

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama menjalani program magang, mahasiswa ditempatkan di Divisi Program dengan peran sebagai *Front-end Developer*. Dalam melaksanakan tugasnya, mahasiswa berada di bawah bimbingan langsung Bapak Christian yang menjabat sebagai Direktur sekaligus berperan sebagai *supervisor* lapangan selama masa magang. Beliau memberikan arahan teknis, supervisi kerja, serta pendampingan dalam setiap tahapan pengembangan sistem.

Mahasiswa bekerja dalam sebuah tim pengembang yang terdiri dari tiga orang, yang secara umum memiliki tanggung jawab untuk mengembangkan sistem *Warehouse Management System (WMS)* berbasis web. Walaupun tugas utama mahasiswa berada pada sisi antarmuka pengguna, mahasiswa tetap dilibatkan dalam proses diskusi perancangan sistem guna mendukung sinergi tim dalam membangun sistem secara menyeluruh.

Sebagai bentuk pengawasan dan pengendalian proyek, pelaporan progres dilakukan setiap hari Jumat. Evaluasi mingguan ini bertujuan untuk meninjau capaian pekerjaan, mengidentifikasi permasalahan teknis yang muncul, serta memberikan masukan atau perbaikan terhadap sistem yang sedang dikembangkan agar hasil akhir sesuai dengan target dan kebutuhan yang telah ditentukan.

3.2 Tugas yang Dilakukan

Selama masa magang di PT Sannin Kreasi Indonesia, mahasiswa ditempatkan sebagai *Front-end Developer* yang bertanggung jawab dalam pengembangan antarmuka pengguna dari aplikasi *Warehouse Management System (WMS)* berbasis web. Seluruh aktivitas dilaksanakan di bawah arahan Bapak Christian selaku Direktur sekaligus supervisor, dengan teknologi inti Laravel untuk *backend* dan React TypeScript untuk *front-end*. Walaupun berfokus pada sisi *front-end*, mahasiswa juga turut membantu dalam pengembangan sistem. Adapun tugas yang dilakukan selama magang adalah sebagai berikut:

1. Membuat desain antarmuka menggunakan aplikasi Figma.
2. Menyusun *flowchart* alur kerja untuk peran admin dan operator.

3. Mengembangkan antarmuka pengguna menggunakan React dengan TypeScript.
4. Melakukan *testing* aplikasi untuk menemukan dan memperbaiki *bug* atau kerusakan.

3.3 Uraian Pelaksanaan Magang

Bagian ini memuat penjelasan mengenai pelaksanaan kegiatan magang. Uraian mencakup rincian pekerjaan yang telah diselesaikan, beserta kendala atau permasalahan yang ditemui selama kegiatan kerja magang.

3.4 Proses Pelaksanaan Magang

Pelaksanaan program kerja magang dilakukan selama empat bulan atau 17 minggu. Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu

Minggu Ke	Pekerjaan yang dilakukan
1	<ul style="list-style-type: none"> • Membahas proyek pembuatan aplikasi <i>Data Measuring Logger</i> berbasis web. • Membuat desain tampilan.
2	<ul style="list-style-type: none"> • Membuat <i>landing page</i> dan halaman <i>login</i>. • Membuat desain untuk <i>NavBar</i> dan <i>SideBar</i>. • Membuat <i>template</i> untuk halaman Parameter dan <i>Measurement</i>. • Mengembangkan halaman untuk mengelola akun operator dan <i>tools</i>.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu Ke	Pekerjaan yang dilakukan
3	<ul style="list-style-type: none"> • Melanjutkan pengerjaan halaman Parameter. • Membuat halaman <i>History</i> untuk data dari halaman <i>Measurement</i>. • Membuat sistem filter untuk halaman <i>Measurement</i>. • Membuat fitur detail barang yang sudah diukur.
4	<ul style="list-style-type: none"> • Membuat filter tanggal menggunakan React Date Picker. • Menambahkan logo dan menyesuaikan tampilan responsif. • Melakukan uji coba aplikasi dengan alat ukur.
5	<ul style="list-style-type: none"> • <i>Meeting</i> pembahasan proyek <i>WMS</i>. • Perancangan penggunaan fitur, pembagian <i>role</i>, dan <i>database</i>. • Menyusun <i>Flowchart</i> untuk alur kerja Operator dan Admin.
6	<ul style="list-style-type: none"> • Membuat <i>landing page</i> dan halaman <i>login</i>. • Membuat desain tampilan untuk Admin di Figma.
7	<ul style="list-style-type: none"> • Membuat <i>Dashboard</i> untuk Admin gudang bahan baku dan gudang setengah jadi.
8	<ul style="list-style-type: none"> • Menyelesaikan tampilan <i>Dashboard</i> untuk Admin gudang <i>outlet</i>. • Membuat tampilan <i>Storage Management</i>. • Pengujian terhadap fitur yang sudah dikembangkan, mulai dari <i>login</i> sampai manajemen gudang.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu Ke	Pekerjaan yang dilakukan
9	<ul style="list-style-type: none"> • Membuat halaman untuk menerima barang (<i>receive</i>), pemindahan lokasi barang (<i>move</i>), pengiriman barang (<i>delivery</i>), dan pengembalian barang (<i>return</i>).
10	<ul style="list-style-type: none"> • Membuat halaman untuk manajemen <i>users</i>. • Membuat <i>form</i> untuk penambahan jumlah <i>users</i>. • Membuat <i>form</i> untuk menghapus dan mengubah informasi <i>users</i>.
11	<ul style="list-style-type: none"> • <i>Meeting</i> dengan <i>supervisor</i> untuk pengarahannya proyek. • Membuat halaman Admin Order. • Membuat halaman <i>History</i> untuk melacak pesanan.
12	<ul style="list-style-type: none"> • Menyesuaikan tampilan <i>receive</i> dan <i>move</i> sesuai dengan tipe <i>warehouse</i> masing-masing.
13	<ul style="list-style-type: none"> • Menyesuaikan halaman <i>history</i> untuk <i>user</i> Admin dan Operator karena ada penyesuaian ulang tabel dalam <i>database</i>.
14	<ul style="list-style-type: none"> • Melakukan <i>testing</i> fitur <i>receive</i>, <i>move</i>, <i>delivery</i>, <i>return</i>, dan <i>issue</i> untuk setiap gudang. • Melakukan pengujian alur transaksi admin order to <i>delivery</i> operator mulai dari Admin order, Operator <i>accept</i> order, Operator <i>delivery</i>, hingga <i>receive</i> pada <i>warehouse</i> tujuan.
15	<ul style="list-style-type: none"> • Mencari dan memperbaiki <i>bug</i> fitur order, <i>move</i>, dan <i>history</i>.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu Ke	Pekerjaan yang dilakukan
16–17	<ul style="list-style-type: none"> • Menambahkan fitur operator pada admin <i>dashboard</i>. • Memperbaiki fitur <i>return</i> barang.

Perangkat lunak yang digunakan selama proses pengembangan *WMS* adalah sebagai berikut:

1. *Visual Studio Code* versi 1.100.1
2. *Laravel Framework* versi 12
3. *Tailwind CSS* versi 3.4.17
4. *Flowbite CSS* versi 3.1.2
5. *Shadcn UI* versi 2.5.0
6. *PostgreSQL Supabase Database*
7. *milon/barcode Library* versi 12.0
8. *TCPDF Library*
9. *Zod Library* versi 3.24.2

Perangkat keras yang digunakan selama proses pengembangan *WMS* adalah Laptop ASUS TUF GAMING F15 FX506HC.FX506HC dengan spesifikasi sebagai berikut:

1. *Processor* 11th Gen Intel(R) Core(TM) i5-11400H
2. *RAM* 16 GB

3.5 Pengembangan Aplikasi Warehouse Management System

Warehouse Management System (*WMS*) merupakan sebuah aplikasi *prototype* berbasis web yang dikembangkan secara tim dengan menggunakan *Framework Laravel* versi 12 pada sisi *back-end* dan *React TypeScript* pada sisi *front-end*. Sistem ini dirancang untuk mencatat aktivitas keluar masuk barang di dalam gudang secara otomatis, serta memungkinkan pemantauan data secara *real-time* guna mendukung efisiensi dan akurasi dalam proses manajemen pergudangan.

3.5.1 Flowchart

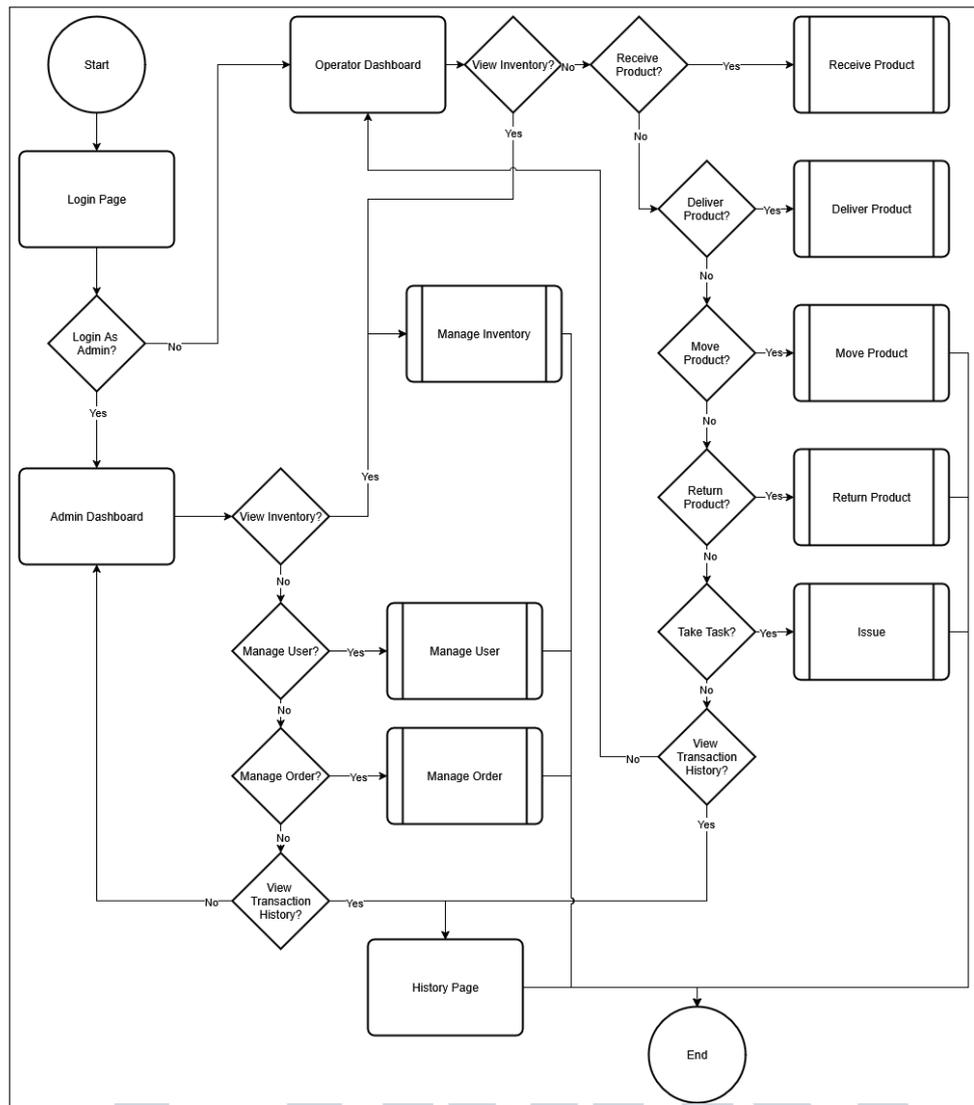
Dalam proses pengembangan aplikasi *Warehouse Management System* berbasis web, langkah awal yang dilakukan adalah merancang seluruh alur kerja melalui pembuatan *flowchart*. *Flowchart* tersebut menjadi panduan utama untuk menggambarkan rangkaian aktivitas yang dilakukan oleh pengguna dalam sistem, baik untuk peran administrator maupun operator. Dengan adanya perancangan ini, tim pengembang dapat memperoleh pemahaman yang utuh terhadap skenario penggunaan dan memastikan bahwa setiap fitur yang dibangun telah sesuai dengan kebutuhan operasional di lapangan. Pendekatan ini turut berperan dalam mengurangi potensi kesalahan logika serta mempermudah tahap implementasi dan pengujian sistem.

A Flowchart Utama

Gambar 3.1 menunjukkan *flowchart* utama dari alur proses penggunaan *Warehouse Management System (WMS)*. Proses diawali dari halaman *login*, diikuti dengan tahapan verifikasi *role* pengguna. Pengguna dengan *role* admin akan diarahkan ke halaman *dashboard* admin, sedangkan pengguna dengan *role* operator akan diarahkan ke halaman *dashboard* operator.

Pengguna dengan *role* admin memiliki akses ke halaman *inventory*, *order*, *user*, dan *history*. Setiap halaman tersebut memiliki fungsi strategis dalam pengelolaan gudang, mulai dari manajemen pengguna, pengelolaan persediaan barang di gudang, pengaturan pesanan antar tipe gudang, hingga pemantauan histori pergerakan barang.

Sementara itu, pengguna dengan *role* operator memiliki akses ke halaman *inventory* serta *dashboard* operator yang menyediakan menu seperti *receive product*, *delivery product*, *move product*, *return product*, dan *issue*. Setiap menu tersebut memiliki peran penting dalam mendukung kelancaran operasional gudang.



Gambar 3.1. Flowchart utama

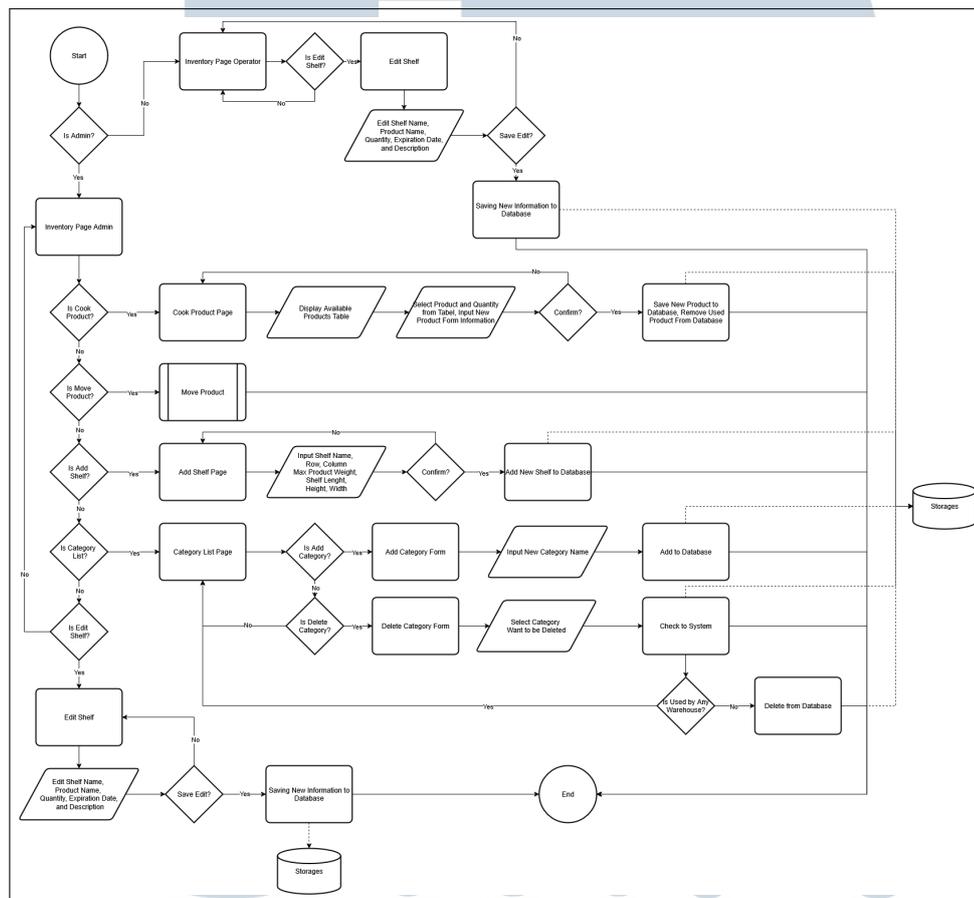
B Flowchart Manage Inventory

Gambar 3.2 menunjukkan *flowchart* dari proses *manage inventory*. Pengguna dengan *role operator* dapat melihat *inventory* dan memiliki akses terhadap fitur *edit shelf*. Proses *edit shelf* melibatkan pengisian *form* untuk memperbarui informasi rak, dan setelah melakukan *save*, data baru akan diperbarui ke dalam *database*.

Pengguna dengan *role admin* memiliki akses terhadap beberapa proses, antara lain *add shelf* yang melibatkan pengisian *form* penambahan rak baru, *move product* yang memerlukan pengisian *form* perpindahan barang, serta *category*

list yang menampilkan daftar kategori dengan opsi untuk menambahkan atau menghapus kategori. Seluruh proses yang disimpan melalui tombol *save* akan ditulis ke dalam *database*.

Sementara itu, gudang bahan setengah jadi memiliki satu proses tambahan, yaitu memasak produk, yang tidak dimiliki oleh tipe gudang lainnya. Proses ini memerlukan pengisian *form* masak produk, dan informasi produk yang telah dimasak akan disimpan ke dalam *database*.

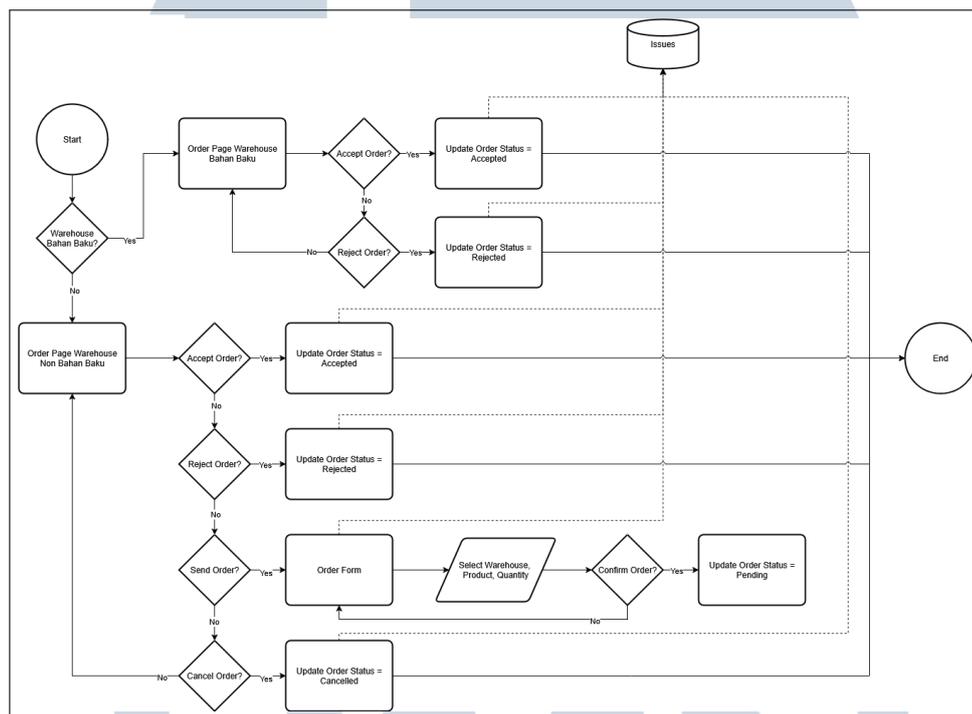


Gambar 3.2. Flowchart manage inventory

C Flowchart Manage Order

Gambar 3.3 menunjukkan proses dalam pengelolaan *order*. Proses ini dibagi menjadi dua berdasarkan jenis gudang, yaitu gudang bahan baku dan gudang non-bahan baku. Gudang bahan baku hanya memiliki kewenangan untuk melakukan *accept* dan *reject order*, karena tidak dapat mengajukan *order* secara langsung melalui sistem.

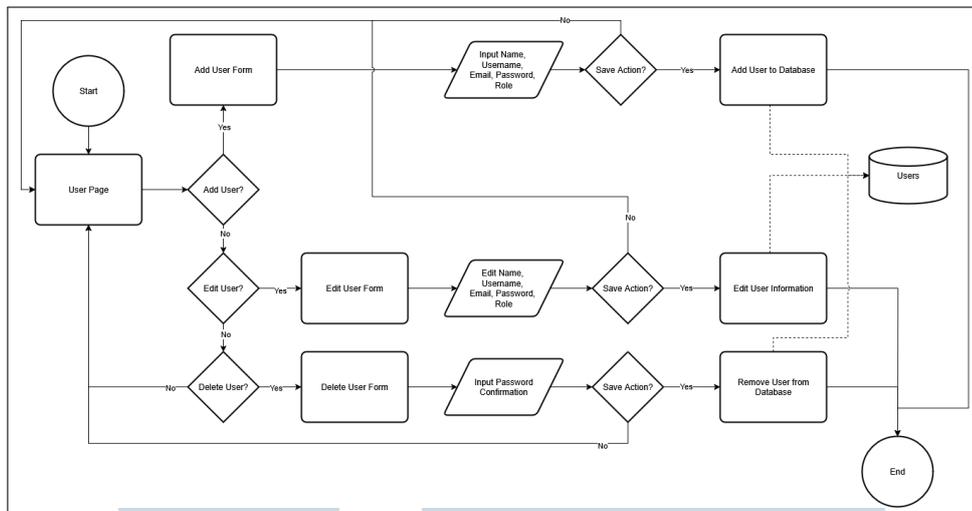
Sementara itu, gudang non-bahan baku memiliki akses terhadap lebih banyak aksi, yaitu *accept order*, *reject order*, *cancel order*, dan *send order* dengan terlebih dahulu mengisi *form order*. Admin perlu memilih barang apa saja yang akan dipesan, mengisi kuantitas produk yang dipesan, dan menuliskan catatan untuk setiap produk yang dipesan secara opsional. Setelah seluruh proses dikonfirmasi, status pada *database* akan diperbarui sesuai dengan tindakan yang diambil.



Gambar 3.3. Flowchart manage order

D Flowchart Manage User

Gambar 3.4 menunjukkan *flowchart user* yang menampilkan tabel *user* saat halaman pertama kali dibuka. Halaman *user* memiliki tiga menu utama, yaitu *add user* untuk menambahkan pengguna baru ke dalam tabel dengan mengisi data pada *form add user*, *edit user data* dengan mengisi *form edit user data*, serta menghapus pengguna dari daftar dengan melakukan konfirmasi melalui *input password*. Setelah setiap proses dilakukan, data akan disimpan ke dalam *database user*.

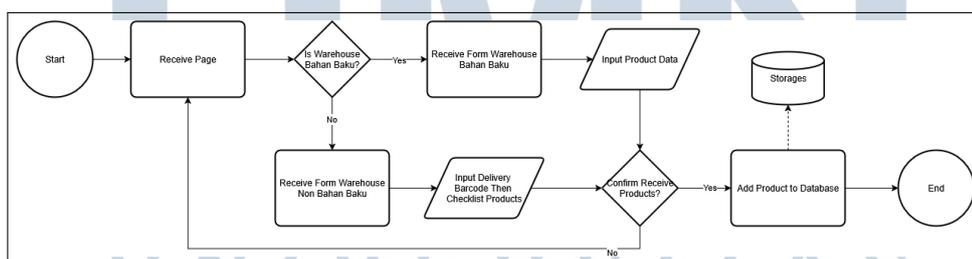


Gambar 3.4. *Flowchart manage user*

E Flowchart Receive Product

Gambar 3.5 menunjukkan proses penerimaan barang yang dibagi menjadi dua tipe gudang, yaitu gudang bahan baku dan gudang non-bahan baku. Pada gudang bahan baku, sistem menampilkan sebuah *form* penerimaan yang harus diisi dengan informasi produk secara manual.

Sementara itu, pada gudang non-bahan baku, proses penerimaan dilakukan melalui pengisian *form* dengan melakukan *input barcode*. Setelah seluruh proses dikonfirmasi dan dilakukan *save*, data akan disimpan ke dalam *database inventory*.

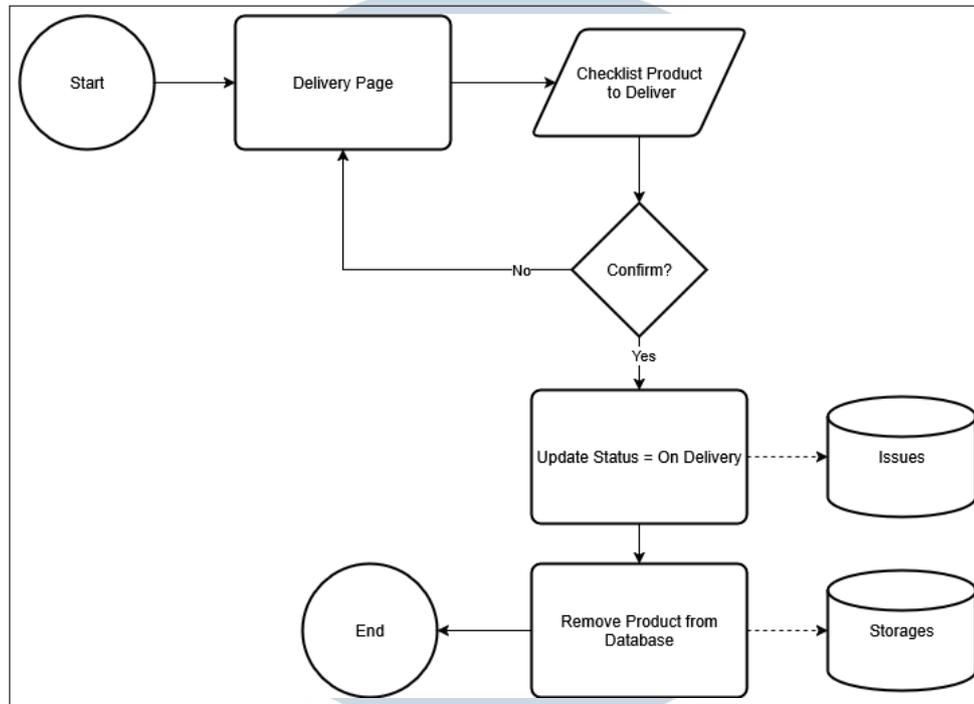


Gambar 3.5. *Flowchart receive product*

F Flowchart Delivery Product

Gambar 3.6 menunjukkan proses pengiriman produk yang telah dipesan. Pengguna akan ditampilkan *form delivery* dan diminta untuk mengisi data seperti nama produk dan kuantitas. Setelah proses dikonfirmasi, akan muncul notifikasi

barang akan segera dikirim dan data pengiriman beserta statusnya akan disimpan ke dalam *database issue*.

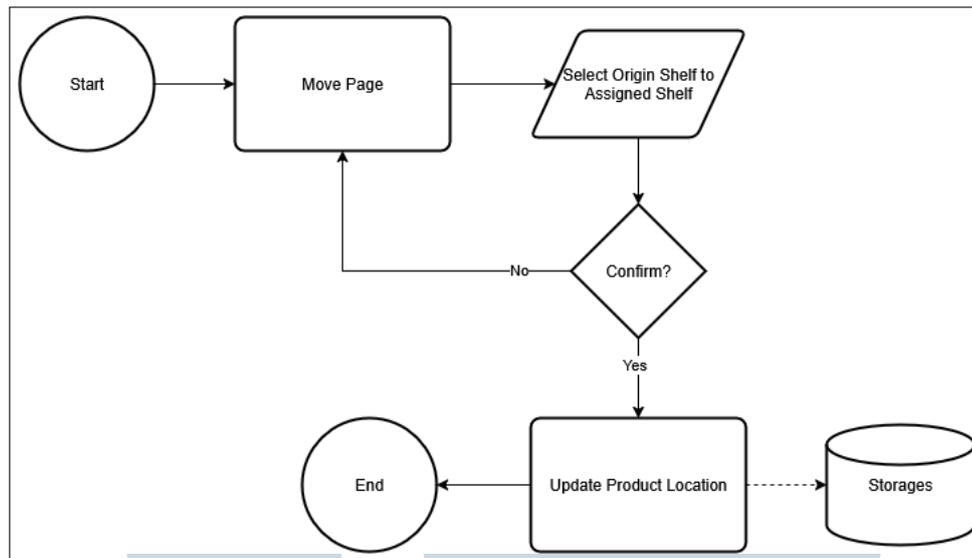


Gambar 3.6. *Flowchart delivery product*

G Flowchart Move Product

Gambar 3.7 menunjukkan proses perpindahan barang yang dilakukan untuk penataan ulang atau sekadar memindahkan barang antar lokasi. Proses dimulai dengan memilih barang dalam rak, kemudian memilih lokasi rak tujuan yang telah disediakan oleh sistem. Setelah proses disimpan melalui tombol *save*, data akan diproses oleh sistem dan disimpan ke dalam *database inventory*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.7. Flowchart move product

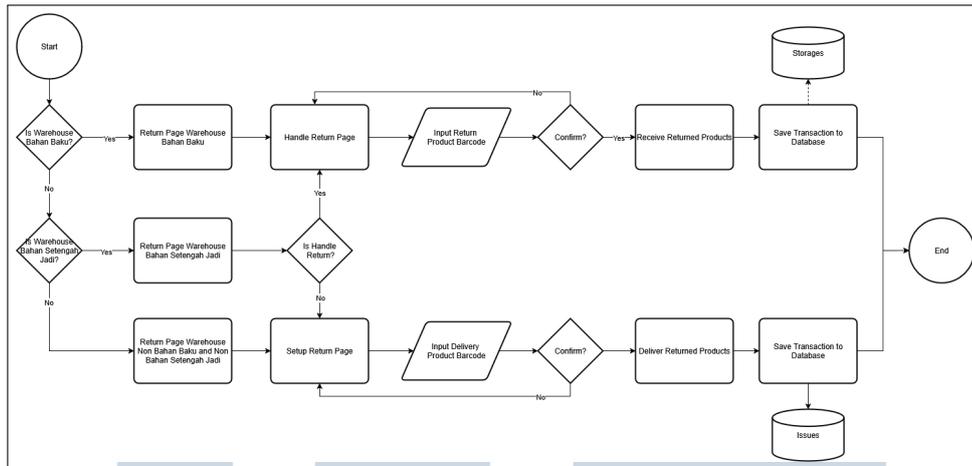
H Flowchart Return Product

Gambar 3.8 menunjukkan alur pengembalian produk. Proses pengembalian barang dibagi menjadi tiga tipe gudang, yaitu gudang bahan baku, gudang bahan setengah jadi, dan gudang outlet. Masing-masing gudang memiliki batasan akses terhadap proses yang tersedia.

Gudang bahan baku hanya dapat melakukan *handle return*, yaitu menangani produk yang dikembalikan. Admin akan ditampilkan daftar produk yang dikembalikan setelah melakukan *input barcode return*. Selanjutnya, admin melakukan *checklist* terhadap produk yang dikembalikan, dan sistem akan secara otomatis menentukan posisi rak tempat penyimpanan produk tersebut.

Sebaliknya, gudang outlet hanya memiliki akses untuk melakukan *setup return*, yaitu proses pengembalian produk. Proses dimulai dengan *input delivery barcode*, lalu sistem akan menampilkan daftar produk. Admin melakukan *checklist* produk dan mengisi jumlah yang akan dikembalikan. Setelah proses dikonfirmasi, sistem akan memperbarui status pada *database issue*.

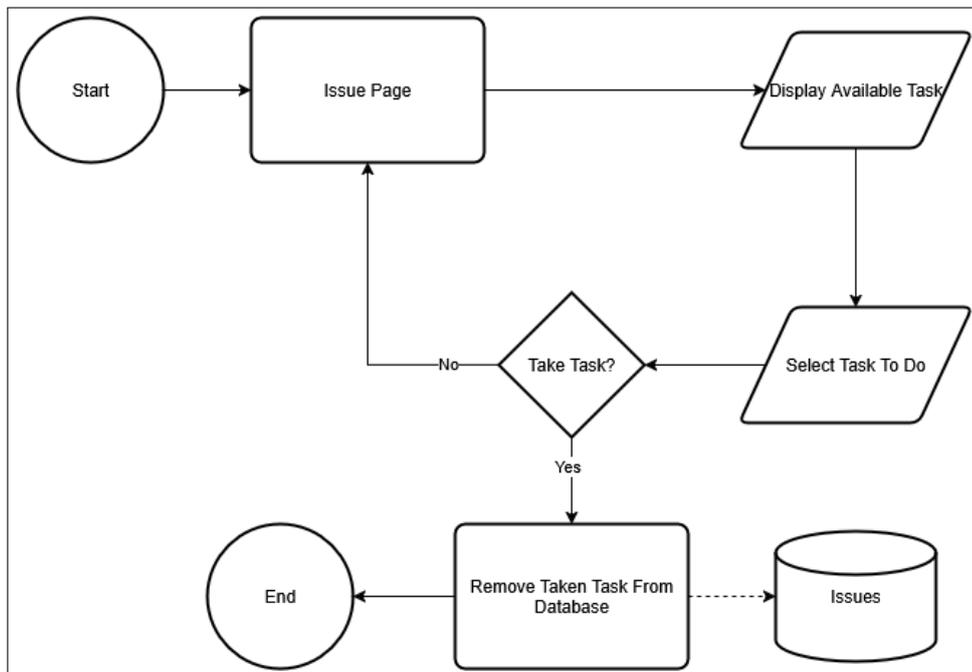
Sementara itu, pada gudang bahan setengah jadi, proses dimulai dengan memilih antara *handle return* atau *setup return*. Mekanisme dari kedua proses tersebut serupa dengan proses yang dilakukan di gudang bahan baku dan gudang outlet.



Gambar 3.8. Flowchart return product

I Flowchart Issue

Gambar 3.9 menunjukkan alur pengelolaan *issue* atau tugas yang belum diambil oleh operator. Operator akan memilih salah satu dari beberapa tugas yang tersedia dalam daftar *issue*, dan setelah dilakukan konfirmasi, sistem secara otomatis akan memperbarui status pada *database issue* serta menghasilkan *form delivery*.



Gambar 3.9. Flowchart issue

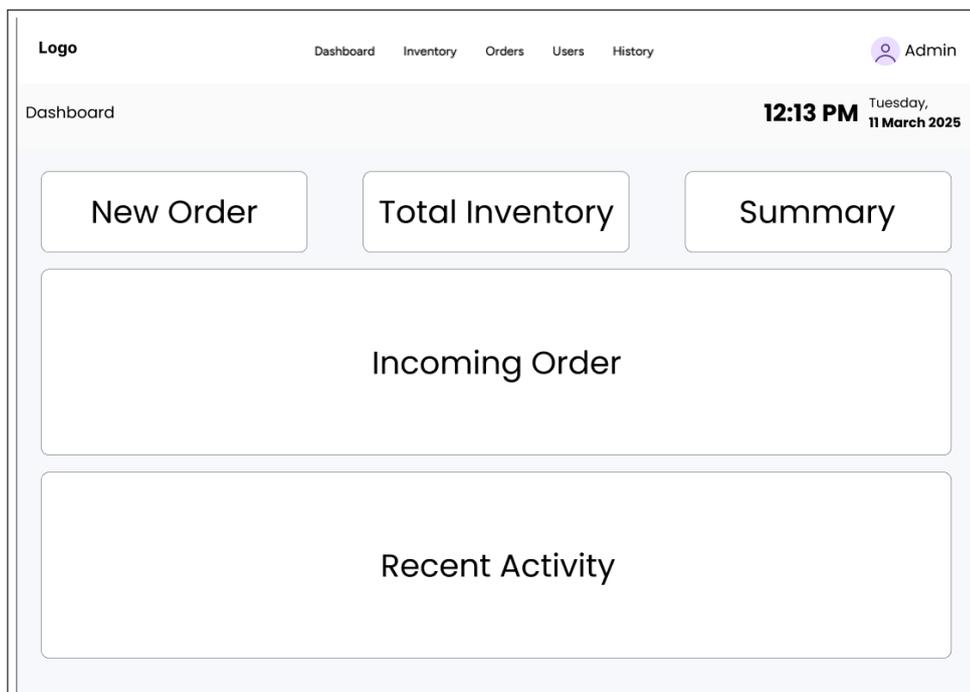
3.5.2 Desain Figma

Bagian ini akan dijabarkan desain keseluruhan tampilan dari aplikasi WMS berbasis web yang dibuat menggunakan aplikasi Figma. Desain ini dibuat sesuai dengan kebutuhan aplikasi dan untuk mempermudah dalam proses implementasi.

A Desain Halaman Admin Dashboard

Halaman *Admin Dashboard* terdapat sebuah *Navigation Bar* yang menampilkan pilihan beberapa halaman yang dapat diakses oleh Admin. Halaman yang bisa diakses oleh Admin terdiri dari halaman *Dashboard* yang akan menampilkan aktivitas yang sedang berjalan di dalam *warehouse* tersebut. Dalam halaman ini juga Admin dapat langsung memantau pesanan dari *warehouse* lain dan juga dapat *generate* laporan yang berisi aktivitas dalam *warehouse* tersebut.

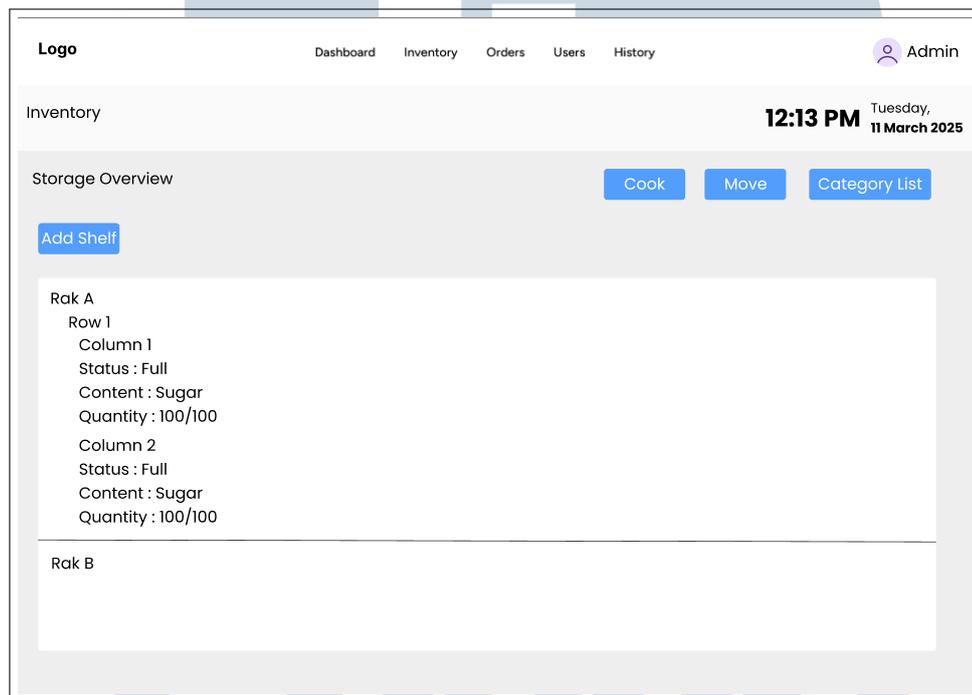
Selain itu, Admin dapat membuka halaman *Inventory* yang akan menampilkan fitur yang dibutuhkan dalam manajemen *warehouse*. Halaman Order untuk memantau pesanan dan membuat pesanan, halaman *Users* untuk melihat daftar akun karyawan, dan halaman *History* untuk melihat semua pergerakan barang dalam *warehouse*. Desain untuk halaman *Admin Dashboard* dapat dilihat pada Gambar 3.10.



Gambar 3.10. Desain halaman *admin dashboard*

B Desain Halaman Inventory

Gambar 3.11 menampilkan halaman *Inventory* yang menampilkan tabel dengan nama rak beserta isi dari rak tersebut. Halaman ini memiliki fungsi dalam manajemen *inventory* gudang, mulai dari penambahan rak baru, proses pemindahan produk, manajemen kategori produk, dan proses memasak atau mengolah produk bahan baku menjadi produk bahan setengah jadi. Untuk proses memasak hanya dapat dilakukan oleh tipe gudang bahan setengah jadi.

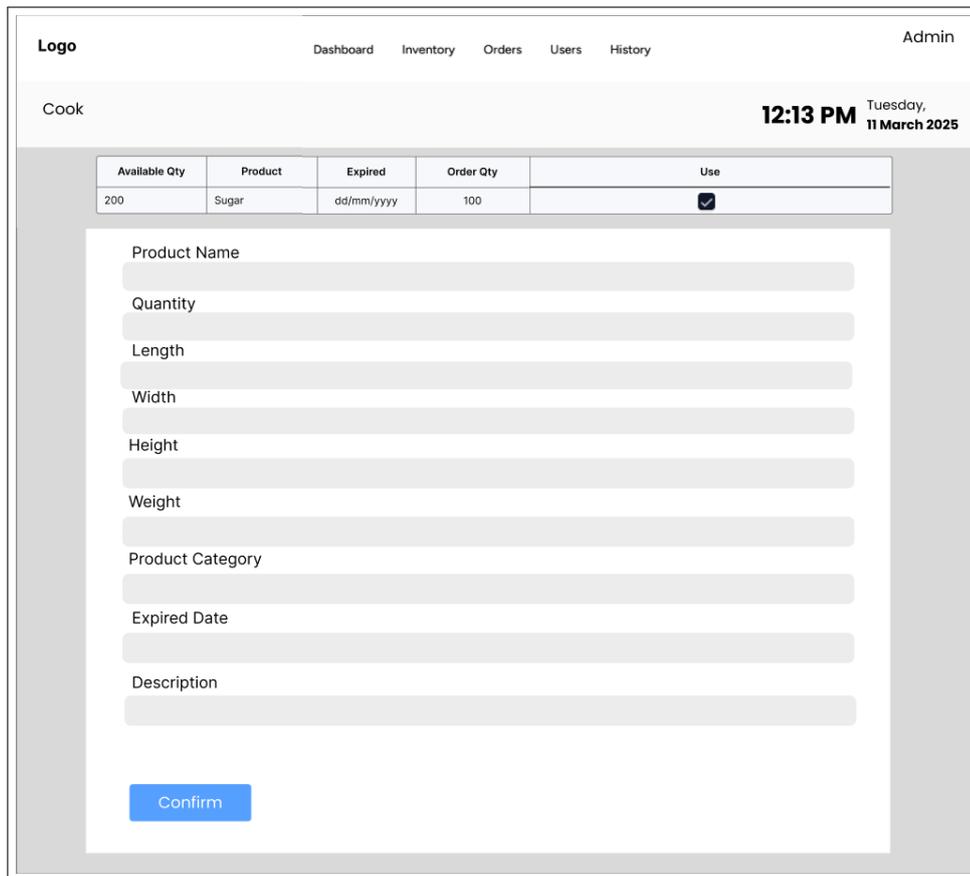


Gambar 3.11. Desain halaman Inventory *warehouse* bahan setengah jadi

C Desain Halaman Cook

Gambar 3.12 merupakan desain halaman *Cook* yang menampilkan tabel berisi daftar produk yang bisa digunakan untuk dimasak. Dalam tabel ini terdapat informasi seperti nama produk, kuantitas produk, tanggal kedaluwarsa, tombol untuk menggunakan produk untuk dimasak, dan kuantitas produk yang akan digunakan.

Pada bagian bawah tabel terdapat sebuah *form* untuk membuat informasi produk hasil masak. *Form* tersebut terdapat informasi dari nama produk, kuantitas produk yang akan dihasilkan, panjang produk, lebar produk, tinggi produk, berat produk, kategori produk, tanggal kedaluwarsa, dan deskripsi produk.



Gambar 3.12. Desain halaman *cook*

D Desain Halaman Move

Gambar 3.13 merupakan desain untuk halaman *Move* yang menampilkan sebuah *form*. Admin dapat memilih rak asal sebuah produk dan rak tujuan produk, kemudian informasi produk dan kuantitas akan muncul setelah Admin menentukan rak asal dan tujuan, dan Admin diharuskan untuk mengisi alasan barang tersebut akan dialokasikan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

The image shows a web application interface for a 'New Order' form. The form is contained within a modal window. At the top of the page, there is a navigation bar with a 'Logo' on the left and menu items 'Dashboard', 'Inventory', 'Orders', 'Users', and 'History' in the center. On the right side of the navigation bar, there is a user profile icon labeled 'Admin'. Below the navigation bar, the page title 'New Order' is on the left, and the date and time '12:13 PM Tuesday, 11 March 2025' are on the right. The main form area has the following fields: 'From' with the value 'Rack A - Row 1 - Col 1', 'To' with the value 'Rack B - Row 1 - Col 1', 'Product Information' with the value 'Jelly', 'Quantity' with the value '10', and 'Reason' with the value 'Expired Soon'. A blue 'Confirm' button is positioned at the bottom left of the form.

Gambar 3.13. Desain halaman *Move*

E Desain Halaman Add Shelf

Gambar 3.14 menampilkan desain *form* untuk menambah rak baru ke dalam sistem. Informasi yang dibutuhkan untuk sebuah rak dibuat adalah nama rak, jumlah baris dalam rak, jumlah kolom per baris, kapasitas maksimal per kolom, panjang kolom, tinggi kolom, dan kategori rak. Informasi ini penting untuk memastikan ukuran rak tidak terlalu besar atau terlalu kecil.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

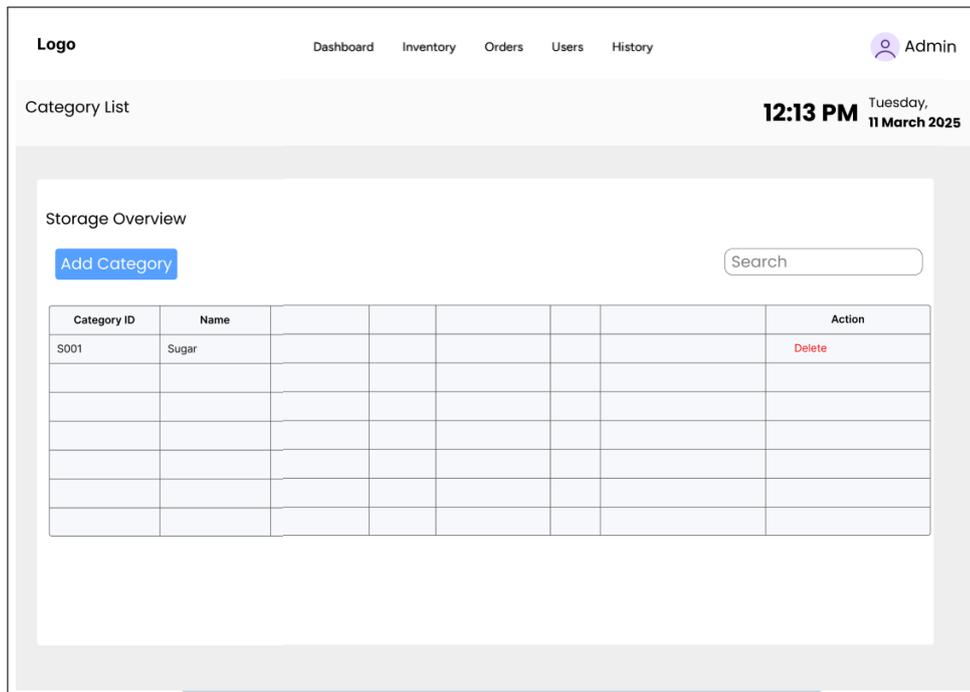
The image shows a web application interface for adding a shelf. At the top, there is a navigation bar with a logo on the left and links for Dashboard, Inventory, Orders, Users, and History on the right. A user profile icon labeled 'Admin' is also present. Below the navigation bar, the page title is 'Add Shelf'. On the right side of the page, the current time and date are displayed: '12:13 PM Tuesday, 11 March 2025'. The main content area contains a form with the following fields: 'Shelf Name', 'Shelf Row', 'Shelf Column per Row', 'Max Capacity per Column', 'Column Length', 'Column Height', and 'Shelf Category'. Each field has a corresponding input box. At the bottom of the form, there is a blue 'Confirm' button.

Gambar 3.14. Desain halaman *Add Shelf*

F Desain Halaman *Category List*

Gambar 3.15 menampilkan desain untuk halaman *Category List*. Halaman ini menampilkan sebuah tabel yang berisikan daftar kategori yang ada pada *warehouse*. Pada sisi atas kiri terdapat sebuah tombol untuk menambahkan kategori baru ke dalam daftar kategori dan pada sisi atas kanan terdapat *search bar* untuk mencari kategori.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

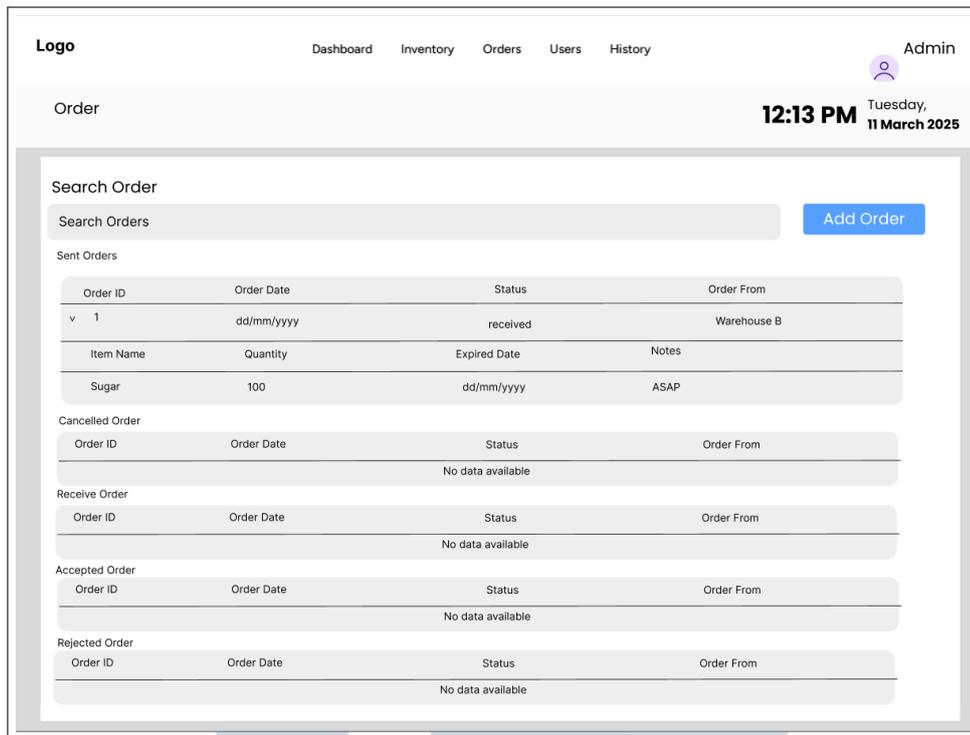


Gambar 3.15. Desain halaman *Category List*

G Desain Halaman Order

Gambar 3.16 menampilkan desain halaman *Order* yang terdiri dari beberapa tabel, antara lain tabel *sent order* yang menampilkan daftar barang yang telah dipesan, tabel *cancelled order* untuk barang yang telah dibatalkan, tabel *receive order* untuk pesanan yang masuk, tabel *accepted order* untuk pesanan yang diterima dan akan segera dikirim, serta tabel *rejected order* untuk pesanan yang ditolak. Pada bagian atas tabel *sent order*, terdapat *search bar* yang memudahkan admin dalam mencari data pesanan tertentu. Khusus untuk *warehouse* bahan baku, halaman ini tidak menampilkan tabel untuk membatalkan atau mengirimkan *order*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

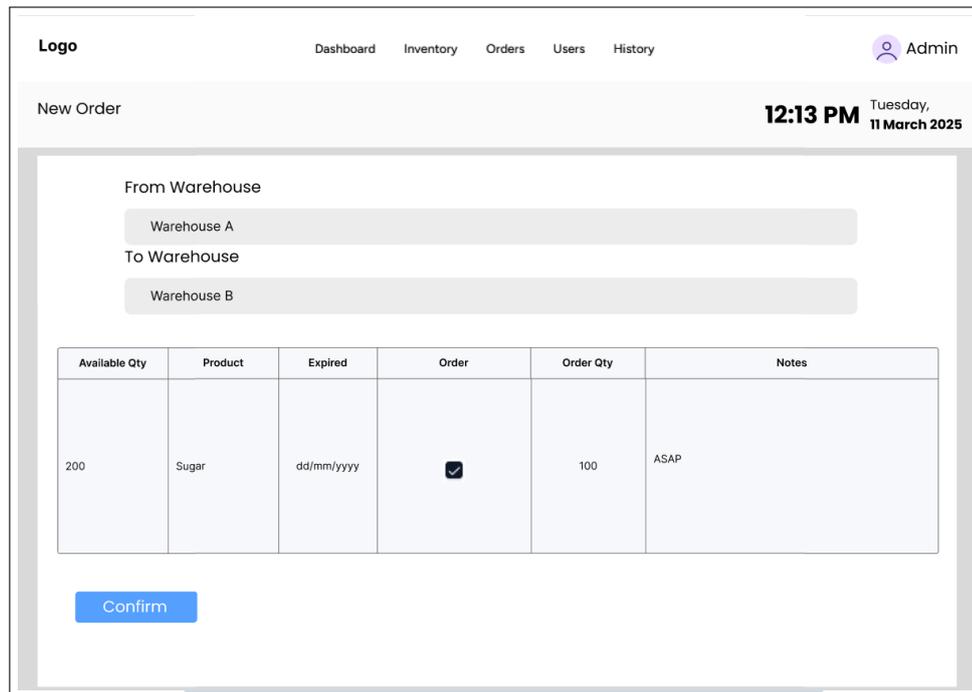


Gambar 3.16. Desain halaman Order

H Desain Halaman Add Order

Gambar 3.17 menampilkan desain halaman Add Order. Pada halaman ini Admin diharuskan untuk memilih *warehouse* terlebih dahulu, setelah *warehouse* dipilih akan ditampilkan tabel dengan daftar barang yang tersedia dalam *warehouse* tersebut. Setelah admin memilih barang yang ingin dipesan, Admin harus mengisi kuantitas yang ingin dipesan dan dapat memberikan catatan(opsional).

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



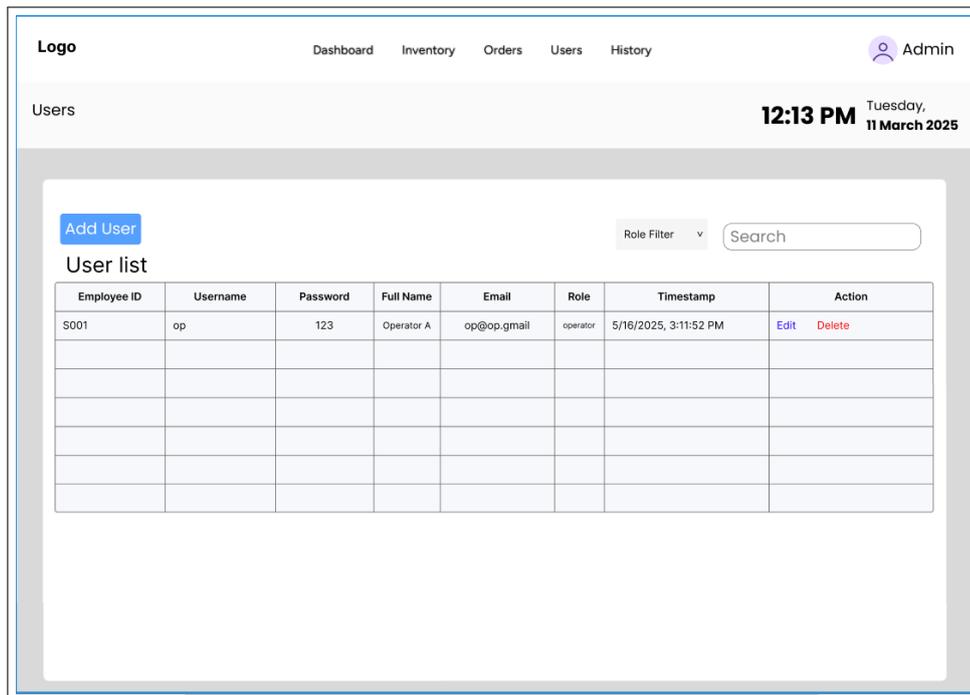
Gambar 3.17. Desain halaman *Add Order*

I Desain Halaman Users

Gambar 3.18 menampilkan desain halaman *add user* dengan menampilkan tabel yang berisi daftar akun karyawan yang bekerja dalam *warehouse* tersebut dan terdapat tombol *add user* pada isi atas kiri, filter berdasarkan *role*, dan *search bar* untuk mempermudah proses pencarian.

Tabel *user* ini memberikan informasi akun berupa *id* karyawan, *username*, nama lengkap, *password*, *e-mail*, *role*, waktu kapan akun dibuat, dan aksi berupa *editing* dan menghapus akun dari sistem.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

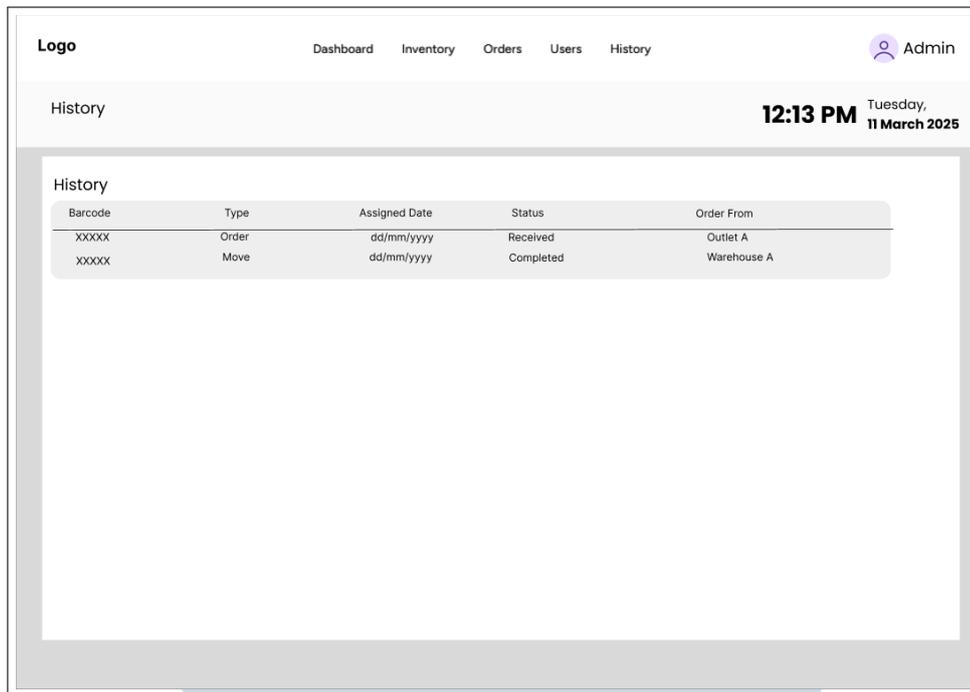


Gambar 3.18. Desain halaman daftar *Users*

J Desain Halaman History

Gambar 3.19 menampilkan desain untuk halaman *history*. Halaman *history* menampilkan sebuah tabel yang berisi informasi transaksi barang yang sudah terjadi. Dalam tabel tersebut akan ditampilkan informasi seperti *barcode id*, tipe transaksi (*Order*, *return*, atau *move*), tanggal transaksi, status transaksi (*received*, *completed*, dan *rejected*), dan tempat transaksi terjadi atau asal transaksi dibuat.

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



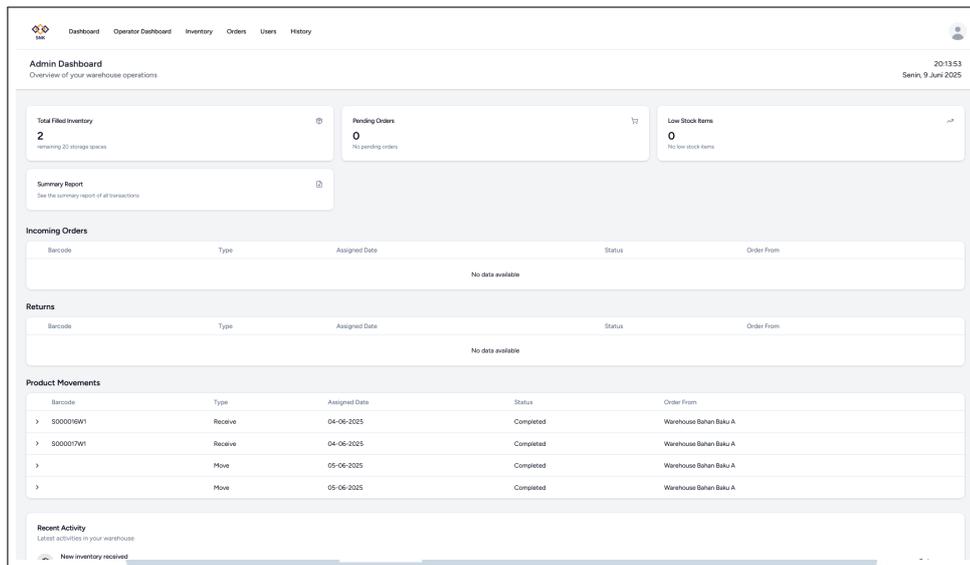
Gambar 3.19. Desain halaman *history*

3.5.3 Implementasi Figma

Setelah selesai mendesain tampilan di Figma, dilakukan pengodingan untuk mengimplementasikan desain yang sudah dibuat.

A Implementasi Desain Halaman Admin Dashboard

Gambar 3.20 menampilkan halaman admin *dashboard* setelah desain dari Figma diimplementasikan. Terdapat penambahan halaman operator *dashboard* pada bagian *navigation bar* dan terdapat tombol *summary* yang akan *generate form pdf* berisi laporan transaksi. Pada halaman ini admin dapat melihat *recent activity* yang terjadi di gudang, melihat tabel *incoming order*, tabel *returns* yang menunjukkan permintaan produk dikembalikan, dan *product movement* untuk melihat pergerakan produk dalam gudang seperti *move product* atau *receive product*.



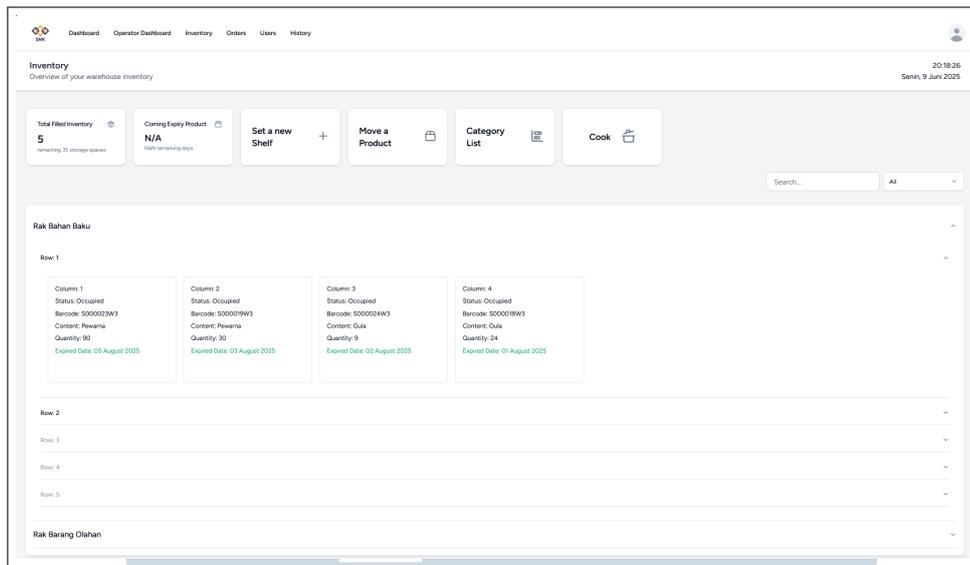
Gambar 3.20. Implementasi halaman admin *dashboard*

B Implementasi Desain Halaman Inventory

Gambar 3.21 menampilkan halaman *inventory* yang menampilkan tabel dengan nama rak beserta isi dari rak tersebut. Halaman ini memiliki peran penting dalam manajemen gudang, mulai dari penambahan rak baru, pemindahan produk ke rak lain, manajemen kategori, dan proses memasak atau mengolah produk bahan baku menjadi produk bahan setengah jadi.

Halaman *inventory* memiliki perbedaan untuk setiap tipe gudang. *Editing* terhadap informasi rak hanya dapat dilakukan pada tipe gudang bahan baku, sedangkan tipe gudang bahan setengah jadi bisa memproses produk menjadi bahan setengah jadi, sedangkan tipe gudang *outlet* tidak memiliki kedua fitur tersebut.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.21. Implementasi desain halaman *inventory warehouse* bahan setengah jadi

C Implementasi Desain Halaman Cook

Gambar 3.22 merupakan hasil implementasi desain halaman *cook* yang menampilkan tabel berisi daftar produk yang bisa digunakan untuk dimasak. Dalam tabel ini terdapat informasi seperti nama produk, kuantitas produk, tanggal kedaluwarsa, tombol untuk menggunakan produk untuk dimasak, dan kuantitas produk yang akan digunakan.

Pada bagian bawah tabel terdapat sebuah *form* untuk membuat informasi produk hasil masak. *Form* tersebut terdapat informasi dari nama produk, kuantitas produk yang akan dihasilkan, panjang produk, lebar produk, tinggi produk, berat produk, kategori produk, tanggal kedaluwarsa, dan deskripsi produk.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Available Quantity	Product	Expired Date	Use	Quantity
9	Gula	02/08/2025	<input checked="" type="checkbox"/>	9
24	Gula	01/08/2025	<input type="checkbox"/>	
30	Peanut	03/08/2025	<input checked="" type="checkbox"/>	23
100	Peanut	05/08/2025	<input type="checkbox"/>	
15	Gula	02/08/2025	<input type="checkbox"/>	
90	Peanut	05/08/2025	<input type="checkbox"/>	

Product Name
A

Quantity
1

Length
1

Width
1

Height
1

Weight (kg)
1

Product Category
Select A Category

Expired Date
June 9th, 2025

Description
Lorem Ipsum

Confirm

Gambar 3.22. Implementasi halaman *cook*

D Implementasi Desain Halaman Move

Gambar 3.23 menampilkan halaman *move* yang berisi sebuah *form* yang memiliki *drop down* untuk memilih rak asal dan rak tujuan, dan sebuah *text box* untuk menulis alasan produk dipindahkan. Setelah rak asal dipilih, *form* akan menampilkan informasi produk dan kuantitas produk yang ada dalam rak terpilih.

From
Rak Masuk Barang Row: 1 Col: 1

To
Rak Masuk Barang Row: 1 Col: 4

Product Information
Gula

Quantity
100

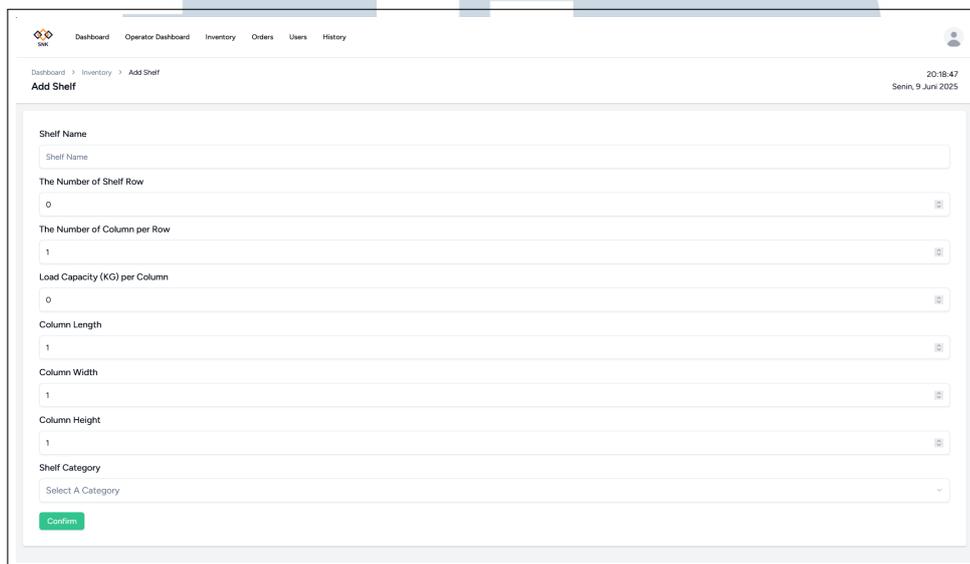
Reason

Confirm

Gambar 3.23. Implementasi halaman *move*

E Implementasi Desain Halaman Add Shelf

Gambar 3.24 menampilkan halaman *add shelf* dengan bentuk *form* yang harus diisi dengan informasi nama rak, jumlah baris dalam rak, jumlah kolom per baris, berat maksimum per kolom, panjang kolom, lebar kolom, tinggi kolom dalam satuan sentimeter, dan *drop down* pilihan kategori untuk rak. Informasi yang dituliskan dalam *form* berperan penting dalam efisiensi tempat pada gudang agar rak yang digunakan tidak terlalu besar atau terlalu kecil.

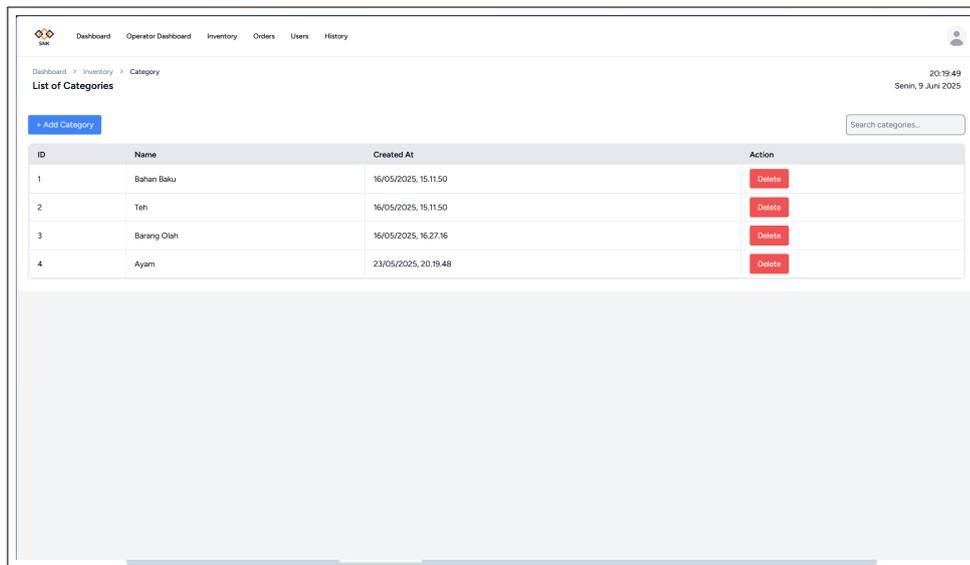


Gambar 3.24. Implementasi halaman *add shelf*

F Implementasi Desain Halaman Category List

Gambar 3.25 menunjukkan halaman *category list* yang berperan penting dalam manajemen kategori produk dalam gudang. Halaman ini menampilkan sebuah tabel daftar kategori yang memberikan informasi *category id*, nama kategori, kapan kategori dibuat, dan tombol aksi untuk menambahkan dan menghapus kategori dari daftar tabel.

Proses penambahan kategori ke dalam sistem hanya perlu untuk menuliskan kategori baru dalam sebuah *form* saat admin mengklik tombol *add category*, dan proses menghapus kategori dari daftar kategori hanya dapat dilakukan ketika nama kategori tersebut sudah tidak digunakan lagi oleh seluruh tipe gudang.



Gambar 3.25. Implementasi halaman *category list*

G Implementasi Desain Halaman Order

Gambar 3.26 menampilkan desain halaman *Order* yang terdiri dari beberapa tabel, antara lain tabel *sent order* yang menampilkan daftar barang yang telah dipesan, tabel *cancelled order* untuk barang yang telah dibatalkan, tabel *receive order* untuk pesanan yang masuk, tabel *accepted order* untuk pesanan yang diterima dan akan segera dikirim, serta tabel *rejected order* untuk pesanan yang ditolak. Pada bagian atas tabel *sent order*, terdapat *search bar* yang memudahkan admin dalam mencari data pesanan tertentu. Khusus untuk *warehouse* bahan baku, halaman ini tidak menampilkan tabel untuk membatalkan atau mengirimkan *order*.

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

Order ID	Order Date	Status	Order From
104	04-06-2025	pending	Wahana Sategyn Jald A
103	04-06-2025	cancelled	Wahana Sategyn Jald A
99	03-06-2025	received	Wahana Sategyn Jald A
98	03-06-2025	received	Wahana Sategyn Jald A
97	03-06-2025	in delivery	Wahana Sategyn Jald A
94	03-06-2025	in delivery	Wahana Sategyn Jald A
91	03-06-2025	rejected	Wahana Sategyn Jald A
91	07-06-2025	in delivery	Wahana Sategyn Jald A
87	07-06-2025	rejected	Wahana Sategyn Jald A
59	07-06-2025	received	Wahana Sategyn Jald A
58	07-06-2025	received	Wahana Sategyn Jald A
30	04-06-2025	received	Wahana Sategyn Jald A

Order ID	Order Date	Status	Order From
29	04-06-2025	cancelled	Wahana Sategyn Jald B

Order ID	Order Date	Status	Order From
No data available			

Order ID	Order Date	Status	Order From
No data available			

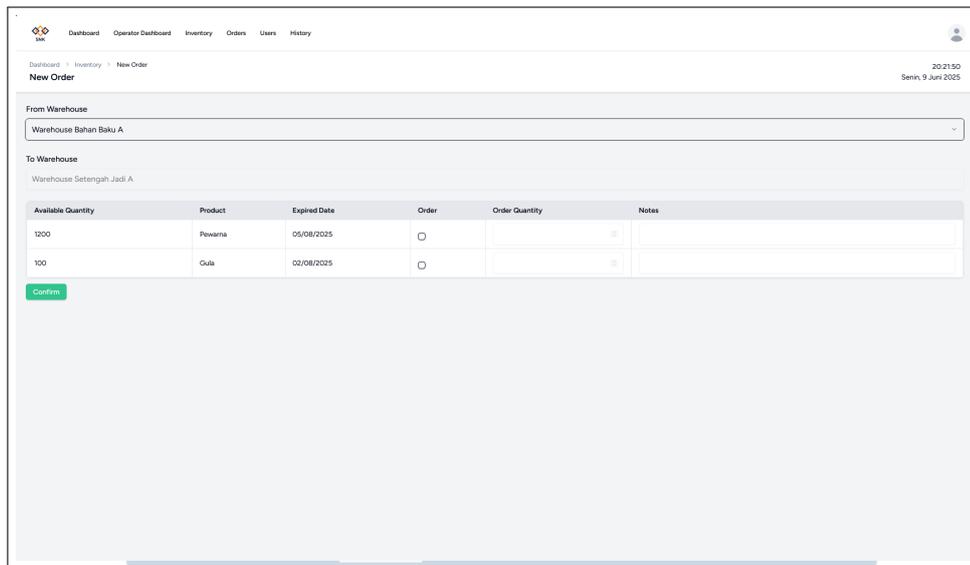
Order ID	Order Date	Status	Order From
100	03-06-2025	rejected	Wahana Sategyn Jald A

Gambar 3.26. Implementasi halaman order

H Implementasi Desain Halaman Add Order

Gambar 3.27 menunjukkan implementasi desain halaman *add order*. Halaman ini akan menampilkan tabel yang berisi informasi produk yang tersedia dalam *inventory* gudang. Admin harus terlebih dahulu memilih asal gudang untuk melihat produk yang tersedia. Setelah admin memilih gudang asal produk, selanjutnya admin akan meng-*checklist* produk yang mau dipesan, menuliskan kuantitas produk yang mau dikirimkan, dan menuliskan catatan untuk produk pesanan (opsional).

U 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

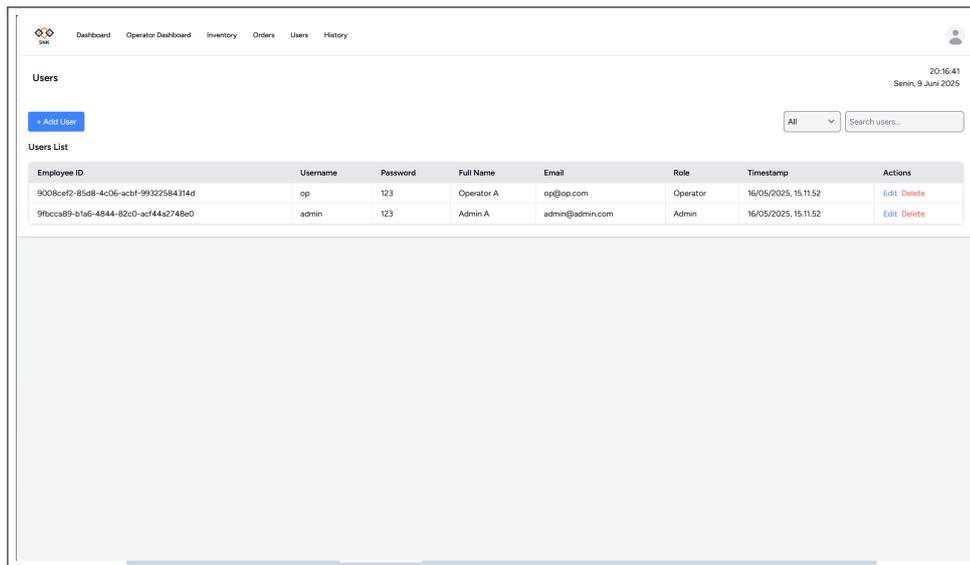


Gambar 3.27. Implementasi halaman *add order*

I Implementasi Desain Halaman User

Gambar 3.28 menampilkan halaman *user* yang memiliki tabel berisi informasi setiap karyawan, informasi terdiri dari *employee id*, *username*, *password*, nama lengkap, *e-mail*, *role*, waktu kapan akun karyawan dibuat, dan aksi yang bisa dilakukan terhadap akun karyawan. Pada bagian kanan atas dari tabel terdapat sebuah filter pengguna berdasarkan *role* dan ada *search bar* untuk mencari akun secara spesifik.

Terdapat tiga aksi yang dapat dilakukan oleh admin, pertama admin dapat menambah akun pengguna dan harus mengisi *form* ketika admin klik tombol *add user*. Aksi edit akan menampilkan *form* dan admin dapat mengedit informasi pengguna dari nama, *username*, *password*, dan *e-mail*. Aksi *delete* akan menampilkan *pop up alert* yang mengharuskan admin untuk menginput *password* sebelum menghapus akun. Hal ini sangat penting guna untuk konfirmasi lanjutan sebelum menghapus akun.



Gambar 3.28. Implementasi desain halaman *user*

J Implementasi Desain Halaman History

Gambar 3.29 menampilkan tabel yang berisi *history* transaksi dari *delivery*, *move*, *return*. Dalam tabel ini pengguna dapat melihat *delivery barcode* dan *return barcode*, tanggal produk dikirim atau dipindahkan, status transaksi barang seperti *"On Delivery"*, *"Completed"*, *"Received"*, dan *"Returning"*, dan dapat melihat asal tipe gudang yang melakukan transaksi. Selain dapat melihat transaksi, pengguna juga dapat melihat isi transaksi dari nama pengguna yang melakukan transaksi, nama produk, kuantitas produk, dan tanggal kedaluwarsa.

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

Barcode	Type	Assigned Date	Status	Order From
▼ D000104	Order	16-05-2025	On delivery	Warehouse Setengah Jadi A
User ID: 9fbcca89-b1a6-4844-82d0-ac144a2748e0 User Name: Admin A				
Item Name		Quantity	Expired Date	
Pewarna		100	04-08-2025	
Gula		25	01-08-2025	
▼ R000101	Return	09-06-2025	Completed	Warehouse Bahan Baku A
User ID: 9fbcca89-b1a6-4844-82d0-ac144a2748e0 User Name: Admin A				
Item Name		Quantity	Expired Date	
Pewarna		100	04-08-2025	
>	Return	09-06-2025	Rejected	Warehouse Bahan Baku A
>	Order	16-05-2025	Received	Warehouse Setengah Jadi A
>	Order	16-05-2025	Received	Warehouse Setengah Jadi A
>	Return	07-06-2025	Completed	Warehouse Bahan Baku A
>	Return	07-06-2025	Completed	Warehouse Bahan Baku A
>	Order	16-05-2025	Received	Warehouse Setengah Jadi A
>	Order	16-05-2025	Received	Warehouse Setengah Jadi A
>	Return	09-06-2025	Returning	Warehouse Bahan Baku A

Gambar 3.29. Implementasi halaman *history*

3.6 Kendala dan Solusi yang Ditemukan

Selama pelaksanaan kerja magang, mahasiswa menghadapi beberapa kendala teknis dan non-teknis. Salah satu kendala utama adalah kurangnya pemahaman awal terhadap alur sistem *Warehouse Management System (WMS)*, khususnya dalam menyesuaikan desain antarmuka dengan kebutuhan sistem nyata yang kompleks. Selain itu, desain antarmuka yang telah dibuat di Figma tidak sepenuhnya dapat diimplementasikan ke dalam React karena adanya keterbatasan dalam penanganan data dinamis dan struktur tampilan yang terlalu ideal.

Untuk mengatasi kendala tersebut, mahasiswa secara mandiri mempelajari alur sistem yang dikembangkan, dan berdiskusi langsung dengan supervisor. Selain itu, mahasiswa melakukan penyesuaian pada tampilan agar tetap fungsional, meskipun tidak identik dengan desain awal. Proses revisi desain dilakukan dengan tetap mempertahankan prinsip kemudahan penggunaan (*usability*). Dengan pendekatan ini, mahasiswa mampu menyelesaikan seluruh tugas pengembangan antarmuka secara fungsional dan terintegrasi dengan sistem back-end yang ada.