

BAB 3

PELAKSANAAN KERJA MAGANG

Bab ini merangkum seluruh aktivitas yang penulis lakukan selama enam bulan menjalani *Career Acceleration Program* di PT. Hanielshen. Fokus pembahasan mencakup realisasi pekerjaan di lapangan, tanggung jawab teknis, serta bagaimana teori akademis diterapkan dalam kasus nyata industri. Teknologi yang menjadi fokus utama penulis adalah pengembangan *backend* menggunakan PHP Native yang dihubungkan dengan basis data MySQL.

Uraian berikut disusun mulai dari posisi penulis dalam tim, detail tugas teknis harian, hasil antarmuka aplikasi yang dibangun, hingga jadwal kegiatan mingguan.

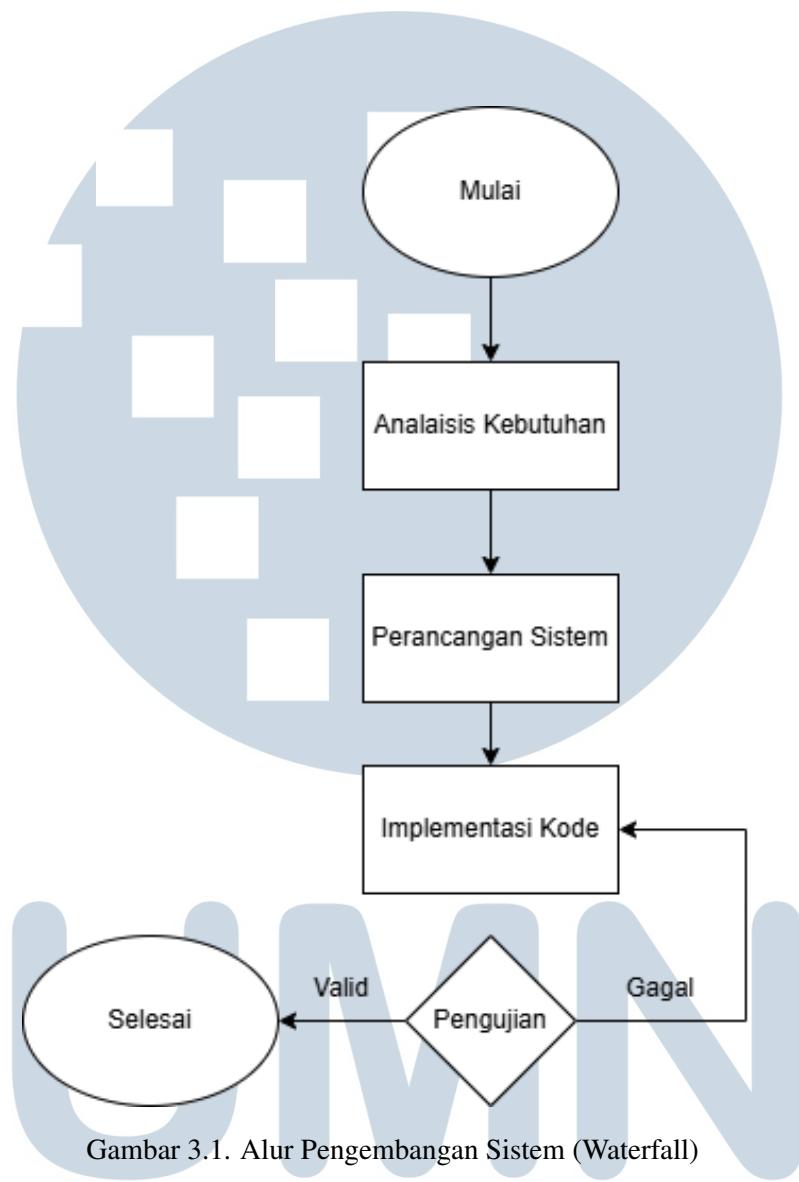
3.1 Kedudukan dan Koordinasi

Dalam struktur perusahaan, penulis bergabung dengan Divisi Pemrograman (*IT Development Division*) sebagai *Backend Developer Intern*. Tugas utamanya cukup spesifik, yaitu membantu tim inti dalam merancang dan memelihara sistem informasi internal. Proyek besar yang dipercayakan kepada penulis adalah pembuatan **“Modul Manajemen Inventori Gudang”** untuk menggantikan pencatatan manual yang rawan kesalahan dan meningkatkan akurasi data stok perusahaan.

Selama bekerja, penulis dibimbing langsung oleh **“Bapak Adityawan, S.Kom., M.Si.”** sebagai mentor lapangan. Agar ritme kerja tetap terjaga, koordinasi dilakukan setiap pagi pukul 09.00 WIB melalui sesi *Daily Standup Meeting*. Di sana, setiap anggota tim, termasuk penulis, melaporkan progres pekerjaan kemarin dan target hari ini.

3.2 Metodologi Pengembangan Sistem

Dalam mengembangkan sistem ini, penulis menerapkan metode *System Development Life Cycle* (SDLC) model **“Waterfall”**. Model ini dipilih karena alur kerjanya yang terstruktur, sistematis, dan berurutan.



Gambar 3.1. Alur Pengembangan Sistem (Waterfall)

Sumber: Dokumen Internal Perusahaan

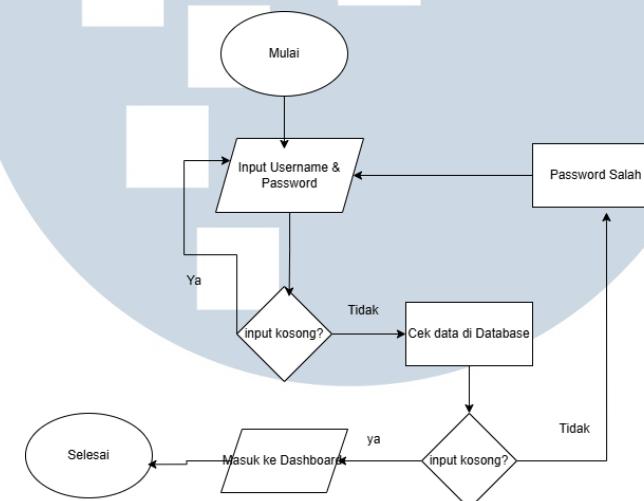
Penjelasan: Gambar 3.1 di atas memvisualisasikan siklus hidup pengembangan sistem yang bersifat linier. Tahapan dimulai dari **Analisis Kebutuhan** untuk memetakan masalah selisih stok, dilanjutkan dengan **Desain Sistem** yang meliputi perancangan basis data dan antarmuka. Setelah desain disetujui, masuk ke tahap **Implementasi Kode** (Coding), dilanjutkan dengan **Pengujian** (Testing) untuk memastikan bebas error, dan diakhiri dengan tahap **Deployment** ke server lokal.

3.3 Perancangan Alur Sistem (Flowchart)

Sebelum baris kode ditulis, penulis terlebih dahulu merancang logika sistem agar alur datanya jelas. Berikut adalah desain alur kerja (*flowchart*) untuk modul-modul yang dikerjakan.

3.3.1 Alur Login Pengguna

Gerbang pertama masuk ke aplikasi adalah halaman *Login*. Logika di sini dibuat ketat agar tidak sembarang orang bisa masuk dan memanipulasi data stok.



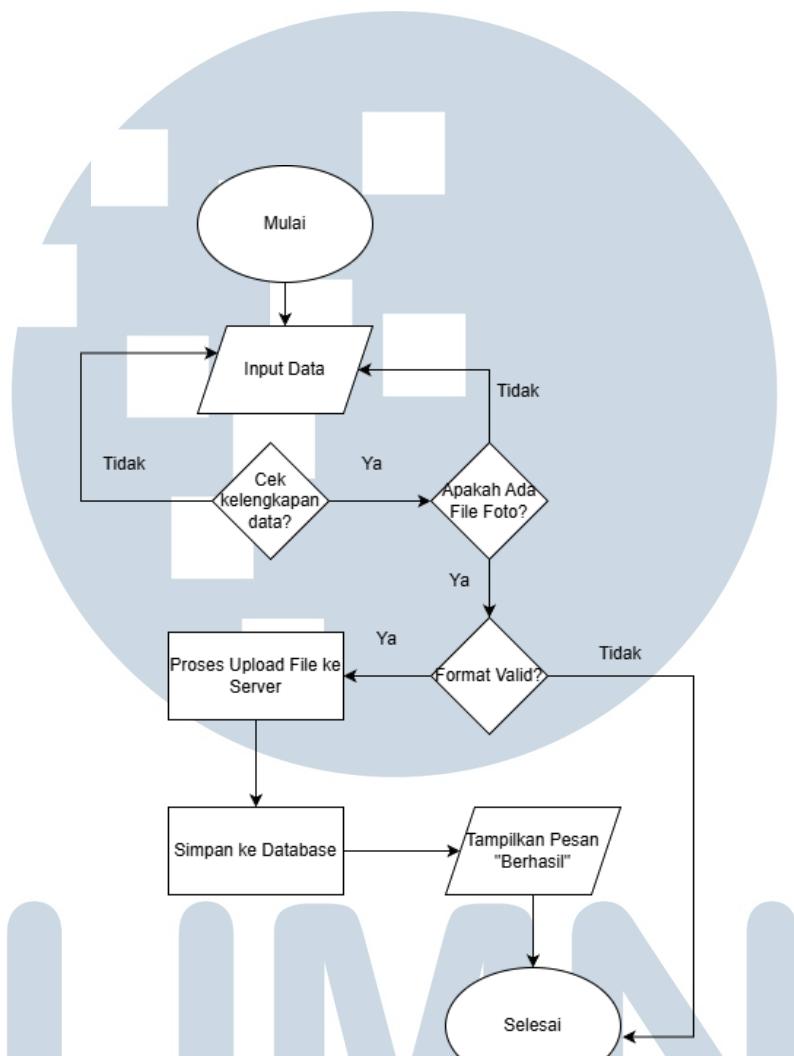
Gambar 3.2. Flowchart Alur Login Sistem

Sumber: Dokumen Internal Perusahaan

Penjelasan Alur: Seperti terlihat pada Gambar 3.2, mekanisme keamanan dimulai saat pengguna menginput *username* dan *password*. Sistem melakukan validasi bertingkat: pertama memeriksa apakah kolom input kosong, kemudian mencocokkan data kredensial dengan tabel pengguna di database. Jika data cocok, sistem akan membuat *Session Login* dan memberikan hak akses ke Dashboard. Jika gagal, akses ditolak dan user diminta input ulang.

3.3.2 Manajemen Data Master (Upload Foto)

Agar petugas gudang lebih mudah mengenali fisik barang, penulis menambahkan fitur upload foto pada menu "Tambah Barang".



Gambar 3.3. Flowchart Tambah Barang dan Validasi Foto

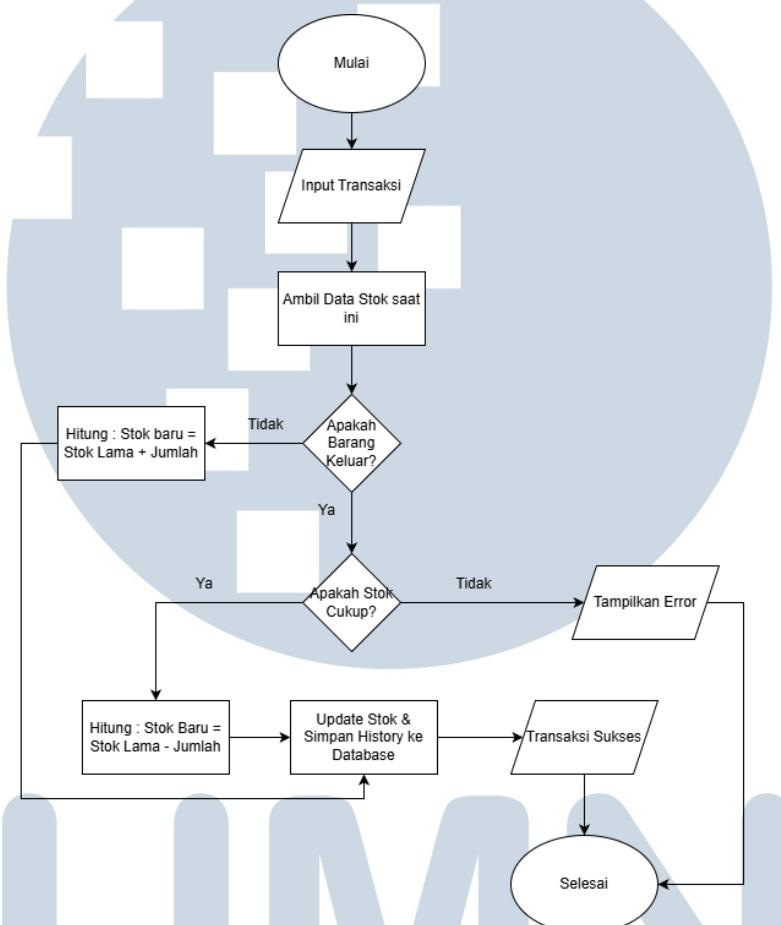
Sumber: Dokumen Internal Perusahaan

Penjelasan Alur: Proses pada Gambar 3.3 menggambarkan alur input data master. Admin mengisi form detail barang (Kode, Nama, Kategori) dan mengunggah gambar produk. Sistem menerapkan validasi tipe file untuk memastikan yang diunggah hanya format gambar (.jpg/.png). Jika validasi lolos, file gambar disimpan ke folder server dan nama filenya disimpan ke database MySQL.

N U S A N T A R A

3.3.3 Transaksi Stok (Logika Matematika)

Bagian paling rumit adalah menangani logika barang keluar. Seringkali terjadi stok di aplikasi menjadi minus karena kesalahan input pengguna.



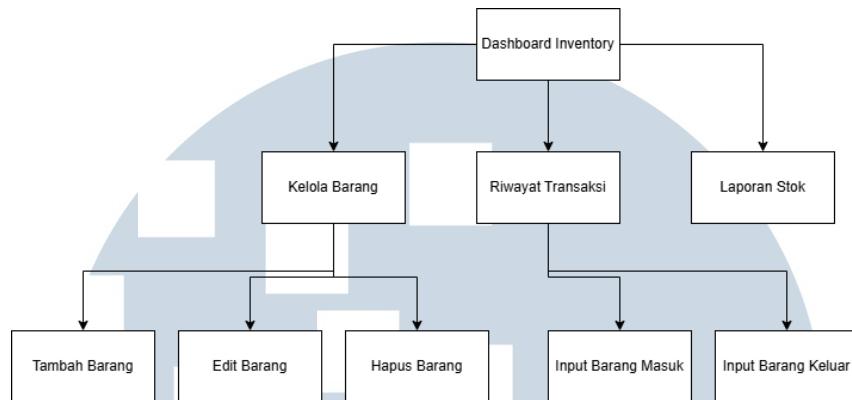
Gambar 3.4. Flowchart Transaksi Stok Barang

Sumber: Dokumen Internal Perusahaan

Penjelasan Alur: Gambar 3.4 menunjukkan logika krusial untuk menjaga integritas data stok. Saat user memilih transaksi "Barang Keluar", sistem secara otomatis mengecek jumlah stok yang tersedia di database. Logika kondisional "Apakah Permintaan & Stok Tersedia?" diterapkan. Jika benar (stok kurang), sistem akan menolak transaksi dan menampilkan pesan error. Jika stok cukup, sistem akan mengurangi jumlah stok dan menyimpan riwayat transaksi.

3.3.4 Perancangan Database dan Menu

Selain alur logika, struktur penyimpanan data dan navigasi menu juga dirancang agar sistem terintegrasi dengan baik.



Gambar 3.5. Sitemap Struktur Menu Aplikasi

Sumber: Dokumen Internal Perusahaan

Penjelasan: Gambar 3.5 memetakan arsitektur informasi aplikasi. Struktur menu dirancang sederhana agar mudah dipahami (*user-friendly*). Menu utama terdiri dari ****Dashboard**** untuk ringkasan data, ****Master Data**** untuk pengelolaan barang (CRUD), ****Transaksi**** untuk mencatat arus masuk-keluar barang, dan ****Laporan**** untuk rekapitulasi akhir.



Gambar 3.6. Entity Relationship Diagram (ERD)

Sumber: Dokumen Internal Perusahaan

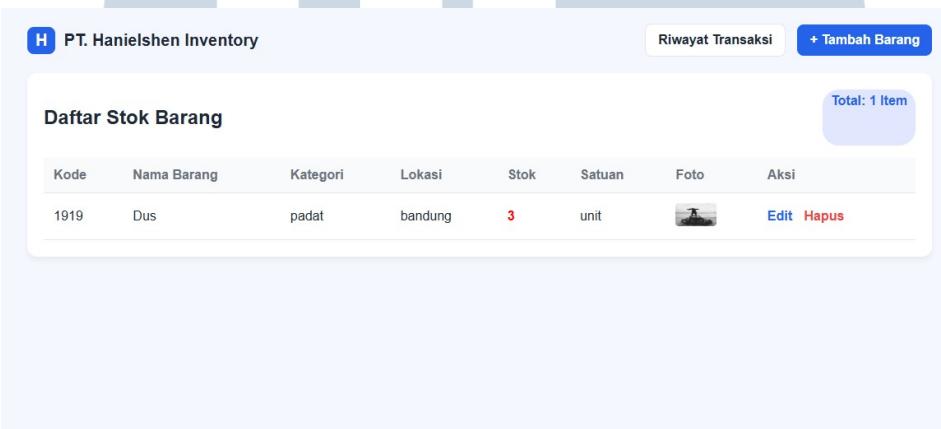
Penjelasan: Gambar 3.6 menampilkan rancangan database relasional yang digunakan. Terdapat relasi *One-to-Many* antara tabel barang dan tabel transaksi_stok. Kolom id pada tabel barang bertindak sebagai *Primary Key* (PK) yang dirujuk oleh kolom barang_id pada tabel transaksi sebagai *Foreign Key* (FK). Struktur ini menjamin integritas data, di mana transaksi tidak bisa dicatat jika barangnya tidak terdaftar.

3.4 Implementasi Antarmuka Sistem

Setelah rancangan logika matang, tahap selanjutnya adalah menerjemahkannya ke dalam tampilan web (*User Interface*). Berikut adalah beberapa tampilan utama yang berhasil diselesaikan.

3.4.1 Halaman Dashboard Utama

Ini adalah halaman yang pertama kali menyambut pengguna setelah login.



Kode	Nama Barang	Kategori	Lokasi	Stok	Satuan	Foto	Aksi
1919	Dus	padat	bandung	3	unit		Edit Hapus

Gambar 3.7. Tampilan Halaman Dashboard Utama

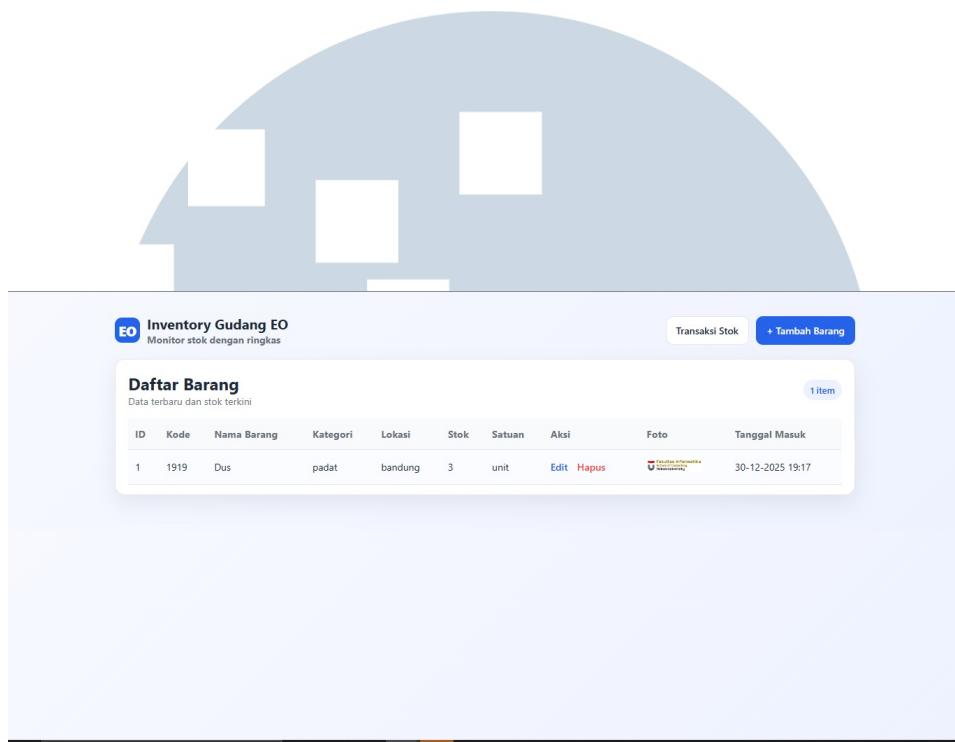
Sumber: Dokumen Internal Perusahaan

Penjelasan: Penulis membuat tampilan Dashboard (Gambar 3.7) dengan desain yang bersih dan minimalis untuk memudahkan pembacaan informasi. Di bagian atas halaman, terdapat komponen kartu statistik (*summary cards*) yang menampilkan ringkasan penting seperti total jenis barang yang terdaftar, total akumulasi stok, dan jumlah transaksi yang terjadi pada hari tersebut. Kartu-kartu ini dirancang dengan warna yang kontras agar data-data krusial dapat langsung terlihat oleh pengguna saat pertama kali masuk.

Selain aspek visual, dashboard ini juga berfungsi sebagai pusat kendali untuk navigasi. Informasi yang disajikan secara *real-time* memungkinkan manajer gudang atau admin untuk mengambil keputusan cepat tanpa harus menggali laporan detail yang panjang. Misalnya, jika jumlah transaksi hari ini terlihat sangat tinggi, manajer bisa segera menyiapkan tim tambahan untuk operasional gudang.

3.4.2 Halaman Daftar Barang (Inventory List)

Pusat pengelolaan data barang ada di halaman ini.



Gambar 3.8. Tampilan Halaman Daftar Barang

Sumber: Dokumen Internal Perusahaan

Penjelasan: Pada Gambar 3.8, data seluruh barang disajikan dalam bentuk tabel yang responsif dan terstruktur rapi. Fitur unggulan di halaman ini adalah adanya kolom *thumbnail* yang menampilkan foto kecil dari setiap produk. Penambahan elemen visual ini bertujuan agar petugas gudang dapat memverifikasi kesesuaian antara data di sistem dengan fisik barang yang ada di rak, sehingga mengurangi risiko kesalahan pengambilan barang.

Dari sisi fungsionalitas, halaman ini dilengkapi dengan tombol aksi untuk *Edit* dan *Hapus* di setiap baris data, memberikan fleksibilitas penuh dalam manajemen data master. Sistem juga menerapkan *pagination* (pembagian halaman) agar tabel tidak terlalu panjang ke bawah jika data barang mencapai ratusan atau ribuan, menjaga performa aplikasi tetap ringan saat dimuat.

3.4.3 Form Input Barang Baru

Halaman ini digunakan admin saat ada stok barang jenis baru yang masuk gudang.



The screenshot shows a web-based form titled 'Input Barang Baru' (New Item Input). The form is part of the 'PT. Hanielshen Inventory' system. At the top right is a 'Kembali' (Back) button. The form fields are as follows:

- Kode Barang:** A text input field.
- Nama Barang:** A text input field.
- Kategori:** A text input field.
- Lokasi Gudang:** A text input field.
- Stok Awal:** A text input field containing the value '0'.
- Satuan:** A text input field containing the value 'unit'.
- Keterangan:** A text area for notes.
- Foto Barang (JPG/PNG):** A file upload input field with a 'Choose File' button and a message 'No file chosen'.

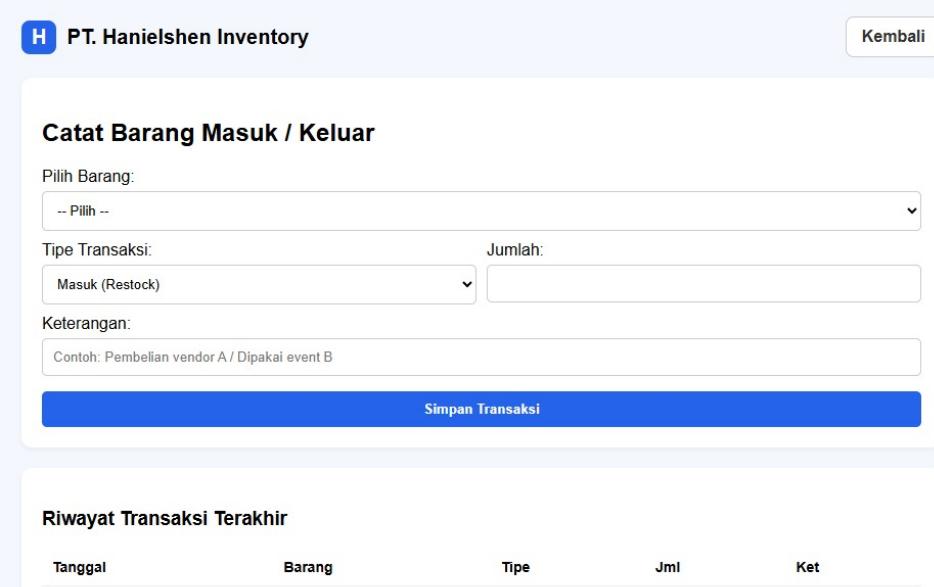
Gambar 3.9. Tampilan Form Input Barang

Sumber: Dokumen Internal Perusahaan

Penjelasan: Tampilan form pada Gambar 3.9 dirancang untuk meminimalkan kesalahan input (*human error*). Form ini memiliki kolom-kolom wajib isi seperti Kode Barang, Nama, dan Kategori. Tata letak form dibuat vertikal agar mudah diikuti alurnya oleh pengguna, mulai dari data teks hingga proses unggah gambar. Secara teknis, penulis menerapkan validasi tipe data yang ketat pada kolom "Stok Awal" dan "Harga". Kolom-kolom ini diprogram (melalui atribut HTML dan validasi JavaScript) agar hanya menerima inputan angka. Jika pengguna mencoba memasukkan huruf atau karakter spesial, sistem akan menolak input tersebut. Hal ini sangat penting untuk menjaga konsistensi tipe data di database MySQL agar operasi matematika (seperti penjumlahan stok) dapat berjalan lancar.

3.4.4 Halaman Transaksi Barang

Halaman operasional harian untuk mencatat barang masuk dan keluar.



PT. Hanielshen Inventory

Kembali

Catat Barang Masuk / Keluar

Pilih Barang:

-- Pilih --

Tipe Transaksi: Jumlah:

Masuk (Restock)

Keterangan:

Contoh: Pembelian vendor A / Dipakai event B

Simpan Transaksi

Riwayat Transaksi Terakhir

Tanggal	Barang	Tipe	Jml	Ket
---------	--------	------	-----	-----

Gambar 3.10. Tampilan Halaman Transaksi

Sumber: Dokumen Internal Perusahaan

Penjelasan: Pada fitur transaksi yang ditunjukkan Gambar 3.10, penulis mengimplementasikan antarmuka yang dinamis menggunakan teknologi AJAX. Ketika pengguna memilih nama barang dari menu *dropdown*, sistem secara otomatis akan mengambil data sisa stok terkini dari database dan menampilkannya di layar tanpa perlu me-refresh halaman. Fitur ini memberikan informasi instan kepada pengguna mengenai ketersediaan barang sebelum transaksi diproses. Selain itu, halaman ini menjadi benteng utama dalam mencegah stok minus. Sistem telah diprogram dengan logika kondisional: jika pengguna memilih jenis transaksi "Barang Keluar" dan memasukkan jumlah yang melebihi sisa stok yang tampil, tombol simpan akan dinonaktifkan atau akan muncul pesan peringatan (alert). Mekanisme ini memastikan integritas data stok tetap terjaga dan akurat sesuai kondisi fisik di gudang.

3.5 Uraian Pelaksanaan Magang

Proses magang berjalan selama 24 minggu penuh. Bulan-bulan awal lebih banyak dihabiskan untuk beradaptasi dengan lingkungan kerja dan mempelajari alur bisnis perusahaan. Setelah paham, barulah masuk ke fase pengembangan intensif (*coding*) dan ditutup dengan fase pengujian (*testing*).

Detail kegiatan mingguan yang penulis lakukan dapat dilihat pada tabel berikut:

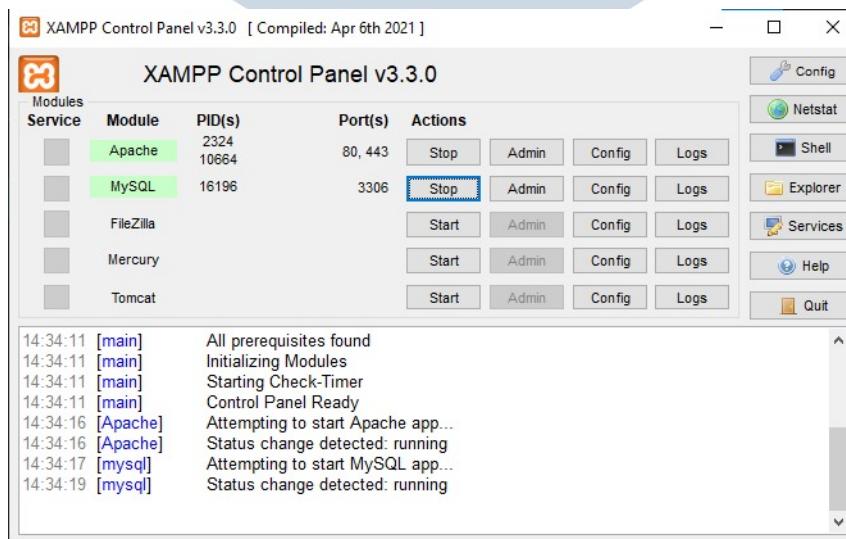
Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu	Rincian Kegiatan
1	Masa orientasi: Mempelajari profil perusahaan, aturan main, serta memahami spesifikasi sistem inventory yang diminta oleh user.
2	Melakukan riset kebutuhan (<i>Requirement Gathering</i>) dan mulai mencoret-coret desain alur sistem (Flowchart) untuk modul barang.
3	Fokus pada perancangan tampilan (UI/UX): Membuat sketsa kasar (<i>Wireframe</i>) dan alur pengguna.
4	Memperhalus desain menjadi <i>Mockup</i> (High-Fidelity) dan membuat <i>Prototype</i> agar bisa diklik dan disimulasikan.
5	Persiapan teknis: Menginstall XAMPP, VS Code, dan mengatur repositori Git untuk kolaborasi kode.
6	Mulai <i>ngoding</i> (Slicing): Mengubah desain gambar menjadi kode HTML/CSS, dimulai dari halaman Login.
7	Lanjut membuat tampilan depan (<i>Frontend</i>) untuk Dashboard dan menu-menu navigasi samping.
8	Tahap integrasi: Menghubungkan tampilan depan yang sudah jadi dengan database MySQL menggunakan PHP.
9	Mengerjakan fungsi dasar aplikasi (CRUD): Fokus pada fitur Input Data Barang (Create) dan Tampil Data (Read).
10	Melanjutkan fungsi CRUD: Mengerjakan fitur Edit Data (Update) dan Hapus Data (Delete) serta fitur Upload Foto.
11	Merapikan tampilan (<i>Styling</i>): Memastikan web terlihat bagus di berbagai ukuran layar dan mengecek validasi form.

Minggu	Rincian Kegiatan
12	Menambahkan fitur pelengkap: Membuat fitur pencarian (<i>Search</i>) dan filter data agar user lebih nyaman.
13	Fase pengujian (<i>Testing</i>): Mencoba semua fitur seolah-olah sebagai user untuk mencari error (<i>bug</i>).
14	Perbaikan: Memperbaiki fitur atau tampilan yang dirasa kurang pas berdasarkan masukan mentor.
15	<i>Deployment</i> : Mengunggah aplikasi ke server lokal kantor agar bisa diakses oleh staf lain dan membuat panduan pakai.
16	Finishing: Menyusun laporan akhir magang, merapikan dokumentasi kode, dan serah terima pekerjaan.

3.5.1 Lingkungan Pengembangan

Perangkat lunak dan alat komunikasi yang digunakan penulis selama proses pengembangan sistem adalah sebagai berikut.



Gambar 3.11. Tampilan XAMPP Control Panel

Sumber: Dokumen Internal Perusahaan

Penjelasan: Gambar 3.11 memperlihatkan panel kontrol aplikasi XAMPP yang digunakan penulis sebagai lingkungan server lokal (*localhost*). Dua modul utama yang wajib diaktifkan adalah **Apache** yang berfungsi sebagai Web Server untuk menjalankan kode PHP, dan **MySQL** yang berfungsi sebagai sistem manajemen basis data untuk menyimpan seluruh data inventory.

Tabel 3.2. Daftar Perangkat Lunak dan Alat Komunikasi Pendukung

No	Nama Alat/Software	Kegunaan
1	Visual Studio Code	Editor andalan untuk menulis semua kode (PHP, HTML, CSS, JS).
2	XAMPP	Paket wajib untuk server lokal (Apache Web Server dan MySQL Database).
3	Google Chrome	Browser utama untuk mengecek hasil kodingan dan debugging.
4	Draw.io	Alat bantu berbasis web untuk membuat flowchart dan ERD.
5	Discord	Media diskusi teknis dan kolaborasi antar anggota tim developer.
6	WhatsApp	Alat komunikasi cepat untuk koordinasi harian dengan supervisor/mentor.
7	Google Meet	Platform tatap muka virtual untuk sesi Daily Standup Meeting dan evaluasi mingguan.

3.6 Kendala dan Solusi

Namanya proses belajar dan bekerja, pasti ada saja hambatannya. Selama magang, penulis menghadapi beberapa kendala teknis yang cukup memusingkan, tapi justru dari situlah penulis banyak belajar.

3.6.1 Kendala Teknis

Ada dua masalah utama yang sempat membuat pengembangan terhambat:

1. **Stok Barang Jadi Minus:** Di awal pengembangan, penulis lupa memikirkan validasi logika. Akibatnya, kalau user menginput jumlah barang keluar lebih banyak dari stok yang ada, database malah mencatat angka negatif.
2. **Upload File Sembarangan:** User seringkali tidak sengaja mengupload file PDF atau Word di kolom foto barang. Karena tidak ada filter, tampilan daftar barang jadi rusak (*broken image*).

3.6.2 Solusi Pemecahan Masalah

Setelah berdiskusi dengan mentor dan mencari referensi, penulis menerapkan solusi berikut:

1. **Validasi Logika (If-Else):** Penulis menambahkan pengecekan sederhana tapi ampuh di kode PHP. Sebelum transaksi diproses, sistem akan mengecek: *"Apakah jumlah minta > stok ada?"*. Jika ya, tolak. Jika tidak, lanjut. Masalah stok minus pun hilang.
2. **Cek Tipe File di Server:** Penulis menggunakan fungsi PHP bernama `finfo_file`. Fungsi ini bisa mengintip "isi dalam" file. Jadi, sistem disetting hanya mau menerima file berekstensi .jpg, .png, atau .webp. Selain itu, ditolak mentah-mentah.

```
if ($tipe === 'IN') {
    $stokBaru = $stokSekarang + $jumlah;
} else { // OUT
    if ($stokSekarang < $jumlah) {
        throw new Exception("Stok tidak cukup. Stok sekarang: $stokSekarang");
    }
    $stokBaru = $stokSekarang - $jumlah;
}
```

Gambar 3.12. Kode Program Validasi Stok

Sumber: Dokumen Internal Perusahaan

Penjelasan: Pada potongan kode di Gambar 3.12, penulis menerapkan logika pencegahan stok minus di sisi backend. Kode program melakukan perbandingan antara variabel `$stokSekarang` dan `$jumlahPermintaan`. Jika jumlah permintaan lebih besar, sistem akan melempar *Exception* (Peringatan) dan membatalkan proses penyimpanan ke database, sehingga integritas data stok tetap terjaga.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

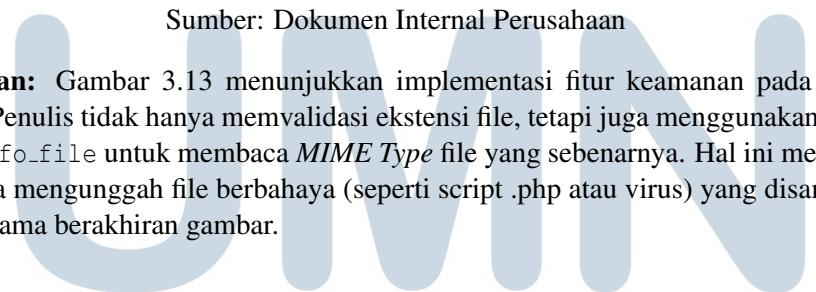


```
43     $ext = strtolower(pathinfo($_FILES['foto']['name'], PATHINFO_EXTENSION));
44     $allowedExt = ['jpg', 'jpeg', 'png', 'webp'];
45     if (!in_array($ext, $allowedExt, true)) {
46         die("Format foto tidak didukung. Gunakan JPG/PNG/WEBP.");
47     }
48
49
50     $finfo = new finfo(FILEINFO_MIME_TYPE);
51     $mime = $finfo->file($_FILES['foto']['tmp_name']);
52     $allowedMime = ['image/jpeg', 'image/png', 'image/webp'];
53     if (!in_array($mime, $allowedMime, true)) {
54         die("File bukan gambar yang valid.");
55     }
```

Gambar 3.13. Kode Program Validasi Upload Foto

Sumber: Dokumen Internal Perusahaan

Penjelasan: Gambar 3.13 menunjukkan implementasi fitur keamanan pada modul upload. Penulis tidak hanya memvalidasi ekstensi file, tetapi juga menggunakan fungsi PHP `finfo_file` untuk membaca *MIME Type* file yang sebenarnya. Hal ini mencegah pengguna mengunggah file berbahaya (seperti script .php atau virus) yang disamarkan dengan nama berakhiran gambar.



UNIVERSITAS
MULTIMEDIA
NUSANTARA