

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan kerja magang dilakukan di PT. Sannin Kreasi Indonesia pada Divisi IT dibawah pimpinan bapak Christian Sianturi selaku divisi program dan sekaligus Direktur perusahaan berperan sebagai *supervisor* lapangan selama masa magang dengan memberikan arahan teknis, supervisi kerja, serta pendampingan dalam setiap tahapan pengembangan sistem. Mahasiswa bekerja dalam sebuah tim pengembang yang terdiri dari tiga orang dengan tanggung jawab yang sama, yaitu mengembangkan sistem *Datalog Measuring* dan *Warehouse Management System (WMS)* Berbasis web. Seluruh anggota tim berkontribusi dalam proses perancangan, implementasi, hingga pengujian sistem secara kolaboratif.

Sebagai bentuk pengawasan dan pengendalian proyek, pelaporan progres dilakukan setiap hari Jumat. Sesi pelaporan dan evaluasi mingguan ini memiliki agenda strategis untuk melakukan verifikasi capaian pekerjaan terhadap target yang ditetapkan, mengidentifikasi secara proaktif permasalahan teknis yang muncul selama siklus pengembangan, serta memformulasi dan mengintegrasikan masukan konstruktif untuk iterasi perbaikan sistem. Proses ini krusial untuk memastikan bahwa produk akhir yang dihasilkan selaras secara optimal dengan target fungsional dan spesifikasi kebutuhan pengguna yang telah didefinisikan.

3.2 Tugas yang Dilakukan

Tugas yang dilakukan sebagai *Fullstack Developer* pada proses magang di PT. Sannin Kreasi Indonesia adalah sebagai berikut:

1. Membuat *Flowchart* untuk aplikasi *Datalog measuring*
2. Membuat *design* web menggunakan aplikasi Figma
3. Melakukan *coding* dengan *Laravel* dengan *React Typescript* dan *Next.js*
4. *Testing* aplikasi untuk evaluasi jika ada bug atau kerusakan

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan <i>meeting</i> dengan supervisor untuk membahas <i>project</i> serta membuat <i>flowchart</i> dan tampilan awal di Figma.
2	Membuat Komponen UI: <i>Sidebar</i> dan <i>Navbar</i> , serta membangun tampilan <i>layout dashboard</i> berdasarkan <i>role</i> (Admin dan Operator)
3	Membangun <i>Page History</i> , <i>Parameter</i> , dan <i>Measurement</i> . Menambahkan fitur filter pada halaman <i>History</i>
4	Menambahkan fitur filter tanggal, serta memperbaiki UI agar responsif dan menambahkan logo dan juga melakukan pengujian sistem dengan alat ukur digital.
5	Mempelajari konsep WMS dan membahas tentang <i>project</i> WMS serta membuat <i>flowchart</i>
6	Membuat <i>design dashboard</i> Operator(fitur <i>Receive</i> , <i>Move</i> , <i>Delivery</i> , <i>Return</i> , <i>Issues</i>)
7	Membuat halaman <i>dashboard</i> awal untuk <i>operator</i> gudang bahan baku dan setengah jadi.
8	Membuat <i>Layout</i> tampilan <i>operator outlet</i> serta melakukan pengujian awal pada fitur dasar mulai dari <i>login</i> hingga pengelolaan stok.
9	Membuat Komponen <i>form</i> untuk menerima barang(<i>receive</i>), Pindahan barang antar lokasi(<i>move</i>), barang untuk dikirim (<i>delivery</i>), barang yang akan dibalikkan(<i>return</i>).
10	Membuat halaman untuk mengerjakan <i>task</i> yang belum diambil(<i>issue</i>) serta menguji <i>page Order</i> dan <i>Receive</i> secara menyeluruh
11	<i>Meeting</i> dengan supervisor mengenai arahan dan melaporkan <i>update progress project</i> .

12	Memperbarui Halaman <i>dashboard</i> sesuai dengan skema model terbaru dan melakukan testing dari halaman yang <i>controller</i> sudah diperbarui
13	Merubah tampilan <i>form order</i> menjadi sebuah komponen dan menyesuaikan tampilan <i>delivery</i> dan <i>return</i> sesuai dengan tipe <i>warehouse</i> masing masing.
14	Melakukan Testing fitur <i>receive, move, delivery, return</i> dan <i>issue</i> sesuai dengan tipe <i>warehouse</i> .
15	Melakukan pengujian alur transaksi admin order to <i>delivery operator</i> mulai dari Admin order, <i>Operator accept order, Operator delivery</i> , hingga <i>receive</i> pada <i>warehouse</i> tujuan.
16-17	Menambahkan fitur Operator pada admin <i>dashboard</i> dan memperbaiki <i>bug</i> yang didapatkan

Dua proyek yang dikerjakan menggunakan kombinasi perangkat lunak dan perangkat keras. Berikut ini merupakan perangkat lunak yang digunakan dalam masing-masing proyek:

Datalog Measuring Menggunakan:

1. *Visual Studio Code* versi 1.100.1
2. *MySQL* versi 10.4.28-MariaDB
3. *Next.js Framework* versi 15.0.3
4. *next-auth Library* versi 5.0.0-beta.25
5. *Tailwind CSS* versi 3.4.1
6. *Zod Library* versi 3.23.8
7. *Prisma Library* versi 5.22.0

Warehouse Management System (WMS) Menggunakan:

1. *Visual Studio Code* versi 1.100.1
2. *XAMPP* versi 3.3.0

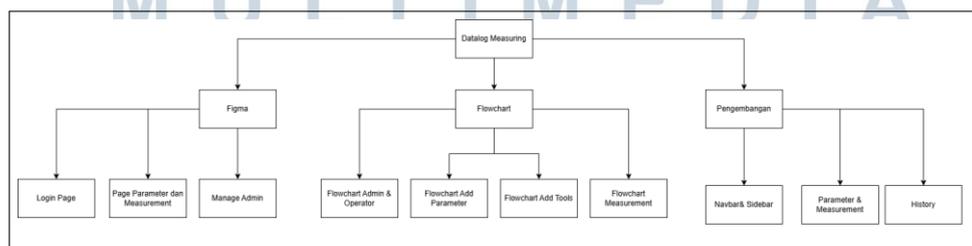
3. *MySQL* versi 10.4.28-MariaDB
4. *Laravel Framework* versi 12
5. *Tailwind CSS* versi 3.4.17
6. *Flowbite CSS* versi 3.1.2
7. *Shadcn UI* versi 2.5.0
8. *PostgreSQL Supabase Database*
9. *milon/barcode Library* versi 12.0
10. *TCPDF Library*
11. *Zod Library* versi 3.24.2

Berikut ini merupakan perangkat keras yang digunakan selama proses kerja magang adalah laptop *Legion 5 15ITH6H* dengan spesifikasi berikut:

1. *Processor* Intel I7-11800H
2. *RAM* 16 GB
3. *Storage* 1.5 TB

3.4 Perancangan *Datalog Measuring*

Datalog Measuring merupakan aplikasi *prototype* berbasis web yang dirancang untuk mencatat dan memantau hasil pengukuran produk dari proses produksi secara digital dan real-time. Aplikasi ini terhubung dengan perangkat *IoT* yang berfungsi sebagai alat ukur dan terintegrasi melalui koneksi *Bluetooth*. Data yang dikirimkan dari perangkat *IoT* akan secara otomatis masuk ke sistem dan tersimpan ke dalam basis data. Hal ini memudahkan proses pencatatan dan mengurangi risiko kesalahan input manual. Gambar 3.1 menampilkan pekerjaan yang dilakukan oleh mahasiswa pada *software datalog measuring*.

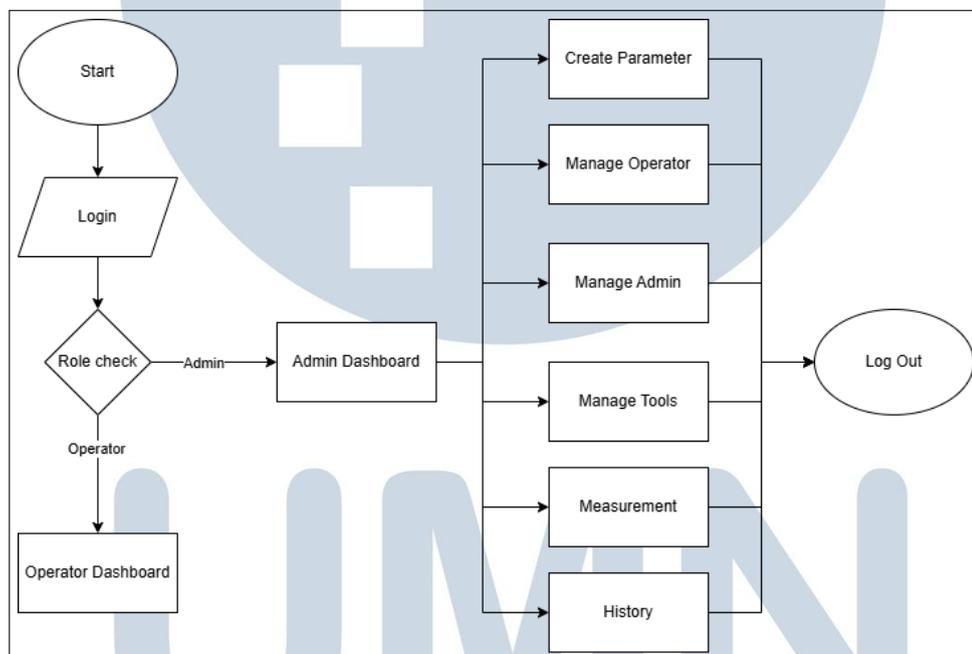


Gambar 3.1. *Bagan Pekerjaan yang dilakukan*

3.4.1 Flowchart Datalog Measuring

A Flowchart Admin

Flowchart Admin menggambarkan alur proses yang dilakukan oleh pengguna dengan peran sebagai Admin dalam sistem. Alur ini mencakup pengelolaan data pengguna (*user*), pengaturan alat pengukuran, serta pembuatan dan pengelolaan parameter pengukuran. Dengan *flowchart* ini, jalannya proses administratif dalam sistem dapat dipahami secara sistematis dan terstruktur, sehingga memudahkan Admin dalam mengelola berbagai aspek yang mendukung kegiatan pengukuran. *Flowchart* dapat dilihat pada gambar 3.2

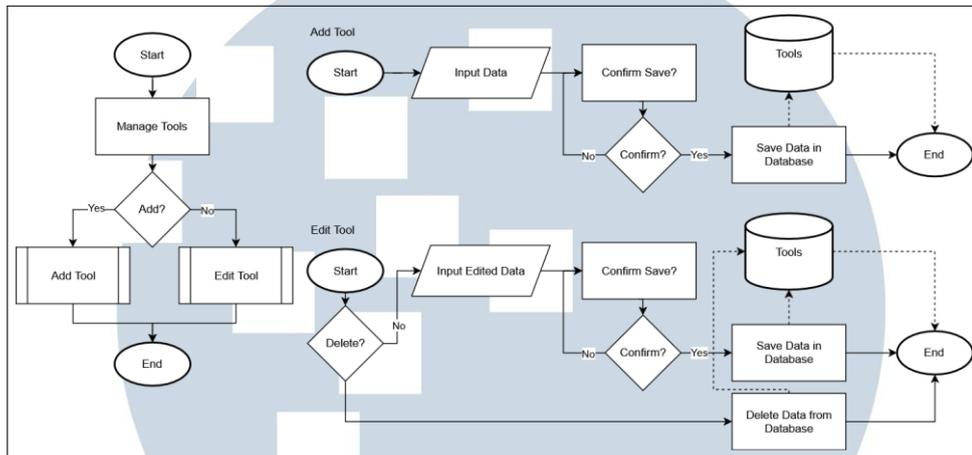


Gambar 3.2. *Flowchart Admin*

B Flowchart Pengelolaan Alat Pengukuran

Gambar 3.3 menggambarkan alur proses manajemen data alat (*tools*) oleh Admin dalam sistem *Datalog Measuring*. Proses dimulai dari pemilihan menu “*Manage Tools*” yang mengarahkan Admin untuk memilih tindakan penambahan (*Add Tool*) atau pengeditan (*Edit Tool*) data alat. Jika Admin memilih untuk menambahkan alat, maka data alat dimasukkan, dikonfirmasi, dan disimpan ke dalam basis data. Sementara itu, jika Admin memilih untuk mengedit data alat, proses dilanjutkan dengan input data yang telah diperbarui, konfirmasi penyimpanan, dan penyimpanan ke *database*. Pada tahap pengeditan juga

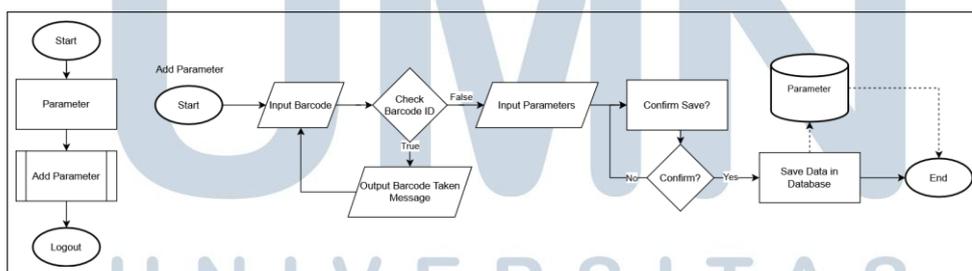
terdapat opsi untuk menghapus data alat, yang hanya dilakukan setelah mendapat konfirmasi. Setiap jalur berakhir setelah tindakan penyimpanan atau penghapusan berhasil dilakukan.



Gambar 3.3. Flowchart Pengelolaan Alat Pengukuran

C Flowchart Add Parameter

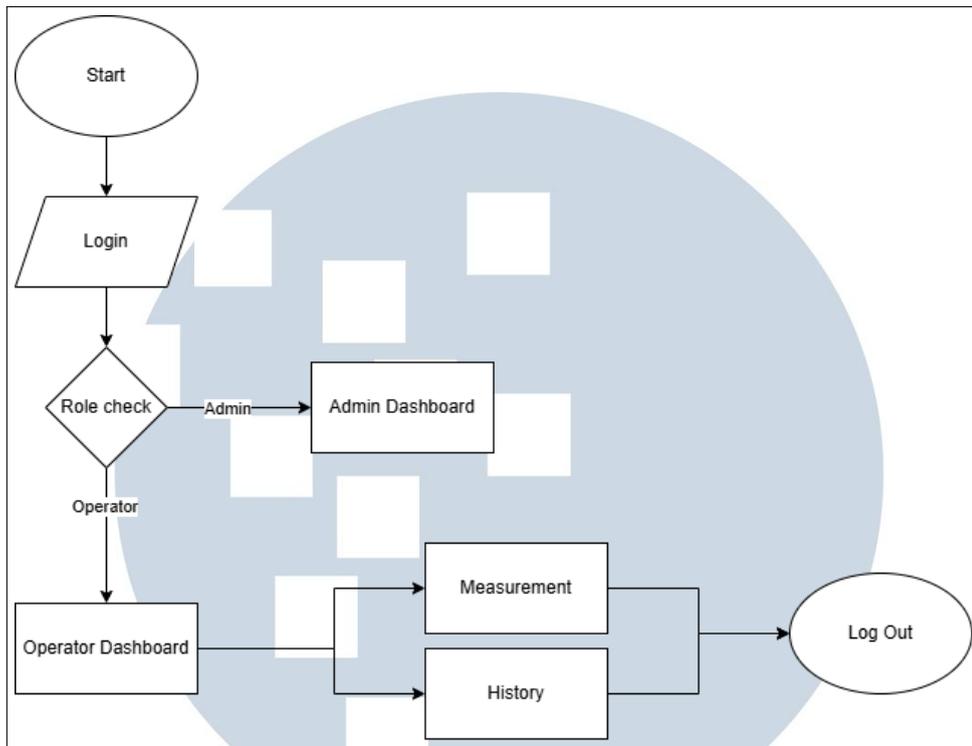
Gambar 3.4 menggambarkan alur penambahan parameter pengukuran dalam sistem berbasis *barcode* suatu barang. Proses dimulai dengan memasukkan *barcode*, kemudian sistem akan validasi ID *barcode* tersebut. Jika valid dan belum terdaftar dalam *database*, *Admin* dapat memasukkan parameter yang diperlukan dan data tersimpan dalam *database*.



Gambar 3.4. Flowchart Pengelolaan Parameter

D Flowchart Operator

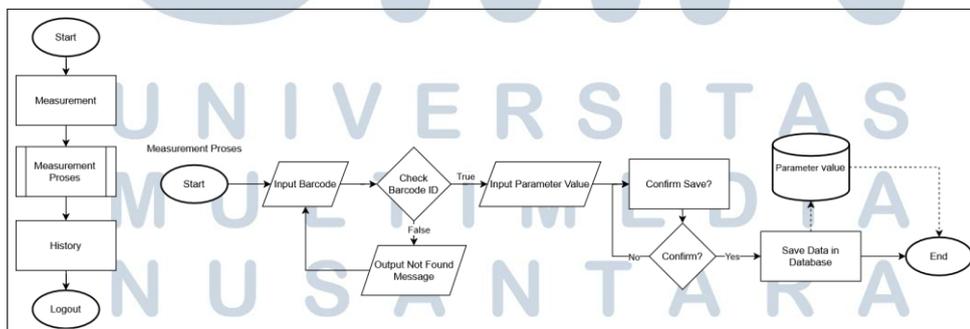
Flowchart Operator menggambarkan alur proses yang dilakukan oleh pengguna dengan peran sebagai *Operator* dalam sistem.



Gambar 3.5. *Flowchart* Umum Operator

E Flowchart Measurement

Gambar 3.6 ini mencakup input data hasil pengukuran, memverifikasi data yang telah dimasukkan, serta mengirimkan data tersebut ke dalam basis data. *Flowchart* ini memberikan gambaran sistematis mengenai langkah-langkah operasional yang harus diikuti oleh *Operator*, sehingga dapat memastikan bahwa data yang dikumpulkan valid, tersimpan dengan baik, dan mendukung kelancaran proses pemantauan serta analisis hasil pengukuran.



Gambar 3.6. *Flowchart* Pengelolaan Parameter

3.4.2 Perancangan *UI* dan Implementasi

Pada tahap ini perancangan antarmuka pengguna (*UI*) memegang peranan krusial untuk memastikan sistem yang dibangun tidak hanya fungsional tetapi juga mudah digunakan dan memiliki identitas visual yang kuat. Fokus mahasiswa pada tahap ini adalah Pengembangan *Layout* untuk setiap halaman dan pembuatan komponen seperti tabel dan filter yang terdapat di halaman *History*

A Design Figma

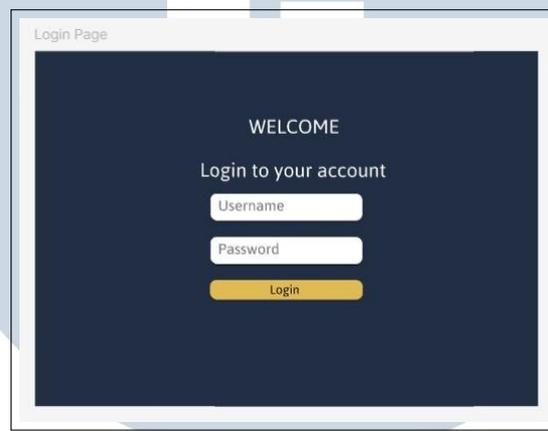
Proses perancangan tampilan aplikasi dimulai dengan penggunaan aplikasi *Figma*. *Figma* dipilih sebagai alat utama untuk pembuatan *mockup* tampilan. Pendekatan ini memungkinkan visualisasi awal dari antarmuka pengguna, yang menjadi panduan utama selama fase pengembangan *frontend*. Tujuan utama dari desain *Figma* ini adalah untuk menciptakan pengalaman pengguna (*user experience*) yang optimal dan kemudahan penggunaan bagi seluruh pengguna. Selain itu, desain juga difokuskan untuk merepresentasikan identitas visual PT Sannin Kreasi Indonesia (SNK), dengan mengadopsi skema warna yang konsisten dengan logo dan citra perusahaan (seperti yang terlihat pada gambar identitas SNK yang telah disediakan), yaitu dominasi warna biru gelap dan aksen warna emas. Identitas SNK terlihat dari gambar 3.7.



Gambar 3.7. Logo Perusahaan

B Login Page

Gambar 3.8 menunjukkan tampilan halaman *login* yang dirancang dengan antarmuka yang bersih dan fokus pada fungsionalitas utama. Desain ini mengadopsi skema warna SNK dengan dominasi biru gelap yang profesional dan kontras dengan elemen input. Halaman ini menyediakan dua bidang input utama untuk *Username* dan *Password*, serta tombol *Login* yang menonjol untuk mempermudah akses pengguna ke sistem.

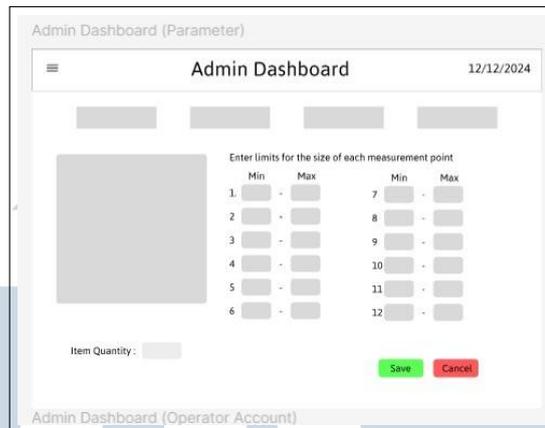


Gambar 3.8. Halaman Login Figma

C Measurement

Gambar 3.9 menyajikan tampilan halaman Parameter, yang merupakan *dashboard* bagi Admin untuk mengatur parameter pengukuran. Halaman ini memungkinkan Admin untuk mendefinisikan informasi seperti *Barcode ID*, Nama Barang, Tipe Barang, dan Gambar Barang. Selain itu, Admin dapat menetapkan nilai minimum dan maksimum yang diizinkan untuk setiap *parameter* pengukuran, serta menentukan alat ukur yang akan digunakan.

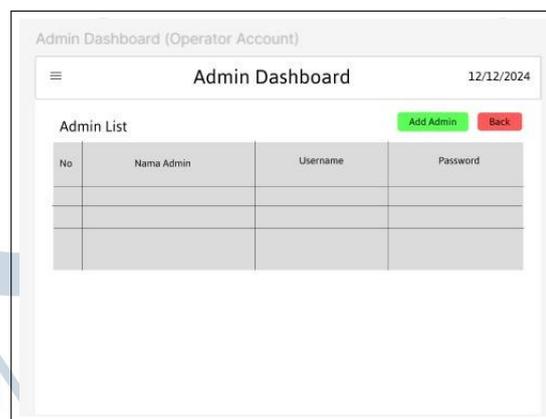
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.9. Halaman Measurement Figma

D Manage Admin

Gambar 3.10 menunjukkan tampilan halaman *Manage Admin*, yang berfungsi untuk pengelolaan akun dengan peran sebagai Admin. Halaman ini menyediakan fitur untuk menambah akun Admin baru, mengedit informasi Admin yang sudah terdaftar, dan menghapus akun Admin. *Layout* halaman ini, terutama penggunaan tabel untuk menampilkan daftar Admin, dirancang mirip dengan halaman pengelolaan *user* lainnya (seperti *Manage Operator*) untuk menjaga konsistensi *UI* dan mempermudah navigasi bagi pengguna



Gambar 3.10. Halaman Manage Admin Figma

Dalam proses desain di *Figma*, kami langsung membuat *mockup* tampilan. Meskipun tidak ada prinsip desain formal tertentu yang secara eksplisit diterapkan, proses ini bersifat iteratif. Kami merancang *mockup* terlebih dahulu, kemudian langsung melanjutkan ke tahap implementasi *frontend*. Umpan balik (*feedback*) dan penyesuaian desain sering kali terjadi selama proses implementasi *UI frontend*

berlangsung, memungkinkan kami untuk melakukan revisi dan penyempurnaan secara dinamis.

E Tampilan *Sidebar*

Gambar 3.11 menunjukkan tampilan navigasi *sidebar* yang dirancang untuk mempermudah pengguna dalam mengakses berbagai fitur utama pada sistem. Navigasi ini terdiri atas beberapa menu, yaitu Parameter, *Manage Operator*, *Manage Admin*, *Manage Tools*, *Measurement*, *History*, dan *Log Out*. Setiap menu disusun secara vertikal dan ditampilkan secara konsisten di sisi kiri antarmuka. Penggunaan *sidebar* ini bertujuan untuk meningkatkan efisiensi navigasi, menjaga konsistensi tampilan, serta memberikan struktur yang jelas dalam penggunaan sistem oleh pengguna.

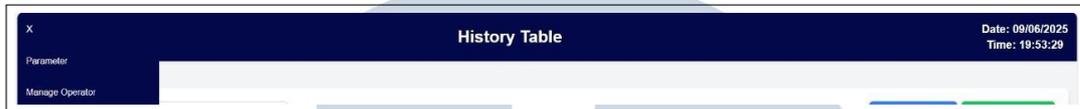


Gambar 3.11. Tampilan Sidebar Datalog Measuring

F Tampilan *Navbar*

Meskipun navigasi utama sistem diatur melalui *sidebar* untuk kemudahan akses dan konsistensi, elemen *navbar* (bilah navigasi atas) juga dirancang untuk melengkapi pengalaman pengguna dengan menyediakan informasi kontekstual dan

aksi cepat yang relevan. Gambar 3.12 menunjukkan implementasi *navbar* pada aplikasi *Datalog Measuring*.



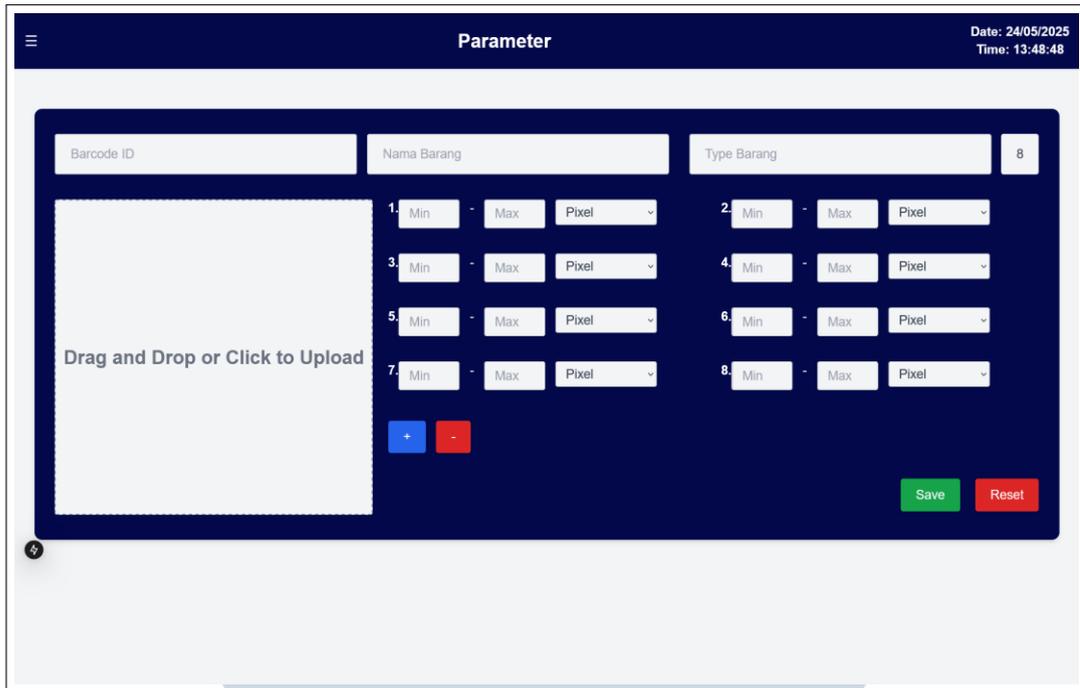
Gambar 3.12. Tampilan Navbar Datalog Measuring

Navbar Datalog Measuring secara khusus digunakan untuk mengidentifikasi halaman seperti "*History Table*" pada gambar 3.12, menampilkan waktu dan tanggal terkini, seperti "Date: 09/06/2025, Time: 19:53:29", yang penting untuk konteks data pengukuran dan histori, dan akses kepada *sidebar* dikiri atas berbentuk *hamburger menu* yang berfungsi sebagai kontrol untuk menampilkan atau menyembunyikan *sidebar*.

G Halaman Parameter

Gambar 3.13 menunjukkan tampilan halaman parameter yang berfungsi untuk mendefinisikan data awal terkait barang yang akan diukur. Pada halaman ini, pengguna dapat memasukkan informasi seperti *Barcode ID*, Nama Barang, Tipe Barang, Gambar Barang, serta menetapkan nilai maksimum dan minimum untuk setiap *parameter* pengukuran. Selain itu, *Admin* juga dapat menentukan alat ukur yang digunakan untuk masing-masing *parameter*. Desain ini mendukung pengelolaan parameter secara dinamis, di mana jumlah *parameter* dapat ditambahkan atau dikurangi sesuai kebutuhan. Dengan penyajian yang terstruktur dan fleksibel, halaman ini dirancang untuk memudahkan proses input data dan memastikan akurasi dalam pengaturan *parameter* pengukuran.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.13. Halaman Parameter

H Halaman *History*

Gambar 3.14 menunjukkan tampilan Halaman *History* yang dirancang untuk menampilkan riwayat hasil pengukuran barang yang telah dilakukan. Dalam pengerjaan halaman ini, fokus utama mahasiswa adalah pada pengembangan komponen tabel dinamis serta *search* bar di bagian atas tabel dan juga mengimplementasikan *date-picker*, yang secara signifikan meningkatkan kemampuan pengguna untuk mengelola dan menganalisis data.

Tabel pada halaman ini memuat detail informasi pengukuran seperti nama barang, nilai masing-masing parameter yang diukur, dan status hasil pengukuran. Status ini ditentukan secara otomatis oleh sistem dengan indikator "OK" jika seluruh nilai parameter berada dalam rentang batas atas dan bawah yang ditentukan, atau "NG" (*Not Good*) jika ditemukan setidaknya satu nilai parameter di luar rentang tersebut. Selain itu, tabel juga mencantumkan nama operator yang melakukan proses pengukuran, sehingga memudahkan pelacakan dan audit data.

Date	Operator	ID Barang	Name Barang	Tipe Barang	Overall Result	Detail
03/03/2025	Operator User	S001	Test Server Icon	Foto Server	OK	Hide Details
Parameter Details: • (Pixel) Min: 10.00 Max: 15.00 Value: 10.00 Status: OK • (Pixel) Min: 5.00 Max: 15.00 Value: 12.00 Status: OK • (Ruler) Min: 10.00 Max: 20.00 Value: 15.00 Status: OK						
03/03/2025	Operator User	S001	Test Server Icon	Foto Server	OK	Hide Details
Parameter Details: • (Pixel) Min: 10.00 Max: 15.00 Value: 10.00 Status: OK • (Pixel) Min: 5.00 Max: 15.00 Value: 12.00 Status: OK • (Ruler) Min: 10.00 Max: 20.00 Value: 15.00 Status: OK						
03/03/2025	Operator User	S001	Test Server Icon	Foto Server	OK	View Details
03/03/2025	Operator User	S001	Test Server Icon	Foto Server	OK	View Details
03/03/2025	Operator User	S001	Test Server Icon	Foto Server	NG	View Details
03/03/2025	Operator User	S001	Test Server Icon	Foto Server	NG	View Details
03/03/2025	Operator User	S001	Test Server Icon	Foto Server	NG	View Details
27/02/2025	Admin User	S001	Test Server Icon	Foto Server	NG	View Details

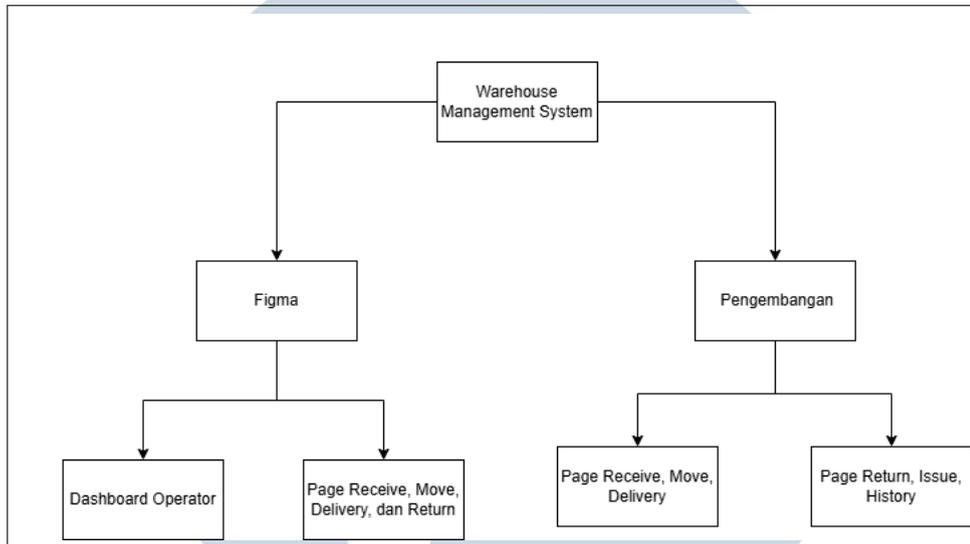
Gambar 3.14. Halaman *History Table* Datalog Measuring

3.5 Pengembangan *Warehouse Management System*

Warehouse Management System (WMS) merupakan proyek kedua yang dikerjakan selama kegiatan magang, setelah penyelesaian proyek *Datalog Measuring*. Proyek ini dikembangkan sebagai prototipe sistem pencatatan barang yang digunakan pada tiga jenis gudang: gudang bahan baku, gudang pabrik pengolahan barang setengah jadi, dan gudang *outlet*. Aplikasi ini merupakan *prototype* yang dibangun untuk membantu proses manajemen stok barang secara digital dan terstruktur, termasuk pencatatan masuk-keluar barang, pengelompokan berdasarkan jenis gudang, serta pelacakan histori pergerakan barang.

Dalam pengembangannya, aplikasi ini *WMS* secara keseluruhan menggunakan *framework Laravel* sebagai *backend* untuk mengelola logika bisnis dan basis data. Sementara itu, untuk sisi *frontend*, sistem ini mengadopsi kombinasi *Inertia.js* dengan *TypeScript* untuk membangun antarmuka pengguna yang dinamis dan interaktif. Dalam lingkup pekerjaan magang ini, fokus utama kontribusi saya adalah pada pengembangan sisi *frontend* dan *routing*. Hal ini mencakup perancangan dan pembuatan berbagai halaman antarmuka pengguna, penerapan *layouting* yang responsif, pengembangan komponen-komponen UI yang dapat digunakan kembali, serta implementasi *routing* untuk navigasi antar halaman yang mulus. Penggunaan kombinasi teknologi ini dipilih untuk mendukung pengembangan sistem yang bersifat modern, terintegrasi, dan mudah dikembangkan lebih lanjut. Selain itu, proyek ini juga menerapkan autentikasi pengguna dan pengelolaan *role* seperti *Admin* dan *Operator* di sisi *frontend*, dengan

menyediakan antarmuka untuk pengelolaan data pengguna, gudang, barang, dan histori aktivitas dalam sistem. Pada gambar 3.15 menggambarkan pekerjaan yang dilakukan mahasiswa pada pengembangan *software WMS*



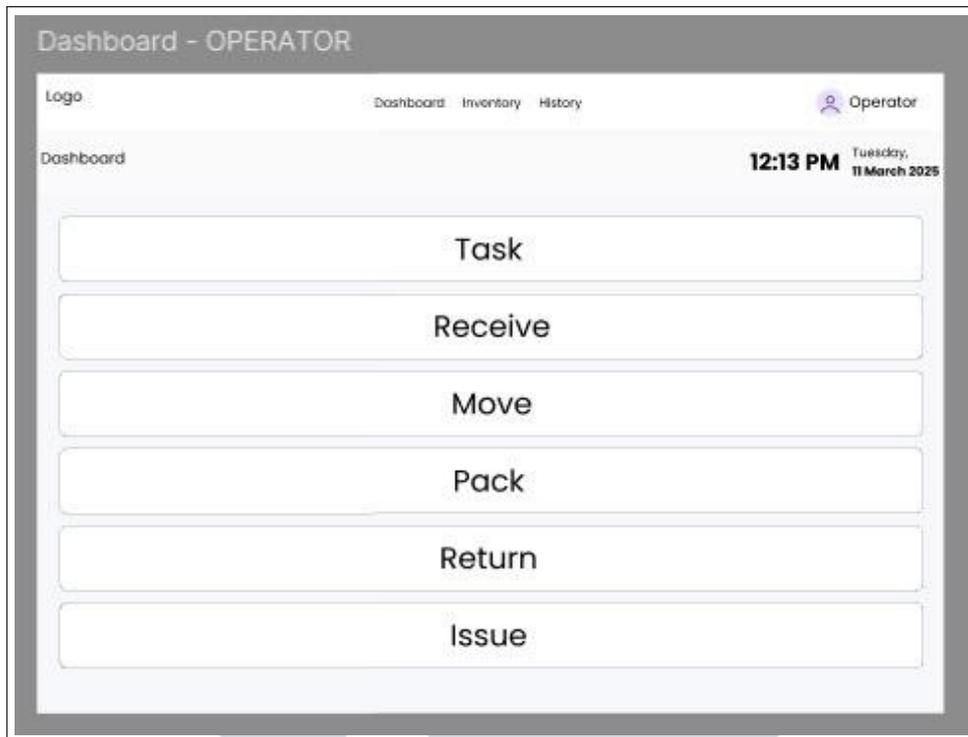
Gambar 3.15. Design *Bagan Pekerjaan yang dilakukan*

3.5.1 Design Figma WMS

A Dashboard Operator Figma

Gambar 3.16 adalah design Figma Dashboard operator dimana operator bisa melakukan pekerjaan yang bisa dilakukan yaitu, menerima barang di *page receive*, memindahkan barang di antar lokasi digudang *page move*, melakukan pengiriman untuk request barang dari gudang lainnya di *page Delivery*, melakukan pengembalikkan barang yang ada kekurangan dari antar gudang, mengambil pekerjaan yang harus dilakukan di *page issue*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.16. Design *Dashboard Operator*

B Receive Figma

Tampilan *form* untuk proses penerimaan barang, khususnya dari gudang bahan baku ke gudang setengah jadi atau dari gudang setengah jadi ke *outlet*. Desain ini berfokus pada penyediaan *layout form* untuk memasukkan detail barang yang diterima, termasuk nama produk, tipe, kuantitas, kategori, tanggal kedaluwarsa, dan deskripsi.

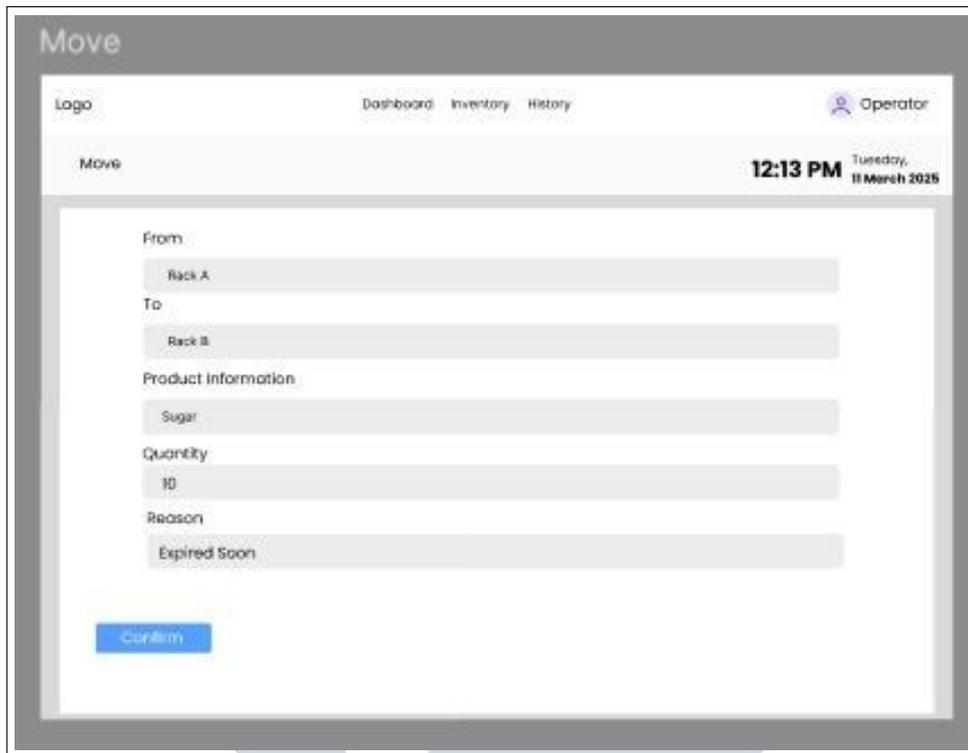
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Gambar 3.17. Design *Receive Operator*

C Move Figma

Tampilan *form* untuk proses pemindahan barang antar rak di dalam gudang. Desain ini memungkinkan Operator memilih lokasi asal (From) dan tujuan (To), serta memasukkan informasi produk, kuantitas, dan alasan pemindahan.

U M M N
 U N I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A

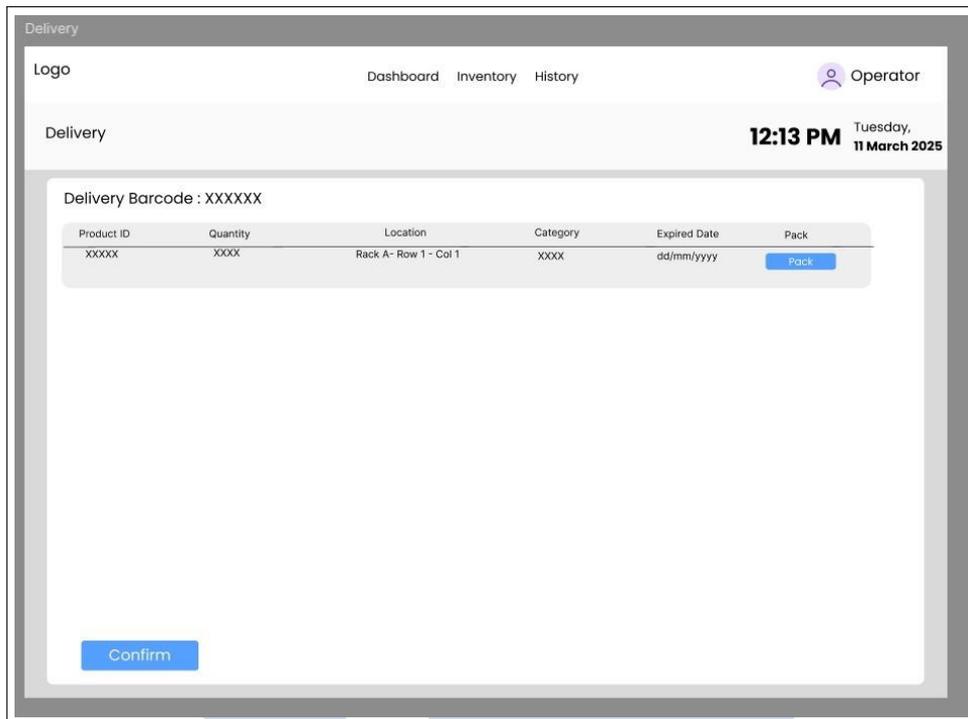


Gambar 3.18. Design *Move Operator*

D Delivery Figma

Tampilan *form* untuk proses pengiriman barang dari gudang bahan baku ke gudang setengah jadi atau dari gudang setengah jadi ke *outlet*, biasanya sebagai respons terhadap permintaan atau pesanan. Desain ini mencakup *layout* untuk memasukkan *delivery barcode* dan menampilkan daftar produk yang akan dikirim.

U M M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA

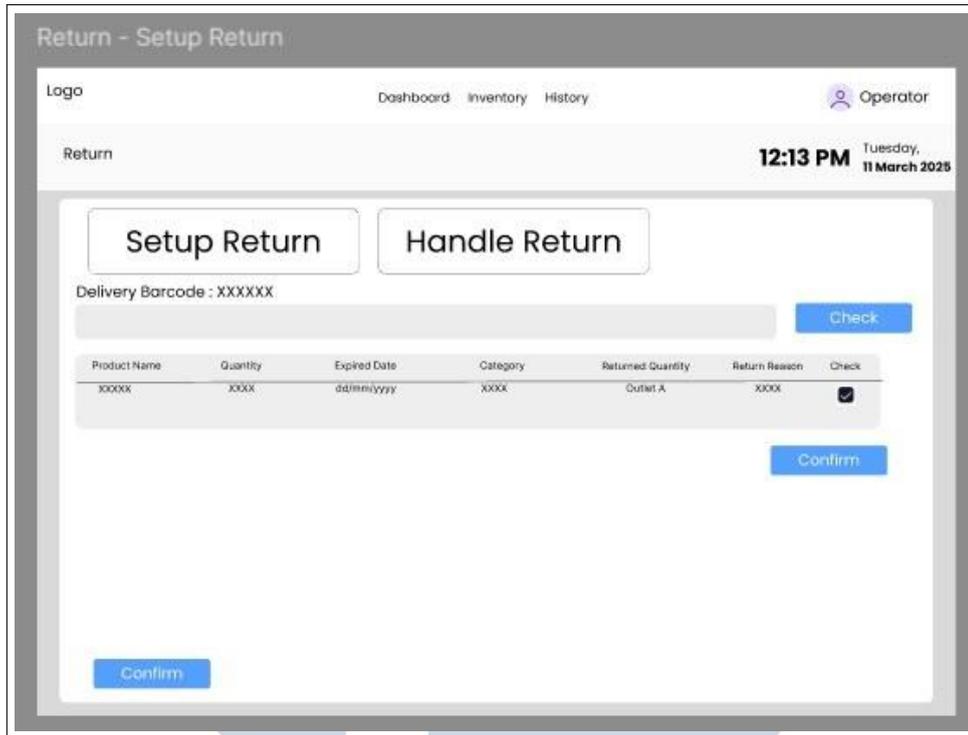


Gambar 3.19. Design *Delivery*

E Return Figma

Tampilan *form* untuk proses pengembalian barang yang rusak atau *reject* dari gudang setengah jadi ke gudang bahan baku. Desain ini mencakup opsi untuk *Setup Return* dan *Handle Return*, melibatkan *input delivery barcode* atau *return barcode* dan detail produk yang dikembalikan

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.20. Design *Return*

3.5.2 Implementasi Aplikasi WMS

A Halaman *Receive*

Gambar 3.21 menunjukkan implementasi Halaman *Receive*. Halaman ini dirancang sebagai *form* untuk penerimaan barang. Fungsi utamanya adalah memungkinkan Operator gudang untuk memasukkan detail barang yang diterima, khususnya saat menerima barang dari gudang bahan baku ke gudang setengah jadi ataupun dari gudang setengah jadi kepada gudang barang jadi.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.21. Receive Page

B Halaman Move

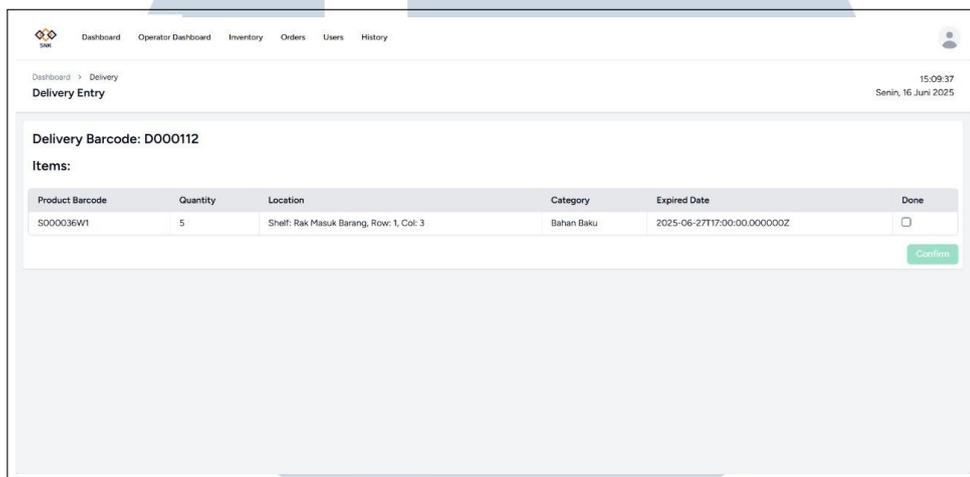
Gambar 3.22 menampilkan implementasi Halaman *Move*. Halaman ini berfungsi untuk Operator mengatur pemindahan barang dari satu rak ke rak lain di dalam gudang. Implementasi ini menyediakan *form* yang memungkinkan Operator memilih lokasi asal dan tujuan, serta memasukkan informasi produk dan kuantitas yang dipindahkan. *Layout* halaman ini dirancang untuk meminimalkan kesalahan input selama proses pemindahan.

Gambar 3.22. Move Page

C Halaman Delivery

Gambar 3.23 menunjukkan implementasi Halaman *Delivery*. Halaman ini dirancang untuk Operator dalam mengelola proses pengiriman barang, misalnya

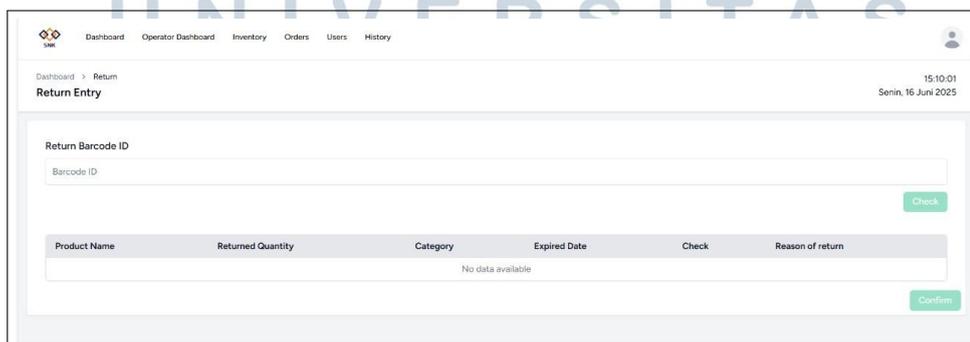
saat mengirimkan barang dari gudang bahan baku ke gudang setengah jadi sebagai respons terhadap permintaan dari gudang setengah jadi ataupun dari permintaan *outlet* ke gudang setengah jadi. Implementasi *frontend* di halaman ini mencakup *layout* untuk memasukkan *delivery barcode* dan menampilkan daftar produk yang akan dikirim, serta tombol konfirmasi untuk menandai selesainya proses pengiriman.



Gambar 3.23. Delivery Page

D Halaman Return

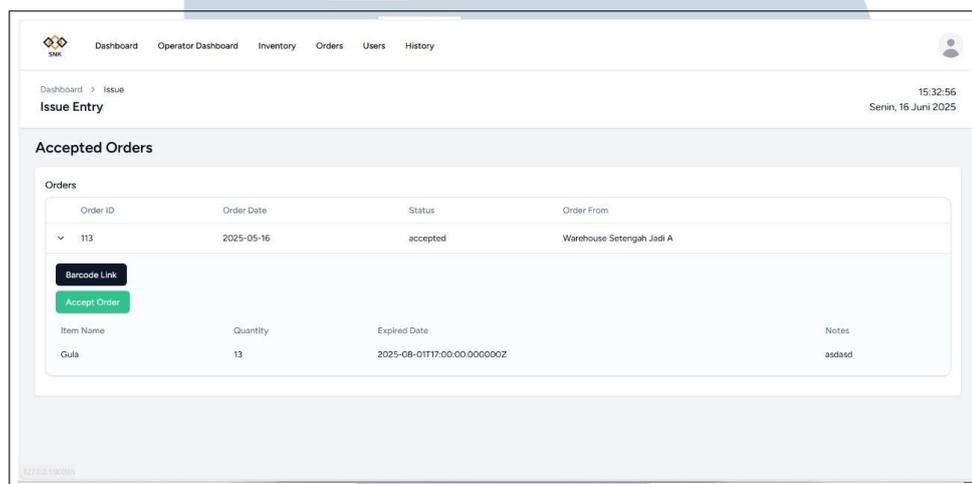
Gambar 3.24 menampilkan implementasi Halaman *Return*. Halaman ini digunakan oleh Operator untuk mencatat pengembalian barang, khususnya barang yang rusak atau *reject*, dari gudang setengah jadi kembali ke gudang bahan baku atau dari *outlet* ke gudang setengah jadi. Implementasi *frontend* pada halaman ini menyediakan *form* yang memungkinkan Operator memasukkan *barcode return* dan detail produk yang dikembalikan, memastikan proses retur terdokumentasi dengan baik.



Gambar 3.24. Return Page

E Halaman Issue

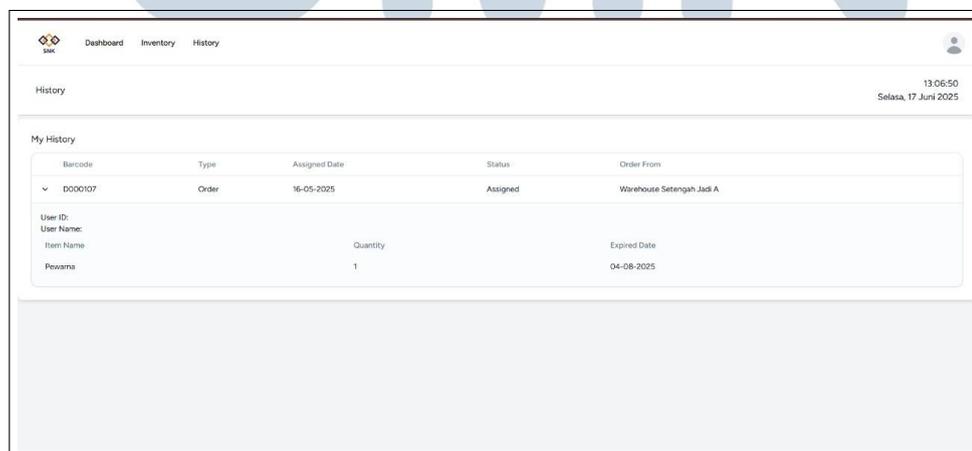
Gambar 3.25 menunjukkan implementasi Halaman *Issue*. Halaman ini berfungsi sebagai daftar tugas bagi Operator, menampilkan pesanan atau pekerjaan yang harus dikerjakan. Implementasi *frontend* di sini berfokus pada penyajian informasi yang jelas mengenai isu-isu yang perlu ditangani, memungkinkan Operator untuk memilih dan menandai pekerjaan yang sedang diproses.



Gambar 3.25. Issue Page

F Halaman History

Gambar 3.26 menunjukkan implementasi Halaman *History* yang dilakukan oleh operatornya sendiri. Halaman ini berfungsi untuk mencatat pekerjaan operator. Implementasi *frontend* disini menggunakan *library* dari *Shadcn Ui* yaitu tabel *dropdown* yang bisa menampilkan informasi lebih detail.



Gambar 3.26. History Operator Page

3.6 Kendala dan Solusi yang Ditemukan

3.6.1 Kendala yang ditemukan

Terdapat Kendala yang dialami selama pelaksanaan magang di PT.Sannin Kreasi Indonesia.

1. Kurangnya pemahaman tentang bahasa pemrograman yang digunakan sehingga dapat memperpanjang waktu pengembangan
2. Kurangnya pemahaman terhadap ruang lingkup (*scope*) aplikasi yang dirancang , terutama dalam hal perubahan atau penambahan fitur.

3.6.2 Solusi yang ditemukan

1. Mempelajari bahasa yang digunakan dengan *framework* yang berbeda.
2. Mengadakan diskusi bersama tim dan supervisi untuk memastikan *scope* terkait fitur yang akan ditambahkan atau diubah

