelanggan. Atribut:

- id: Kunci unik tiap pesan
- name: Nama pelanggan
- phone: Nomor HP pelanggan
- address: Alamat pelanggan
- items: Daftar menu (JSON) berisi detail pesanan
- total: Total pembayaran
- status: Status pesanan (pending, paid, dll)
- timestamps: Waktu pembuatan dan update pesanan

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('menus', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->text('description')->nullable();
18             $table->decimal('price', 8, 2);
19             $table->string('image')->nullable();
20             $table->timestamps();
21         });
22     }
23
24     /**
25      * Reverse the migrations.
26      */
27     public function down(): void
28     {
29         Schema::dropIfExists('menus');
30     }
31 };
32
```

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('orders', function (Blueprint $table) {
15             $table->id();
16             $table->string('name'); // Nama pembeli
17             $table->string('phone'); // No HP
18             $table->string('address'); // Alamat
19             $table->json('items'); // Daftar item (json)
20             $table->decimal('total', 12, 2); // Total harga
21             $table->string('status')->default('pending'); // Status order
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('orders');
32     }
33 };
34
```

## 2. Endpoint API

Dokumentasi API berikut menjelaskan fungsi-fungsi utama backend, baik untuk keperluan frontend web/mobile, maupun pengelolaan data.

- GET /api/menus: Menampilkan seluruh daftar menu kebab.
- GET /api/menus/{id}: Mendapatkan detail dari menu tertentu berdasarkan ID.
- GET /api/cart: Melihat isi keranjang belanja pengguna saat ini (berbasis session atau token).
- POST /api/cart/add: Menambah satu atau beberapa menu ke dalam keranjang.

Contoh Body request:

```
{
  "menu_id": 1,
  "qty": 2
}
```

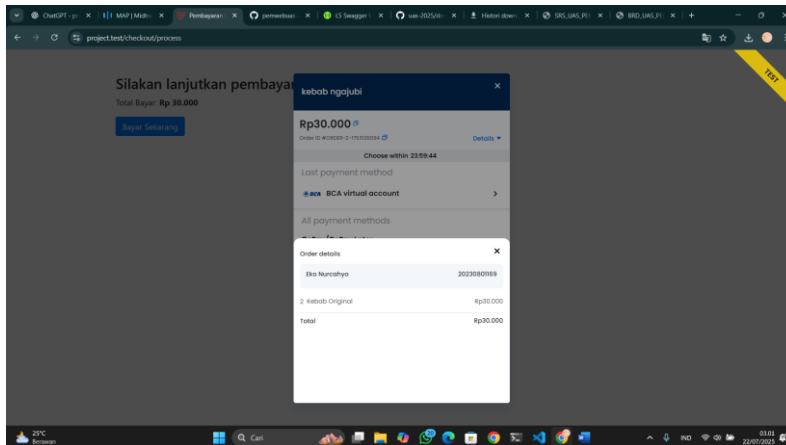
- POST /api/cart/remove: Menghapus menu tertentu dari keranjang.
- POST /api/orders: Membuat pesanan baru sekaligus memulai proses pembayaran. Sistem akan membaca data dari keranjang, memvalidasi ulang harga dari database, lalu menyimpan pesanan ke tabel orders dengan status awal pending. Jika pembayaran berhasil, status otomatis berubah sesuai notifikasi dari payment gateway.
- GET /api/orders: Mengambil daftar seluruh pesanan yang telah dibuat (khusus admin/owner, atau customer via token).

```
src > routes > api.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\Api\MenuController;
5 use App\Http\Controllers\Api\CartController;
6 use App\Http\Controllers\Api\OrderController;
7
8 // ----- MENU API -----
9 // GET /api/menus -> List all menu items
10 // GET /api/menus/{id} -> Show detail menu item
11 Route::get('menus', [MenuController::class, 'index']);
12 Route::get('menus/{id}', [MenuController::class, 'show']);
13
14 // ----- CART API -----
15 // GET /api/cart -> list items in cart (by session/token/user)
16 // POST /api/cart/add -> Add item to cart
17 // POST /api/cart/remove -> Remove item from cart
18 Route::get('cart', [CartController::class, 'index']);
19 Route::post('cart/add', [CartController::class, 'add']);
20 Route::post('cart/remove', [CartController::class, 'remove']);
21
22 // ----- ORDER API -----
23 // POST /api/orders -> Place new order
24 // GET /api/orders -> list all orders (or order history)
25 Route::post('orders', [OrderController::class, 'store']);
26 Route::get('orders', [OrderController::class, 'index']);
27
```

### 3. Integrasi Payment Gateway

#### - Midtrans Snap

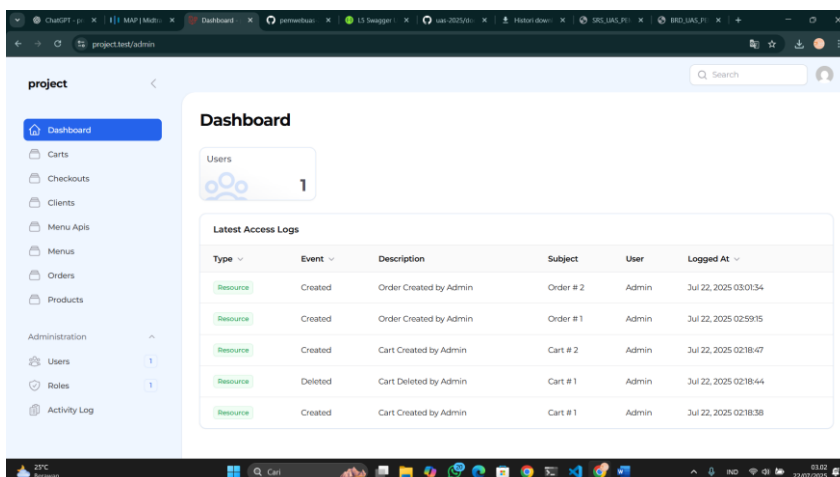
Sistem terintegrasi dengan Midtrans menggunakan Snap API. Saat checkout, backend menghasilkan Snap Token yang diberikan ke frontend. Pelanggan dapat membayar dengan GoPay, ShopeePay, QRIS, dan metode lain yang tersedia. Status pesanan akan diupdate otomatis setelah pembayaran selesai (via webhook/callback).



### 4. Panel Admin

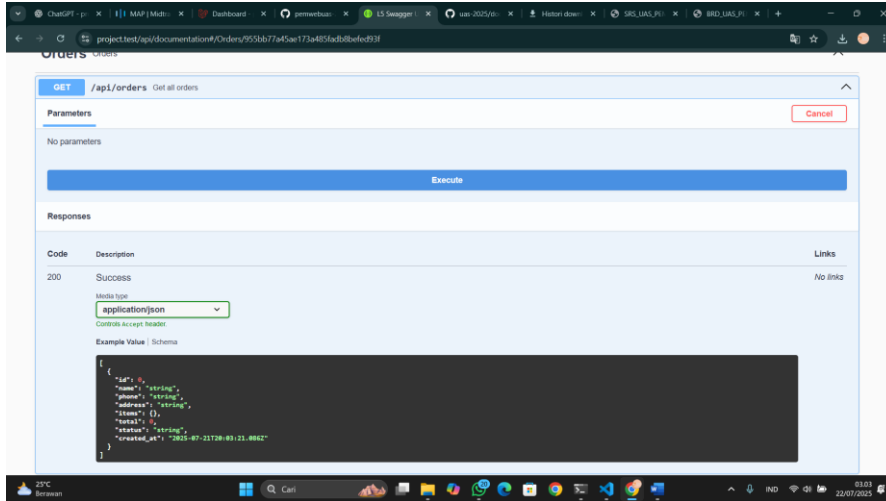
Panel backend dikembangkan menggunakan Filament, framework admin modern untuk Laravel. Fitur utama:

- CRUD Menu (tambah, edit, hapus, lihat menu)
- CRUD Order (pantau, update status, hapus pesanan)
- Security: akses admin dashboard hanya untuk user admin
- Tampilan user-friendly, responsif, dan siap diakses lewat browser



## 5. Dokumentasi API

Semua endpoint otomatis terdokumentasi melalui anotasi PHP (OpenAPI) dan dapat diakses pada `/api/documentation`. Dokumentasi ini memberikan gambaran lengkap endpoint, struktur request/response, serta memudahkan integrasi pihak ketiga.



## 6. Keamanan Sistem

- CORS (Cross-Origin Resource Sharing): Sudah dikonfigurasi agar API dapat diakses frontend/web tools pada tahap pengembangan. Pada lingkungan produksi, pengaturan CORS dapat diperketat sesuai kebutuhan keamanan.
- Role-based Access: Panel admin hanya dapat diakses oleh user dengan hak akses admin (bukan public/customer), sehingga seluruh data operasional tetap terjaga privasinya.

## 7. Workflow Penggunaan Sistem

1. Pelanggan membuka website, memilih menu, menambah ke keranjang.
2. Setelah selesai, pelanggan melakukan checkout dan mengisi data diri.
3. Sistem backend memproses order, menghitung total harga, dan memulai proses pembayaran digital.
4. Jika pembayaran berhasil, status order otomatis berubah, dan admin dapat memantau status serta histori pesanan di dashboard.

## 8. Hasil Uji Implementasi

- Semua proses utama (pemesanan, pembayaran, CRUD menu/order) berjalan dengan lancar.
- Panel admin sangat membantu dalam monitoring operasional tanpa perlu akses langsung ke database.
- Dokumentasi API memudahkan pengembangan frontend maupun integrasi eksternal.
- Struktur sistem siap dikembangkan lebih lanjut, misal untuk penambahan fitur analitik, laporan penjualan, atau modul notifikasi.

## **Penutup**

Dokumentasi ini dapat dijadikan referensi dalam pengembangan lebih lanjut, troubleshooting, maupun onboarding anggota tim baru ke dalam proyek. Seluruh source code telah disusun dengan best practice Laravel, sehingga maintainable dan scalable untuk kebutuhan bisnis Kebab Ngajubi ke depan.