

## DRAFT STRUKTUR FOLDER

```
kelompok_24/
    ├── README.md          # [Wajib] Dokumentasi Project & Anggota
    ├── database.sql        # [Wajib] Schema Database Lengkap (Hasil Export)
    ├── erd.jpg             # [Wajib] Diagram Relasi Entitas
    ├── docs/               # Dokumentasi Pendukung
    │   └── WORKFLOW_GUIDE.md # SOP Git yang tadi kita buat
    └── src/                # SOURCE CODE UTAMA
        ├── index.html       # Halaman Login (Retro Style - Entry Point)
        ├── dashboard.html    # Shell Aplikasi (Sidebar + Content Container)
        ├── assets/
        │   ├── images/         # Logo, Foto Produk Dummy
        │   └── icons/          # (Opsiional) SVG icons jika tidak pakai inline
        ├── api/               # [BACKEND] PHP Native (Output JSON Only)
        │   ├── config.php      # Konstanta (DB Credentials, Cors Header)
        │   ├── db.php          # Koneksi Database ($conn)
        │   ├── auth.php         # Login, Logout, Session Check
        │   ├── users.php        # [Admin] CRUD Kasir (Tabel USERS)
        │   ├── categories.php   # [Admin] CRUD Kategori (Tabel CATEGORIES)
        │   ├── items.php         # [Admin] CRUD Menu Jual (Tabel MENU)
        │   ├── ingredients.php  # [Admin] CRUD Bahan Baku (Tabel INGREDIENTS)
        │   ├── orders.php        # [Kasir] Proses Transaksi (Tabel TRANSACTIONS & ITEMS)
        │   ├── reports.php       # [Admin] Data Laporan & Grafik (Query JOIN berat)
        │   └── recipes.php       # [Admin] Setting Resep Menu (Tabel MENU_RECIPES)
        ├── js/                 # [FRONTEND] Vanilla JS Logic
        │   ├── theme.js         # Konfigurasi Tailwind (Warna WarkOps)
        │   ├── router.js        # Logic Navigasi SPA (Hash Router)
        │   ├── utils.js         # Helper (FormatRupiah, FormatDate, PrintStruk)
        │   ├── components/
        │   │   ├── modal.js       # Logic Pop-up Form (Tambah/Edit)
        │   │   └── toast.js        # Logic Notifikasi (Sukses/Gagal)
        │   └── controllers/     # Logic per Halaman (Menghubungkan View & API)
        │       └── auth.js        # Logic Login Page
```

```
    └── dashboard.js # Logic Home (Chart Omzet)
    └── pos.js      # Logic Kasir (Cart, Kalkulasi, Checkout)
    └── inventory.js # Logic Tab Menu & Bahan Baku
    └── users.js    # Logic Manajemen Karyawan
    └── reports.js  # Logic Filter Laporan

└── views/       # [VIEWS] HTML Fragments (Tanpa <head>)
    └── home.html   # Dashboard Utama (Welcome & Stats)
    └── pos.html    # Interface Kasir (Grid Menu + Sidebar Cart)
    └── inventory.html # Tabel Menu, Bahan, & Resep
    └── users.html   # Tabel Manajemen User
    └── reports.html # Tampilan Laporan & Grafik
```

JUDUL : WarkOps – Sistem Point of Sale (POS) Terintegrasi untuk Manajemen Operasional Warkop Berbasis Web  
 TEMA : Sistem P.O.S

**WarkOps** merupakan sistem POS berbasis web dengan pendekatan SPA dan REST API yang dirancang untuk mengoptimalkan manajemen operasional warkop secara modern, cepat, dan efisien.

ARSITEKTUR	
DB	<b>MySQL:</b> Stores relational data (Users, Products, Transactions). <b>Artifacts:</b> database.sql (Schema), ERD image.
Backend (API)	<b>PHP Native:</b> Acts purely as a RESTful API. <b>Role:</b> It does not render HTML. It receives requests, talks to MySQL, and outputs JSON. <b>Authentication:</b> PHP Sessions (session_start()) or simple Token-based auth.
Frontend (client)	<b>Structure:</b> HTML5 (The "Shell" architecture). <b>Styling:</b> Tailwind CSS (via CDN is easiest for this setup to avoid node_modules). <b>Logic:</b> JavaScript Native (ES6+). Router: Custom Hash Router (handling #inventory, #pos). HTTP Client: Native fetch() API (to talk to PHP). DOM manip.: Native document.getElementById, innerHTML, etc.

Data Flow	
1	Browser loads index.html (The Shell) + router.js.
2	User clicks "Inventory" menu.
3	Router fetches views/inventory.html and injects it into the main <div>.
4	Controller (inventory.js) fires fetch('/api/items.php').
5	PHP queries MySQL, gets rows, encodes to JSON (json_encode), and returns it.
6	JavaScript receives JSON and loops through it to build the HTML Table rows.

## **FITUR PER ROLE**

### **A. ADMIN (Pemilik Warkop)**

#### **1. Manajemen Menu dan Inventori**

- Tambah menu baru (kopi susu, mie goreng, dll.)
- Edit harga menu
- Edit stok bahan
- Hapus menu
- Kelola kategori (kopi, makanan ringan, dll.)

#### **2. Manajemen User / Kasir**

- Tambah akun kasir
- Reset password
- Aktif/nonaktifkan kasir

#### **3. Laporan dan Analitik**

- Penjualan harian
- Penjualan mingguan
- Penjualan per menu
- Penjualan berdasarkan jam ramai
- Inventory usage (bahan cepat habis)

#### **4. Riwayat Transaksi**

- Cek transaksi per kasir
- Filter berdasarkan tanggal
- Export CSV/PDF

#### **5. Dashboard (Chart)**

Data real-time:

- Total pendapatan hari ini
- Menu paling laku
- Grafik omzet 7 hari

### **B. KASIR / STAFF**

#### **1. Point of Sale (POS) - Halaman Kasir**

- Pilih menu
- Tambah jumlah
- Hitung total otomatis
- Input pembayaran
- Hitung kembalian otomatis
- Cetak struk (opsional)

**2. Cek Stok (Read-Only)**

- Kasir hanya bisa melihat stok, bukan mengubah. Untuk menghindari manipulasi.

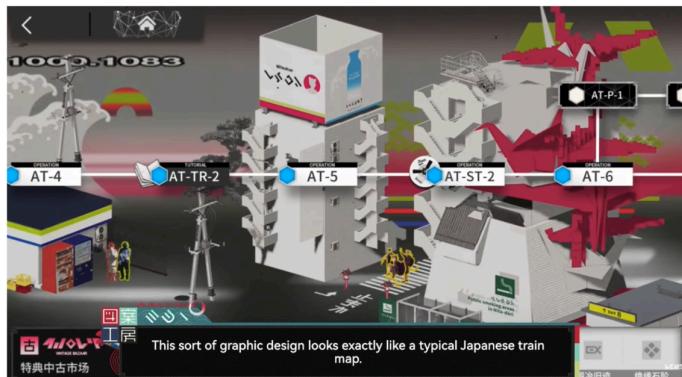
**3. Riwayat Transaksi Harian (Read-Only)**

- Kasir hanya bisa melihat transaksi *yang dia buat sendiri*.

**4. LOGOUT**

Agar bergantian shift aman.

## REFERENCE BOARD - ARKNIGHTS ATO EVENT VISUALS





中文CV ◆ kiyo 日文CV ◆ 丰田萌绘

辅助·护佑者



## Struktur Database

### 1. USERS

Digunakan untuk Admin & Kasir (role-based).

Field	Tipe
user_id (PK)	INT
username	VARCHAR
password_hash	VARCHAR
nama	VARCHAR
role	ENUM('admin','kasir')
is_active	BOOLEAN

Kenapa: karena kamu butuh fitur aktif/nonaktif & reset password.

### 2. CATEGORIES

Kategori menu.

Field	Tipe
category_id	INT
name	VARCHAR

### 3. MENU

Data menu yang dijual di POS.

Field	Tipe
menu_id	INT
category_id (FK)	INT
name	VARCHAR
price	DECIMAL
is_available	BOOLEAN

---

## 4. INGREDIENTS (Bahan Baku)

Untuk kebutuhan inventory usage.

Field	Tipe
ingredient_id	INT
name	VARCHAR
stock_qty	DECIMAL
unit	VARCHAR (gram/liter/sachet dll.)

---

## 5. MENU\_RECIPES (Bill of Materials)

Relasi banyak-banyak: 1 menu bisa butuh banyak bahan, 1 bahan dipakai banyak menu.

Field	Tipe
recipe_id	INT
menu_id (FK)	INT
ingredient_id (FK)	INT
qty_used	DECIMAL

Alasan kritis: tanpa ini, tidak bisa menghitung bahan cepat habis.

---

## 6. TRANSACTIONS (Header)

Transaksi utama.

Field	Tipe
trx_id	INT
user_id (FK)	INT
datetime	DATETIME
total	DECIMAL
payment	DECIMAL
change	DECIMAL

## 7. TRANSACTION\_ITEMS (Detail)

Setiap item menu yang dibeli.

Field	Tipe
item_id	INT
trx_id (FK)	INT
menu_id (FK)	INT
qty	INT
price_at_time	DECIMAL

Kenapa harus ada `price_at_time` ?

Karena laporan lama harus tetap valid meskipun harga menu berubah.

---

## 8. INVENTORY\_LOGS

Catatan perubahan stok.

Field	Tipe
log_id	INT
ingredient_id	INT
change_qty	DECIMAL (+/-)
reason	ENUM('used','restock','correction')
datetime	DATETIME

Untuk kebutuhan auditing & laporan bahan cepat habis.



