Penjelasan Terkait Yang ada Di Folder

1. **Dependencies yang digunakan**

**Dependencies Utama**

**1. bcryptjs (^3.0.2)**

* Library untuk hashing password secara aman
* Digunakan untuk mengenkripsi password user sebelum disimpan ke database
* Mencegah password tersimpan dalam bentuk plain text

**2. body-parser (^2.2.0)**

* Middleware untuk parsing body request HTTP
* Memungkinkan Express.js membaca data JSON, URL-encoded, dan form data dari request

**3. cors (^2.8.5)**

* Cross-Origin Resource Sharing middleware
* Memungkinkan frontend dari domain berbeda mengakses API backend
* Mengatasi kebijakan same-origin browser

**4. dotenv (^17.2.1)**

* Library untuk memuat environment variables dari file .env
* Menyimpan konfigurasi sensitif seperti database credentials, JWT secret, dll

**5. express (^5.1.0)**

* Web framework utama untuk Node.js
* Digunakan untuk membuat REST API, routing, middleware, dll

**6. express-validator (^7.2.1)**

* Middleware untuk validasi dan sanitasi input data
* Memvalidasi request body, params, dan query sebelum diproses

**7. helmet (^8.1.0)**

* Security middleware untuk Express.js
* Mengatur HTTP headers untuk keamanan (XSS protection, content security policy, dll)

**8. jsonwebtoken (^9.0.2)**

* Library untuk membuat dan memverifikasi JSON Web Tokens (JWT)
* Digunakan untuk sistem authentication dan authorization

**9. morgan (^1.10.1)**

* HTTP request logger middleware
* Mencatat semua request yang masuk ke server untuk monitoring dan debugging

**10. multer (^2.0.2)**

* Middleware untuk handling multipart/form-data
* Digunakan untuk upload file (gambar produk, avatar user, dll)

**11. mysql2 (^3.14.3)**

* MySQL client untuk Node.js
* Database driver untuk koneksi ke MySQL database

**12. nodemailer (^7.0.5)**

* Library untuk mengirim email
* Biasanya digunakan untuk reset password, konfirmasi email, notifikasi, dll

**13. sequelize (^6.37.7)**

* ORM (Object-Relational Mapping) untuk Node.js
* Mempermudah operasi database dengan syntax JavaScript object
* Mendukung migrations, validations, associations, dll

**14. socket.io (^4.8.1)**

* Library untuk real-time communication
* Memungkinkan fitur chat, notifikasi real-time, live updates

**Dev Dependencies**

**15. nodemon (^3.1.10)**

* Development tool yang secara otomatis restart server saat ada perubahan code
* Hanya digunakan saat development, tidak di production

**Kesimpulan**

Dependencies ini membentuk stack backend e-commerce yang lengkap dengan fitur:

* Authentication & Authorization (JWT, bcryptjs)
* Database ORM (Sequelize, MySQL2)
* File Upload (Multer)
* Email Service (Nodemailer)
* Real-time Features (Socket.io)
* Security (Helmet, CORS)
* Validation (Express-validator)
* Logging & Monitoring (Morgan)

1. **Struktur Folder**

## 📁 Struktur Folder Backend

### 🔧 **Config**

* database.js - Konfigurasi koneksi Sequelize ke MySQL database

### 🎮 **Controllers**

Folder ini berisi logic bisnis untuk setiap fitur:

* authController.js - Login & register user
* cartController.js - Manajemen keranjang belanja
* categoryController.js - CRUD kategori produk
* complaintController.js - Sistem komplain customer
* messageController.js - Live chat/messaging
* newsController.js - Product news/highlights
* orderController.js - Manajemen pesanan
* productController.js - CRUD produk dengan upload gambar
* reviewController.js - Review & rating produk
* transactionController.js - Payment transactions
* variantController.js - Varian produk (warna, ukuran)

### 🛡️ **Middlewares**

* auth.js - JWT authentication & authorization
* upload.js - Multer untuk upload gambar produk

### 🗄️ **Models** (Sequelize ORM)

Definisi struktur database:

* user.js - Model user (admin & customer)
* product.js - Model produk utama
* product\_category.js - Kategori produk
* product\_image.js - Gambar produk
* product\_variant.js - Varian produk
* product\_news.js - Product highlights
* cart.js & cart\_item.js - Keranjang belanja
* order.js & order\_item.js - Pesanan
* review.js - Review produk
* transaction.js - Transaksi pembayaran
* complaint.js - Komplain customer
* message.js - Live chat messages
* index.js - **Associations/Relasi antar model**

### 🛣️ **Routes**

API endpoints untuk setiap fitur:

* authRoutes.js - /api/auth/\*
* productRoutes.js - /api/products/\*
* categoryRoutes.js - /api/categories/\*
* cartRoutes.js - /api/cart/\*
* orderRoutes.js - /api/orders/\*
* reviewRoutes.js - /api/reviews/\*
* transactionRoutes.js - /api/transactions/\*
* messageRoutes.js - /api/messages/\*
* complaintRoutes.js - /api/complaints/\*
* newsRoutes.js - /api/news/\*
* variantsRoutes.js - /api/variants/\*

### 📁 **Uploads**

Folder untuk menyimpan file upload (gambar produk, dll)

### 📄 **Root Files**

* [app.js](vscode-file://vscode-app/c:/Program%20Files/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) - **Main application file** dengan Express server setup
* [package.json](vscode-file://vscode-app/c:/Program%20Files/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) - Dependencies & project configuration
* .env - Environment variables (database config, JWT secret)
* [README.md](vscode-file://vscode-app/c:/Program%20Files/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) - Dokumentasi API endpoints
* [.gitignore](vscode-file://vscode-app/c:/Program%20Files/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) - Files to ignore in git

## 🏗️ ****Arsitektur Backend****

### **Tech Stack:**

* **Framework:** Express.js 5.1.0
* **Database:** MySQL dengan Sequelize ORM
* **Authentication:** JWT (JSON Web Tokens)
* **File Upload:** Multer
* **Security:** Helmet, CORS, bcryptjs
* **Validation:** Express-validator
* **Real-time:** Socket.io (untuk live chat)

### **Fitur Utama:**

1. **🔐 Authentication & Authorization**
   * Register/Login dengan JWT
   * Role-based access (admin/customer)
2. **🛍️ E-commerce Core**
   * Product management dengan kategori & varian
   * Shopping cart & checkout
   * Order management dengan status tracking
3. **💳 Payment & Transactions**
   * Transaction recording
   * Payment gateway integration ready
4. **⭐ Customer Features**
   * Product reviews & ratings
   * Complaint system
   * Live chat messaging
5. **📊 Admin Features**
   * Product & category management
   * Order status updates
   * Customer support via complaints & messages
6. **🖼️ Media Management**
   * Product image upload
   * Static file serving with CORS

### **Database Relations (dari**[**index.js**](vscode-file://vscode-app/c:/Program%20Files/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)**):**

* User ↔ Orders, Cart, Reviews, Complaints, Messages
* Product ↔ Categories, Images, Variants, Order Items, Cart Items
* Orders ↔ Order Items, Transactions
* Kompleks many-to-many relationships untuk e-commerce

1. **Folder config** 
   * + **File database.js**

* *// Konfigurasi koneksi database menggunakan Sequelize*
* *const* { Sequelize } = require("sequelize");
* *// Menggunakan dotenv untuk mengelola variabel lingkungan*
* require("dotenv").config();
* *const* sequelize = new Sequelize(
* process.env.DB\_NAME, *// Nama database*
* process.env.DB\_USER, *// Username database*
* process.env.DB\_PASS, *// Password database*
* {
* *host:* process.env.DB\_HOST, *// Host database*
* *dialect:* "mysql", *// Jenis database yang digunakan*
* *logging:* false, *// Menonaktifkan log SQL queries*
* }
* );
* module.exports = sequelize;

1. **Gambaran Database**

**1. Tabel products (Produk Utama)**

Tabel ini akan menyimpan informasi detail tentang setiap produk. Penyesuaian dari product awal Anda:

product\_id: INT (Primary Key, Auto Increment) - ID unik untuk setiap produk.

category\_id: INT (Foreign Key ke product\_categories.category\_id) - Menghubungkan produk dengan kategori.

product\_name: VARCHAR(255) - Judul/nama produk (misal: "JKT48 Official T-Shirts - [Nama Acara/Member]").

description: TEXT - Deskripsi lengkap produk.

price: DECIMAL(10, 2) - Harga produk.

stock: INT - Jumlah stok produk yang tersedia.

sold\_quantity: INT (Default 0) - Jumlah produk yang sudah terjual.

average\_rating: DECIMAL(2, 1) (Default 0.0) - Rata-rata rating produk.

total\_reviews: INT (Default 0) - Jumlah total ulasan/komentar.

main\_image: VARCHAR(255) - Path atau URL gambar utama produk.

created\_at: TIMESTAMP (Default CURRENT\_TIMESTAMP) - Waktu produk ditambahkan.

updated\_at: TIMESTAMP (Default CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP) - Waktu terakhir produk diperbarui.

**2. Tabel product\_images (Galeri Gambar Produk)**

Untuk menampung banyak gambar detail produk:

image\_id: INT (Primary Key, Auto Increment)

product\_id: INT (Foreign Key ke products.product\_id) - Menghubungkan gambar ke produk.

image\_path: VARCHAR(255) - Path atau URL gambar detail.

alt\_text: VARCHAR(255) - Teks alternatif untuk gambar (penting untuk SEO dan aksesibilitas).

**3. Tabel product\_variants (Variasi Produk: Warna & Ukuran)**

Ini lebih baik daripada menempatkan colors\_product dan size\_product langsung di tabel products karena satu produk bisa punya banyak variasi.

variant\_id: INT (Primary Key, Auto Increment)

product\_id: INT (Foreign Key ke products.product\_id)

color: VARCHAR(50) (Nullable jika tidak semua produk punya variasi warna)

size: VARCHAR(50) (Nullable jika tidak semua produk punya variasi ukuran, contoh: S, M, L, XL, ALL SIZE)

variant\_stock: INT - Stok spesifik untuk varian ini.

**4. Tabel product\_categories (Kategori Produk)**

Dari category\_product Anda, kita akan menyederhanakannya dan menjadikannya tabel yang berdiri sendiri untuk relasi yang lebih baik.

category\_id: INT (Primary Key, Auto Increment)

category\_name: VARCHAR(100) (Unique) - Nama kategori (contoh: "JKT48 Official T-Shirts", "Photobook").

slug: VARCHAR(100) (Unique) - Versi "URL-friendly" dari nama kategori (misal: "jkt48-official-tshirts").

Daftar kategori yang Anda berikan akan menjadi data awal di tabel ini:

JKT48 Official T-Shirts

Birthday T-Shirts

JKT48 Official Birthday T-Shirts

Photobook

JKT48 Official Photobook Stationery

Accessories

JKT48 Accessories

**5. Tabel product\_news (Produk Terbaru/Sorotan)**

Ini adalah perbaikan dari product\_terbaru Anda, untuk menyimpan produk yang ingin disorot.

news\_id: INT (Primary Key, Auto Increment) - Dari id\_news\_product.

product\_id: INT (Foreign Key ke products.product\_id) - Produk yang sedang disorot.

image\_highlight: VARCHAR(255) - Path atau URL gambar sorotan (dari img).

highlight\_link: VARCHAR(255) - Link untuk sorotan ini (dari link).

alt\_text: VARCHAR(255) - Teks alternatif untuk gambar sorotan (dari alt).

display\_order: INT (Opsional) - Urutan tampil di halaman.

is\_active: BOOLEAN (Default TRUE) - Untuk mengaktifkan/menonaktifkan sorotan.

**6. Tabel users (Pengguna: Admin & Konsumen)**

Untuk menangani login dan registrasi.

user\_id: INT (Primary Key, Auto Increment)

full\_name: VARCHAR(255)

email: VARCHAR(255) (Unique)

password\_hash: VARCHAR(255) (Untuk menyimpan password yang di-hash)

phone\_number: VARCHAR(20) (Nullable)

address: TEXT (Nullable)

city: VARCHAR(100) (Nullable)

postal\_code: VARCHAR(10) (Nullable)

role: ENUM('admin', 'customer') - Untuk membedakan hak akses.

created\_at: TIMESTAMP

updated\_at: TIMESTAMP

**7. Tabel orders (Pesanan)**

order\_id: INT (Primary Key, Auto Increment)

user\_id: INT (Foreign Key ke users.user\_id)

order\_date: TIMESTAMP (Default CURRENT\_TIMESTAMP)

total\_amount: DECIMAL(10, 2)

order\_status: ENUM('pending\_payment', 'processing', 'shipped', 'delivered', 'completed', 'cancelled')

payment\_method: VARCHAR(50)

shipping\_address: TEXT (Bisa ambil dari users atau diisi saat checkout)

shipping\_city: VARCHAR(100)

shipping\_postal\_code: VARCHAR(10)

tracking\_number: VARCHAR(100) (Nomor resi pengiriman)

notes: TEXT (Opsional, catatan dari konsumen)

**8. Tabel order\_items (Detail Item Pesanan)**

order\_item\_id: INT (Primary Key, Auto Increment)

order\_id: INT (Foreign Key ke orders.order\_id)

product\_id: INT (Foreign Key ke products.product\_id)

variant\_id: INT (Foreign Key ke product\_variants.variant\_id, Nullable) - Jika produk memiliki varian.

quantity: INT

price\_at\_purchase: DECIMAL(10, 2) - Harga produk saat dipesan (untuk mencegah masalah jika harga berubah).

**9. Tabel cart (Keranjang Belanja)**

Untuk menyimpan item di keranjang belanja sebelum checkout.

cart\_id: INT (Primary Key, Auto Increment)

user\_id: INT (Foreign Key ke users.user\_id, Unique) - Setiap user hanya punya satu keranjang.

created\_at: TIMESTAMP

updated\_at: TIMESTAMP

**10. Tabel cart\_items (Item dalam Keranjang)**

cart\_item\_id: INT (Primary Key, Auto Increment)

cart\_id: INT (Foreign Key ke cart.cart\_id)

product\_id: INT (Foreign Key ke products.product\_id)

variant\_id: INT (Foreign Key ke product\_variants.variant\_id, Nullable)

quantity: INT

**11. Tabel reviews (Ulasan Produk)**

Untuk fitur komentar/ulasan.

review\_id: INT (Primary Key, Auto Increment)

product\_id: INT (Foreign Key ke products.product\_id)

user\_id: INT (Foreign Key ke users.user\_id)

rating: TINYINT (1-5)

comment: TEXT (Nullable)

review\_date: TIMESTAMP (Default CURRENT\_TIMESTAMP)

**12. Tabel transactions (Log Pembayaran)**

Untuk mencatat setiap transaksi pembayaran.

transaction\_id: INT (Primary Key, Auto Increment)

order\_id: INT (Foreign Key ke orders.order\_id)

payment\_gateway\_ref: VARCHAR(255) (ID transaksi dari payment gateway)

amount: DECIMAL(10, 2)

currency: VARCHAR(5) (Misal: IDR)

transaction\_status: ENUM('pending', 'success', 'failed', 'refunded')

transaction\_date: TIMESTAMP (Default CURRENT\_TIMESTAMP)

**13. Tabel complaints (Komplain Konsumen)**

Untuk fitur layanan purna jual (aftersales service).

complaint\_id: INT (Primary Key, Auto Increment)

user\_id: INT (Foreign Key ke users.user\_id)

order\_id: INT (Foreign Key ke orders.order\_id, Nullable jika komplain bukan terkait order)

subject: VARCHAR(255) (Subjek komplain)

description: TEXT (Detail komplain)

complaint\_status: ENUM('open', 'in\_progress', 'resolved', 'closed')

created\_at: TIMESTAMP

updated\_at: TIMESTAMP

**14. Tabel messages (Live Chat)**

Untuk fitur live chat antara admin dan konsumen.

message\_id: INT (Primary Key, Auto Increment)

sender\_id: INT (Foreign Key ke users.user\_id)

receiver\_id: INT (Foreign Key ke users.user\_id)

message\_text: TEXT

sent\_at: TIMESTAMP (Default CURRENT\_TIMESTAMP)

is\_read: BOOLEAN (Default FALSE) - Untuk menandai pesan sudah dibaca.

1. **Folder Models**