

DOKUMENTASI APLIKASI FOOD ORDER



Di susun oleh :

Nama : An akhfriza hamdani

Jurusan : Pemrograman web

PESANTREN TEKNOLOGI INFORMASI DAN KOMUNIKASI
Creates Future Skilled Professionals
2026

Pendahuluan

Perkembangan teknologi digital mendorong perubahan dalam cara masyarakat memesan makanan menjadi lebih praktis dan efisien. Melalui aplikasi berbasis web, proses pemesanan dapat dilakukan dengan cepat tanpa harus datang langsung ke tempat.

Project **Food Order App** ini dikembangkan untuk mempermudah pengguna dalam melihat menu, memilih makanan berdasarkan kategori, serta melakukan pemesanan hingga pembayaran secara online. Selain itu, sistem ini juga menyediakan fitur bagi admin untuk mengelola data makanan, kategori, dan pesanan.

Aplikasi ini dibangun menggunakan konsep fullstack, yang mengintegrasikan frontend, backend, dan database agar sistem dapat berjalan secara terstruktur dan saling terhubung. Diharapkan, project ini dapat menjadi implementasi nyata dari pengembangan aplikasi web serta memberikan pengalaman pemesanan yang mudah bagi pengguna.

A. Penggunaan Aplikasi

1. Tentang aplikasi

Aplikasi Nom Eat adalah platform pemesanan makanan berbasis web (Food Order App) yang memisahkan antara antarmuka Pelanggan (Customer) dan Pengelola (Admin). Pelanggan dapat melihat menu, memasukkan ke keranjang, dan melakukan pemesanan. Admin dapat mengelola menu, pengguna, dan pesanan.

2. Kebutuhan sistem

- Node.js (v16 atau lebih baru)
- NPM atau Yarn
- Web Browser modern (Chrome, Firefox, Safari, Edge)

3. Cara Instalasi dan menjalankan aplikasi

- Buka terminal atau command prompt.
- Arahkan ke direktori project front-end: ``cd food-order-app``
- Jalankan perintah instalasi dependensi: ``npm install``
- Jalankan server pengembangan lokal: ``npm run dev``
- Buka link localhost yang muncul (biasanya `http://localhost:5173`) di browser.

4. Alur penggunaan singkat

- Akses Halaman Utama (Landing Page)
Pengguna disambut dengan tampilan awal aplikasi.
- Registrasi & Login
 - Pelanggan baru harus mendaftar (Sign Up) terlebih dahulu.
 - Setelah sukses, login dengan username dan password.
- Memesan Makanan
 - Masuk ke halaman Menu.
 - Klik "Tambah ke Keranjang" pada makanan yang diinginkan.
 - Buka ikon Keranjang (Cart) di pojok kanan atas, lalu klik "Checkout".
- Mengelola Sistem (Khusus Admin)
 - Login menggunakan akun dengan role 'admin'.
 - Klik profil di kanan atas, pilih "Admin Dashboard".
 - Di sini Admin bisa mengatur seluruh data melalui sidebar yang tersedia.

B. FITUR APLIKASI

1. Teknologi yang di gunakan

- Frontend Framework: React.js (v19)
- Build Tool: Vite
- Routing: React Router Dom (v7)
- Styling: React Bootstrap & CSS Vanilla (CSS Modules/Global)
- HTTP Client: Axios
- Notifikasi & Alert: React Toastify & Custom Bootstrap Modals
- Charts/Grafik: Recharts
- Icons: React Icons (Feather Icons & Ionicons)

2. Struktur direktori utama (src)

- /Components : Berisi komponen UI yang dapat digunakan kembali (Navbar, Footer, Sidebar, ConfirmModal, dll).
- /Context : Mengelola state global aplikasi (AuthContext untuk login, CartContext untuk keranjang belanja).
- /Pages : Berisi komponen halaman spesifik (LandingPage, LoginPage, MenuPage, dll).
- /Pages/Admin : Berisi halaman khusus role admin (Dashboard, Foods, Categories, Users, Orders).
- /Utils : Utilitas bantuan seperti `axiosInstance.jsx` untuk mengatur base URL dan interceptor token.

3. Arsitektur dan state management

Aplikasi ini menghindari kerumitan Redux dan menggunakan Context API bawaan React untuk state global yang sifatnya ringan:

- AuthContext: Menyimpan data token di LocalStorage, decode JWT payload, dan menyediakan fungsi login/logout ke seluruh komponen.
- CartContext: Menyimpan array item keranjang dan menghitung total harga/jumlah secara dinamis.

4. Sistem roiting dan proteksi keamanan

Terdapat dua jenis proteksi rute pada aplikasi:

- `<ProtectedRoute>` : Membungkus rute yang hanya boleh diakses oleh user yang sudah login (misal: /checkout, /orders). Jika belum login, user dilempar ke /login.
- `<AdminRoute>` : Membungkus rute yang hanya boleh diakses oleh akun dengan role 'admin'. Rute ini akan mengecek token payload. Jika bukan admin, user akan dicegah mengakses /admin dan dialihkan ke beranda.

5. Integrasi backend (API)

- Komunikasi dengan backend dilakukan melalui `axiosInstance` yang otomatis menyematkan header `Authorization: Bearer <token>` pada setiap request.
- Hal ini mencegah repetisi penulisan header secara manual saat melakukan CRUD ke tabel User, Food, Kategori, maupun Orders.

6. Desain responsif

- Aplikasi ini menggunakan Grid System dari Bootstrap yang di kombinasikan dengan CSS Media Queries.
- Contoh nyata: Sidebar Admin pada layar mobile (< 991px) akan bergeser keluar layar (TranslateX: -100%) dan hanya akan muncul jika menu hamburger ditekan (dengan background overlay gelap).

7. Penanganan error & konfirmasi (UX)

- Semua aksi destruktif (Logout, Hapus Data) menggunakan Custom ConfirmModal (berbasis React-Bootstrap Modal) yang lebih modern dan menyatu dengan tema aplikasi.
- Success & Error API Response ditangkap (catch) dan ditampilkan kepada pengguna melalui pop-up toast ringan (React-Toastify).

C. FITUR PELANGGAN (CUSTOMER)

1. Autentikasi

- Register: Pengguna dapat membuat akun baru dengan memasukkan username, email, dan password.
- Login: Pengguna terdaftar dapat masuk ke sistem.
- Logout: Keluar dari sesi dengan aman. (Dilengkapi konfirmasi).

2. Halaman Menu Utama

- Lihat Menu: Menampilkan daftar seluruh makanan yang tersedia lengkap dengan foto, harga, dan kategori.
- Detail Makanan: Mengklik menu akan menampilkan informasi spesifik makanan.

3. Keranjang Belanja (Cart)

- Add to Cart: Menambah item ke keranjang.
- Update Quantity: Menambah/mengurangi jumlah porsi per item di keranjang.
- Remove Item: Menghapus makanan dari keranjang.

4. Checkout & Pesanan

- Checkout: Mengkonfirmasi daftar belanjaan dan menghitung total harga.
- Riwayat Pesanan: Melihat status pesanan (Pending, Diproses, Selesai, Dibatalkan) di halaman My Orders.

D. FITUR PENGELOLA (ADMIN)

1. Dashboard Analytics

- Menampilkan total Makanan, Pengguna, Pesanan, dan Pendapatan.
- Bar Chart: Grafik pendapatan dan jumlah pesanan per hari.
- Pie Chart: Persentase pembagian status pesanan saat ini.

2. Manajemen Makanan (Manage Foods)

- Read: Melihat tabel menu makanan lengkap.

- Create: Menambah menu baru (Nama, harga, kategori, deskripsi, foto).
- Update: Mengedit menu yang sudah ada.
- Delete: Menghapus menu (dengan konfirmasi keamanan).

3. Manajemen Kategori (Manage Categories)

- Mengelola pengelompokan makanan (misal: Makanan Berat, Minuman, Snack).

4. Manajemen Pesanan (Customer Orders)

- Admin dapat melihat semua pesanan yang masuk.
- Admin dapat memperbarui status pesanan secara bertahap:
Pending -> Diproses -> Selesai.
- Admin juga dapat membatalkan pesanan (Dibatalkan).

5. Manajemen Pengguna (Manage Users)

- Melihat daftar seluruh pelanggan dan sesama admin.
- Menambah pengguna baru.
- Mengubah profil atau password pengguna.
- Menghapus pengguna (Admin tidak dapat menghapus dirinya sendiri di baris tabel yang sama).

Penutup

Project **Food Order App** ini merupakan implementasi dari pengembangan aplikasi berbasis web yang mengintegrasikan frontend, backend, dan database dalam satu sistem yang utuh. Melalui aplikasi ini, proses pemesanan makanan dapat dilakukan dengan lebih mudah, cepat, dan terstruktur.

Selama proses pengembangan, berbagai tantangan seperti pengelolaan data, integrasi API, serta penanganan error menjadi bagian dari pembelajaran yang berharga. Hal ini membantu dalam memahami alur kerja sistem secara lebih mendalam, khususnya dalam membangun aplikasi yang siap digunakan.

Diharapkan, project ini dapat menjadi dasar untuk pengembangan lebih lanjut, baik dari segi fitur, performa, maupun keamanan, sehingga dapat mendekati standar aplikasi yang digunakan di dunia industri.