

Nama : Dewi Farah Aulia Bahiyyah

NIM : 20230801226

LAPORAN Pengerjaan

Step 1: Prasyarat

- Docker Engine: Harus terpasang di sistem operasi.
- Docker Compose: Diperlukan untuk mengorkestrasi beberapa layanan kontainer.

Step 2: Hapus fork dan clone repository UTS (yang sudah dienkripsi dan diautentikasi)

- Repository lama: <https://github.com/defalia2b/ksi>
- Repository baru: <https://github.com/defalia2b/ksi-project>

Step 3: Menambahkan analisa studi kasus pada Readme.MD

 README

 

Dokumentasi Proyek: Sistem Inventaris dan Kasir (KSI)

Studi Kasus: Transformasi Digital "Toko Ritel Jaya Abadi"

Latar Belakang

"Toko Ritel Jaya Abadi" adalah sebuah toko kelontong skala menengah yang telah beroperasi selama 10 tahun. Selama ini, pemilik toko, Bapak Budi, mengandalkan pencatatan manual menggunakan buku besar dan kalkulator untuk mengelola stok barang, melacak penjualan, dan mengelola hubungan dengan pemasok (supplier).

Masalah yang Dihadapi

Seiring dengan meningkatnya jumlah pelanggan dan variasi produk, metode manual yang digunakan Bapak Budi menimbulkan berbagai masalah serius:

- **Ketidakakuratan Stok Barang:** Sering terjadi perbedaan antara jumlah stok fisik di rak dengan catatan di buku. Hal ini menyebabkan kerugian, baik karena stok kosong (pelanggan kecewa) maupun stok berlebih (modal tertahan dan risiko barang kedaluwarsa).
- **Proses Transaksi Lambat:** Kasir harus memasukkan harga setiap barang secara manual, yang memperlambat antrean pembayaran dan meningkatkan risiko salah hitung.
- **Sulit Melacak Laporan Penjualan:** Bapak Budi kesulitan mendapatkan gambaran cepat tentang produk mana yang paling laku atau kapan waktu penjualan paling ramai. Laporan penjualan bulanan membutuhkan waktu sehari-hari untuk direkapitulasi.
- **Manajemen Supplier Tidak Efisien:** Informasi kontak dan riwayat pemesanan dari supplier tersebar di berbagai catatan, sehingga sulit untuk melakukan pemesanan ulang atau negosiasi harga.

Solusi yang Ditawarkan oleh Aplikasi

Untuk mengatasi kekacauan ini, "Toko Ritel Jaya Abadi" mengimplementasikan **Aplikasi KSI (Kasir dan Stok Inventaris)**. Aplikasi ini dirancang untuk menjadi pusat kendali operasional toko dengan fitur-fitur yang menjawab langsung setiap masalah.

Aplikasi ini menawarkan solusi digital terpusat yang memungkinkan admin (pemilik atau manajer toko) untuk:

- **Mengelola Produk dan Stok:** Melakukan input data produk baru, memperbarui harga, dan melihat sisa stok secara real-time.
 - **Melakukan Penyesuaian Stok:** Mencatat penyesuaian jika ada barang yang rusak atau hilang melalui fitur Penyesuaian Stok.
 - **Mencatat Transaksi Penjualan:** Mempercepat layanan kasir dengan sistem pencatatan transaksi yang terintegrasi langsung untuk mengurangi stok.
 - **Mengelola Supplier dan Kategori:** Menyimpan data supplier dan mengelompokkan produk berdasarkan kategori untuk manajemen yang lebih rapi.
-

🔮 ✨ Fitur-Fitur Utama Aplikasi

Berdasarkan analisis file Filament Resources, aplikasi ini memiliki beberapa fitur manajemen inti:

- **Manajemen Kategori:** Admin dapat menambah, mengubah, dan menghapus kategori produk untuk pengelompokan yang lebih baik.
 - **Manajemen Supplier:** Mengelola data pemasok barang, termasuk nama, alamat, dan kontak yang bisa dihubungi.
 - **Manajemen Produk:** Mengelola detail setiap produk, seperti nama, kode, harga beli, harga jual, dan jumlah stok awal. Fitur ini terhubung dengan Kategori dan Supplier.
 - **Manajemen Transaksi:** Mencatat semua transaksi penjualan yang terjadi. Setiap transaksi akan berisi detail produk yang dibeli, jumlah, dan total harga.
 - **Penyesuaian Stok:** Sebuah fitur khusus untuk mencatat perubahan stok di luar transaksi penjualan (misalnya, stok opname, barang rusak, atau barang hilang).
 - **Manajemen Pengguna:** Mengatur siapa saja yang bisa mengakses sistem, dengan peran yang bisa didefinisikan (misalnya, admin atau kasir).
-

Desain Database

Struktur database dirancang untuk mendukung semua fitur di atas. Berikut adalah rincian tabel utama berdasarkan file migrasi di dalam proyek:

1. users (dari 0001_01_01_000000_create_users_table.php)

Menyimpan data login untuk pengguna sistem (Admin/Kasir).

Kolom	Tipe Data	Keterangan
id	BIGINT	Primary Key
name	STRING	Nama pengguna
email	STRING	Email unik untuk login
password	STRING	Kata sandi (sudah di-hash)

2. kategoris (dari 2025_05_21_155442_create_kategoris_table.php)

Menyimpan daftar kategori produk.

Kolom	Tipe Data	Keterangan
id	BIGINT	Primary Key
nama	STRING	Nama kategori (contoh: "Makanan Ringan")

3. suppliers (dari 2025_05_21_155448_create_suppliers_table.php)

Menyimpan data pemasok barang.

Kolom	Tipe Data	Keterangan
id	BIGINT	Primary Key
nama	STRING	Nama supplier
alamat	TEXT	Alamat supplier
telepon	STRING	Nomor telepon

4. produks (dari 2025_05_21_155449_create_produks_table.php)

Tabel inti yang menyimpan semua data produk.

Kolom	Tipe Data	Keterangan
id	BIGINT	Primary Key
kategori_id	FK	Terhubung ke kategoris.id
supplier_id	FK	Terhubung ke suppliers.id
nama	STRING	Nama produk
harga_beli	DECIMAL	Harga modal dari supplier
harga_jual	DECIMAL	Harga jual ke pelanggan
stok	INTEGER	Jumlah stok saat ini

5. transaksis (dari 2025_05_21_155450_create_transaksis_table.php)

Mencatat setiap transaksi (header).

Kolom	Tipe Data	Keterangan
id	BIGINT	Primary Key
total_harga	DECIMAL	Total nilai dari satu transaksi
tanggal_transaksi	DATE	Tanggal saat transaksi terjadi

6. detail_transaksi (dari 2025_05_21_155451_create_detail_transaksi_table.php)

Mencatat rincian barang per transaksi.

Kolom	Tipe Data	Keterangan
id	BIGINT	Primary Key
transaksi_id	FK	Terhubung ke transaksis.id
produk_id	FK	Terhubung ke produks.id
jumlah	INTEGER	Jumlah produk yang dibeli
subtotal	DECIMAL	Harga total untuk produk ini (harga_jual * jumlah)

7. penyesuaian_stoks (dari 2025_05_21_160355_create_penyesuaian_stoks_table.php)

Mencatat semua aktivitas penyesuaian stok.

Kolom	Tipe Data	Keterangan
id	BIGINT	Primary Key
produk_id	FK	Terhubung ke produks.id
jumlah	INTEGER	Jumlah yang disesuaikan (bisa positif/negatif)
keterangan	TEXT	Alasan penyesuaian (misal: "Barang rusak")
tanggal	DATE	Tanggal penyesuaian stok

VULNERABILITY ASSESSMENT

Step 1: Persiapan environment di Windows

- ➔ Install Chocolatey & mkcert
- ➔ Buat sertifikat SSL Lokal

```
1. # Instal otoritas sertifikat lokal (hanya perlu sekali)
2. mkcert -install
3.
4. # Buat sertifikat untuk domain ksi.test
5. mkcert ksi.test
```

➔ Daftarkan domain lokal ke host file

```
1. Add-Content -Path "C:\Windows\System32\drivers\etc\hosts" -Value "`n127.0.0.1 ksi.test"
```

Step 2: Persiapan di Ubuntu

- ➔ Install ptt scanner, berikan izin eksekusi, dan pindahkan ke direktori bin agar bisa dipanggil dari mana saja.
- ➔ Unduh paket .deb Cloudflared dan instal menggunakan dpkg.
- ➔ Instal jq untuk pemrosesan JSON dan pastikan Docker sudah terinstal dan berjalan di sistem.

Step 3: Menjalankan proyek ksi-project

```
1. cd /home/defalia/ksi-project/ksi-project
2. ./start.sh ksi
```

Step 4: Mengekspos proyek lokal ke internet

```
1. cloudflared tunnel --url http://ksi.test
```

Mengapa http bukan https? Karena layanan Nginx di dalam Docker saya berjalan pada port 80 (HTTP). Cloudflared akan menangani enkripsi HTTPS untuk akses publik, jadi koneksi dari cloudflared ke Docker cukup menggunakan HTTP.

Didapatkan URL

```
2025-07-23T07:53:34Z INF +-----+
-----+
2025-07-23T07:53:34Z INF | Your quick Tunnel has been created! Visit it at (it may take some time to be rea
chable): |
2025-07-23T07:53:34Z INF | https://telephony-singles-gradually-predict.trycloudflare.com
2025-07-23T07:53:34Z INF +-----+
-----+
```

Jalankan:

```
1. ptt run website-scanner https://telephony-singles-gradually-predict.trycloudflare.com
```

Contoh tampilan website scanner:

- Referrer-Policy
- Content-Security-Policy
- X-Content-Type-Options

Middleware ini juga telah saya daftarkan secara global agar semua halaman yang diakses pengguna otomatis menyertakan header keamanan ini.

2. Flag 'Secure' pada Session Cookie

Untuk memastikan cookie sesi tidak dikirim melalui koneksi yang tidak aman, saya telah mengatur agar flag Secure selalu aktif secara default. Pengaturan ini saya terapkan langsung di dalam file config/session.php.

3. Keamanan pada Konfigurasi Nginx

Saya memperbarui file konfigurasi nginx/default.conf untuk melakukan dua hal:

- Menambahkan semua security headers yang direkomendasikan.
- Menonaktifkan server_tokens untuk menyembunyikan informasi versi server yang berjalan.

4. Penanganan Mixed Content

Untuk mencegah isu mixed content, saya mengubah URL aplikasi default di dalam file config/app.php menjadi https://localhost. Saya juga sudah memeriksa dan tidak menemukan adanya link asset yang menggunakan http://localhost secara hardcoded di dalam file-file PHP proyek ini.

5. File robots.txt

Saya memastikan bahwa file robots.txt yang ada saat ini tidak mengekspos direktori atau path yang sensitif. Konfigurasi yang ada sekarang sudah aman karena secara default mengizinkan semua crawler.

Dari:

```
+----- TEST summary -----+
|
| URL: https://telephony-singles-gradually-predict.trycloudflare.com/|
| High Risk Findings: 0                                           |
| Medium Risk Findings: 1                                         |
| Low Risk Findings: 7                                            |
| Info Risk Findings: 31                                          |
| Start time: 2025-07-23 10:56:49                                |
| End time: 2025-07-23 10:57:13                                  |
|
+-----+

```

Menjadi:

```
+----- TEST summary -----+
|
| URL: https://telephony-singles-gradually-predict.trycloudflare.com/|
| High Risk Findings: 0
| Medium Risk Findings: 0
| Low Risk Findings: 4
| Info Risk Findings: 35
| Start time: 2025-07-23 11:16:49
| End time: 2025-07-23 11:17:18
|
+-----
```

Langkah selanjutnya:

Menjadi: