

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Kegiatan kerja Profesi dilaksanakan selama 3 bulan di Meants Club Clothing brand, saya ditempatkan di divisi IT yang mencakup maintenance *hardware* dan juga *software*, dan juga sebagai programmer untuk proyek aplikasi *inventory* barang. Dalam laporan ini saya berfokus pada proyek pembuatan aplikasi yang dimana main role saya adalah sebagai sistem analis dan juga programmer. Dan dalam proyek ini saya mengerjakan sendiri diawasi langsung oleh owner.

Aplikasi yang akan dibuat ini akan digunakan oleh sales, manager sales dan juga owner untuk memantau stok barang yang tersedia melalui aplikasi web. Dan juga menghasilkan *print out* laporan stok barang.

3.2 ● Pelaksanaan Kerja Profesi

Kerja profesi dimulai sejak tanggal 01 Maret 2023 sampai dengan 31 Mei 2023 di Meants Club Clothing Brand, keahlian yang harus dimiliki disini adalah kemampuan merancang web aplikasi secara jelas dan rinci kebutuhan dan goals aplikasi tersebut.

Untuk mencapai prestasi kerja yang tinggi mahasiswa harus aktif dalam update progress setiap menyelesaikan sebuah progress dalam jangka waktu tertentu dan aktif dalam berkomunikasi dengan owner

Berikut merupakan table rincian kegiatan yang dilakukan selama kegiatan kerja profesi di Meants Club Clothing Brand :

Tabel 3.1 Jadwal Pelaksanaan Kerja Profesi

No	Deskripsi Kegiatan	Maret	April	Mei
1	Analisis Kebutuhan Website	✓		
2	Proses Perancangan UI	✓	✓	✓
3	Proses Perancangan Website	✓	✓	✓
4	Menyusun Laporan Kerja Profesi			✓

3.2.1. Analisis Kebutuhan Website

Analisis kebutuhan sistem pada fase pertama dan yang paling penting adalah membuat dasar untuk Langkah pengembangan sistem lebih lanjut. Analisis kebutuhan mencakup analisis kebutuhan fungsional dan non fungsional. Analisis kebutuhan fungsional bertujuan untuk mendapatkan informasi kebutuhan proses yang dibutuhkan sistem inventory barang ini.

Berikut ini merupakan tabel analisis kebutuhan sistem *inventory* :

Tabel 3.2 Analisis Kebutuhan Sistem

No	Daftar Kebutuhan Aplikasi Inventory Berbasis Web
1	Website memiliki url tersembunyi yang harus diketik manual untuk login admin
2	Website dapat insert, update & delete stock
3	Web site dapat menampilkan laporan bang tersedia dan mencetak menjadi pdf

Dalam daftar didalam Tabel 1.2 diatas akan dirancang sebuah Aplikasi yang bisa memenuhi kebutuhan Bisnis.

1. Pada bagian awal akan ada halaman login untuk admin
2. Pada menu admin panel akan muncul table stok barang dan tombol tambah, update dan hapus

3.2.2. Software Yang Digunakan

Berikut adalah software yang digunakan untuk merancang Aplikasi web inventory milik Meants Club Clothing Brand.

1. Figma



Gambar 3.1 Logo Figma

Untuk membuat desain UI halaman admin dan landing page untuk catalog akan digunakan figma untuk membuat desain UI rancangannya

2. VSCODE



Gambar 3.2 Logo VSCODE

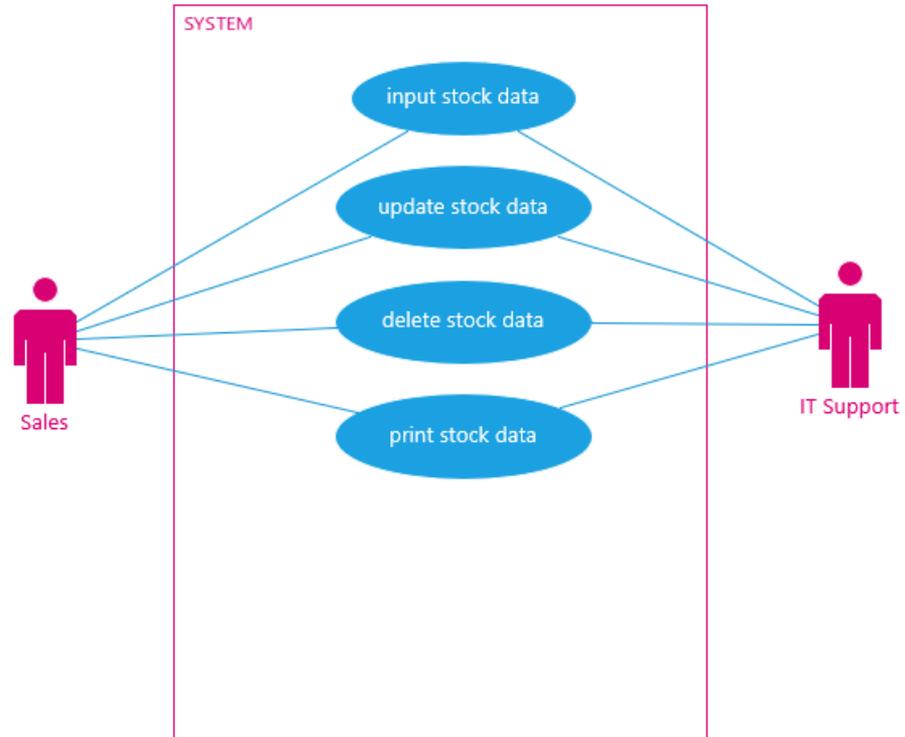
Visual Studio Code (VSCode) adalah editor teks *open source* yang dikembangkan oleh Microsoft. Ini adalah salah satu *editor* kode yang paling populer digunakan oleh pengembang perangkat lunak dan *web developer*. Berikut adalah beberapa poin penting tentang VSCode:

- a. Antarmuka Pengguna yang Responsif: VSCode memiliki antarmuka pengguna yang sederhana dan responsif. Fitur-fitur utama ditampilkan dengan jelas, dan pengguna dapat mengaksesnya melalui bilah menu, panel samping, atau melalui pintasan keyboard. Antarmuka pengguna yang intuitif dan mudah digunakan membuat pengembangan perangkat lunak menjadi lebih efisien.
- b. Antarmuka Pengguna yang Responsif: VSCode memiliki antarmuka pengguna yang sederhana dan responsif. Fitur-fitur utama ditampilkan dengan jelas, dan pengguna dapat mengaksesnya melalui bilah menu, panel samping, atau melalui pintasan keyboard. Antarmuka pengguna yang intuitif dan mudah digunakan membuat pengembangan perangkat lunak menjadi lebih efisien.
- c. Antarmuka Pengguna yang Responsif: VSCode memiliki antarmuka pengguna yang sederhana dan responsif. Fitur-fitur utama ditampilkan dengan jelas, dan pengguna dapat mengaksesnya melalui bilah menu, panel samping, atau melalui pintasan keyboard. Antarmuka pengguna yang intuitif dan mudah digunakan membuat pengembangan perangkat lunak menjadi lebih efisien.

- d. Antarmuka Pengguna yang Responsif: VSCode memiliki antarmuka pengguna yang sederhana dan responsif. Fitur-fitur utama ditampilkan dengan jelas, dan pengguna dapat mengaksesnya melalui bilah menu, panel samping, atau melalui pintasan keyboard. Antarmuka pengguna yang intuitif dan mudah digunakan membuat pengembangan perangkat lunak menjadi lebih efisien.
- e. Antarmuka Pengguna yang Responsif: VSCode memiliki antarmuka pengguna yang sederhana dan responsif. Fitur-fitur utama ditampilkan dengan jelas, dan pengguna dapat mengaksesnya melalui bilah menu, panel samping, atau melalui pintasan keyboard. Antarmuka pengguna yang intuitif dan mudah digunakan membuat pengembangan perangkat lunak menjadi lebih efisien.
- f. Antarmuka Pengguna yang Responsif: VSCode memiliki antarmuka pengguna yang sederhana dan responsif. Fitur-fitur utama ditampilkan dengan jelas, dan pengguna dapat mengaksesnya melalui bilah menu, panel samping, atau melalui pintasan keyboard. Antarmuka pengguna yang intuitif dan mudah digunakan membuat pengembangan perangkat lunak menjadi lebih efisien.
- g. Antarmuka Pengguna yang Responsif: VSCode memiliki antarmuka pengguna yang sederhana dan responsif. Fitur-fitur utama ditampilkan dengan jelas, dan pengguna dapat mengaksesnya melalui bilah menu, panel samping, atau melalui pintasan keyboard. Antarmuka pengguna yang intuitif dan mudah digunakan membuat pengembangan perangkat lunak menjadi lebih efisien.

VSCode adalah editor teks yang sangat populer dan terus berkembang dengan komunitas pengguna yang aktif. Dengan dukungan lintas platform (Windows, macOS, dan Linux), VSCode menjadi pilihan yang populer di antara pengembang perangkat lunak dan web developer.

3.2.3 Use Case Diagram



Gambar 3.3 Use Case Diagram User

Tabel 3.3 Use Case scenario table system

Use Case Name	:	System
Use case description	:	Admin login kedalam sistem dan akses fitur system
Actor	:	Admin
Pre-condition	:	<ul style="list-style-type: none"> • System terhubung ke internet • System memiliki database
Post-Condition		<ul style="list-style-type: none"> • Admin dapat memanipulasi data dalam database melalui fitur didalam adminpanel

Gambar diagram di atas menjelaskan bagaimana admin dan sales sebagai aktor dapat berinteraksi dengan Aplikasi. Setelah memasuki halaman login, admin akan masuk ke halaman panel admin yang berisi data stok dan bisa dimanipulasi seperti insert, update dan delete dan juga dapat membuat report stok barang berbentuk pdf.

1. Insert Stock

Dalam bagian ini admin bisa menambahkan stok barang sesuai detail nama barang, type barang, size barang, jumlah stok.

2. Update Stock

Apabila stok barang sudah habis atau ada penambahan maka stok bisa edit di fitur ini.

3. Delete Stock

Apabila barang sudah tidak digunakan maka barang akan di delete dari list stok.

4. Print Stock Report

Halaman yang menampilkan view stok terbaru.

3.2.4 Use Case Scenario

Use case scenario adalah tabel yang digunakan untuk menggambarkan dan mendokumentasikan skenario penggunaan (*use case scenario*) dari suatu sistem atau aplikasi. Tabel ini menyajikan informasi terstruktur tentang skenario-skenario yang melibatkan interaksi antara aktor-aktor dan sistem.

Tabel 3.4 Use Case Scenario login admin

Use Case Name	Login	
Goal in context	Admin berhasil login	
Precondition	Menampilkan admin panel	
Successful End Condition	Berhasil menampilkan admin panel	
Failed End Condition	-	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	-	
Main Flow	Step	Action
	1	Sistem menampilkan halaman login
	2	Admin memasukan username dan password
	3	Sistem akan mengarahkan admin ke Admin Panel

Tabel 3.5 Use Case Scenario Input Stock

Use Case Name	<i>Input stock</i>	
Goal in context	Admin berhasil mengisi stok baru	
Precondition	Mengisi stock baru ke database	
Successful EndCondition	Berhasil mengisi stok baru ke database	
Failed End Condition	-	
Primary Actor	<i>Admin</i>	
Secondary Actor	-	
Trigger	-	
Main Flow	Step	Action
	1	Sistem menampilkan halaman login
	2	Admin Mengisi username dan password
	3	Sistem akan mengarahkan admin ke Menu Input Stock
	4	Admin mengisi form input stock
	5	Stock baru akan masuk ke database

Tabel 3.6 Use Case Scenario Update Stock

Use Case Name	<i>Update Stock</i>	
Goal in context	Admin berhasil update stock	
Precondition	Mengganti jumlah stok sisa barang di database	
Successful EndCondition	Berhasil mengubah isi stok baru ke database	
Failed End Condition	-	
Primary Actor	<i>Admin</i>	
Secondary Actor	-	
Trigger	-	
Main Flow	Step	Action
	1	Sistem menampilkan halaman Stok
	2	Admin memilih stock yang ingin di ubah dan memilih tombol update
	3	Sistem akan menampilkan jendela value stok
	4	Admin menekan tombol simpan
	5	Stock akan diperbarui

Tabel 3.7 Use Case Scenario Delete Stock

Use Case Name	<i>Delete Stock</i>	
Goal in context	Admin berhasil delete stock	
Precondition	Hapus data stok di database	
Successful EndCondition	Berhasil hapus data stok di database	
Failed End Condition	-	
Primary Actor	<i>Admin</i>	
Secondary Actor	-	
Trigger	-	
	Step	Action
Main Flow		
	1	Sistem menampilkan halaman Stok
	2	Admin memilih stock yang ingin di ubah dan memilih tombol delete

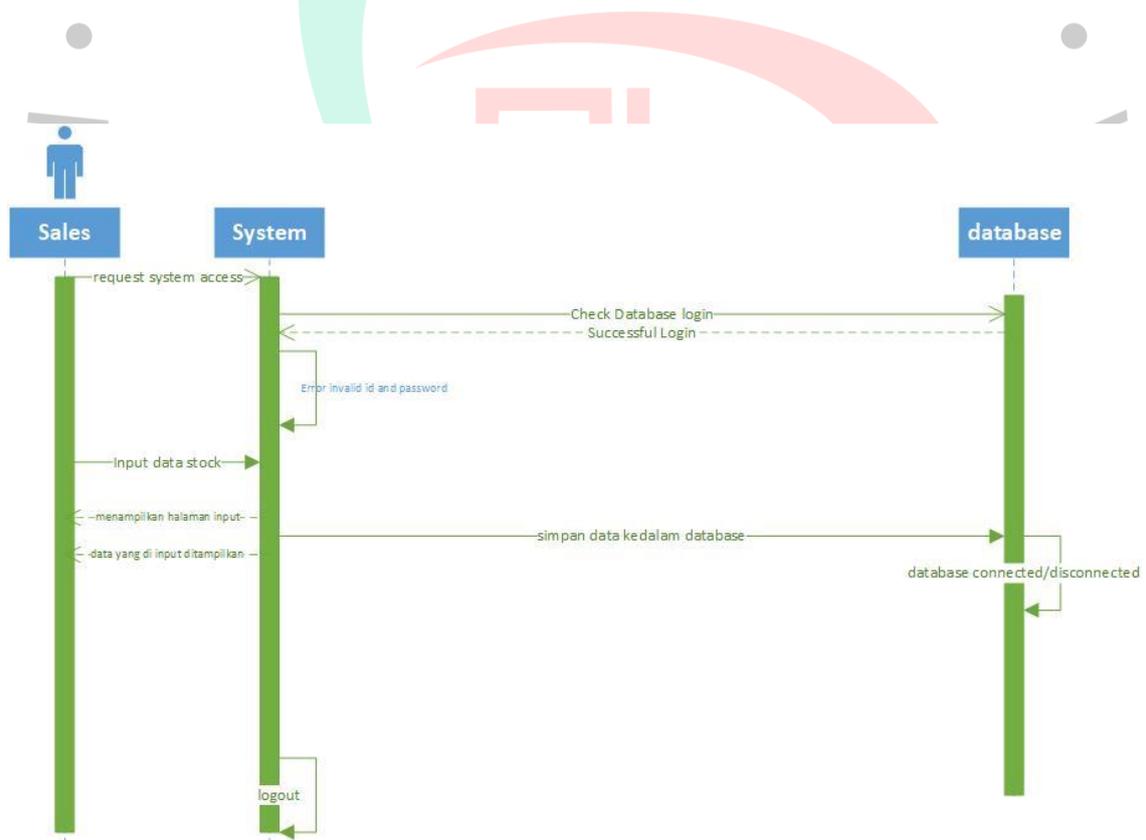
Tabel 3.8 Use Case Scenario Print Stock

Use Case Name	<i>Print Report</i>	
Goal in context	Admin berhasil print report	
Precondition	Mendownload pdf dari hasil stok yang ditampilkan	
Successful EndCondition	Berhasil menyimpan file pdf dari data stok	
Failed End Condition	-	
Primary Actor	<i>Admin</i>	
Secondary Actor	-	
Trigger	-	
	Step	Action
Main Flow		
	1	Sistem menampilkan halaman Stok
	2	Admin akan menekan tombol print report
	3	Sistem akan menampilkan halaman download
	4	File akan di download

3.2.5 Sequence Diagram

Diagram urutan (sequence diagram) adalah jenis diagram interaksi dalam rekayasa perangkat lunak dan analisis sistem yang menggambarkan interaksi antara objek-objek dalam suatu sistem secara urut. Diagram urutan memberikan representasi visual yang jelas tentang bagaimana objek-objek tersebut saling berinteraksi satu sama lain dalam suatu skenario.

Diagram urutan menggambarkan urutan pesan yang dikirim antara objek-objek dalam sistem, bersama dengan urutan peristiwa yang terjadi. Dalam diagram ini, objek-objek direpresentasikan sebagai kotak atau rektangle vertikal, dan pesan-pesan dikirim antara objek-objek dengan menggunakan panah. Panah menunjukkan arah pesan yang dikirim, dan nomor di sepanjang panah menunjukkan urutan pesan.



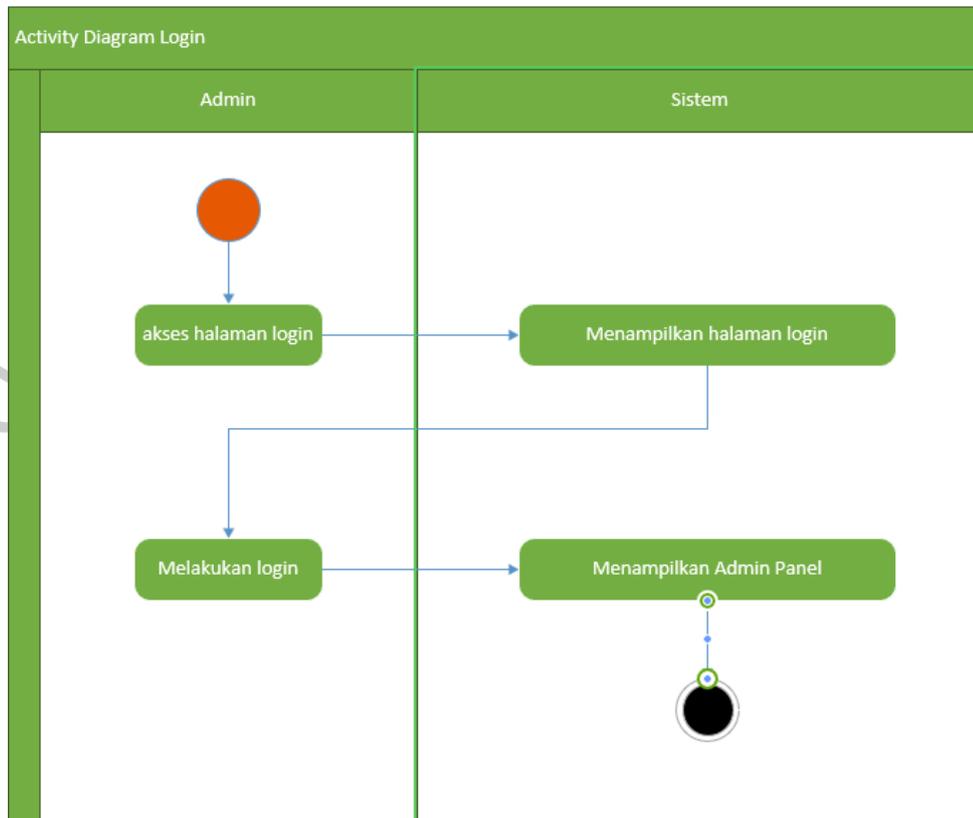
Gambar 3.4 Sequence Diagram

3.2.6 Activity Diagram

Activity diagram adalah diagram yang menggambarkan interaksi pengguna didalam sistem. Diagram ini juga digambarkan untuk menggambarkan fitur dalam

sistem yang sudah dirancang, seperti bagaimana setiap fungsi bekerja dari awal hingga akhir.

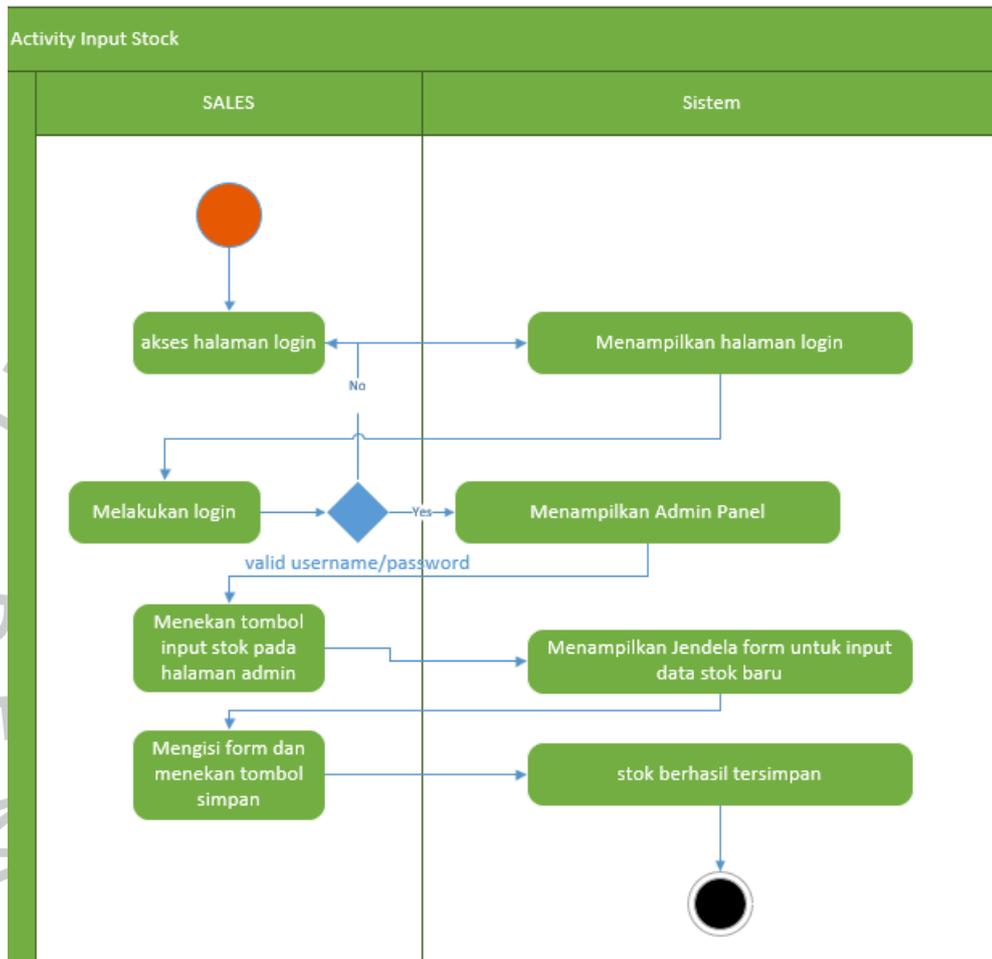
1. Activity Diagram Login



Gambar 3.5 Activity Diagram login

Activity Diagram login bermula pada fase admin melakukan proses login didalam halaman login dan setelah autentikasi berhasil akan menampilkan halaman login

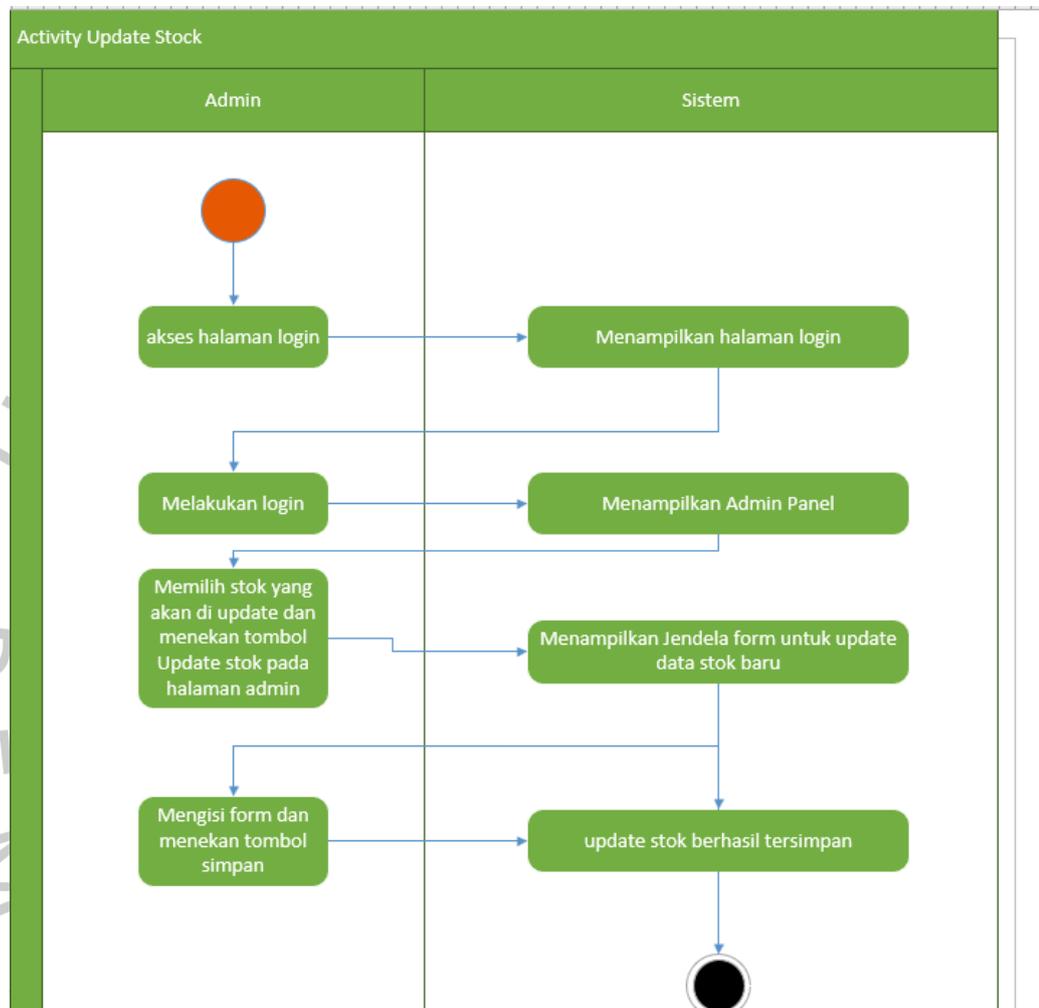
2. Activity Diagram Insert Stock



Gambar 3.6 Activity Diagram Input Stock

Activity Diagram input stok bermula pada fase admin melakukan proses login didalam halaman login dan setelah autentikasi berhasil akan menampilkan halaman login lalu admin menekan tombol insert stok untuk menampilkan jendela form yang harus diisi admin untuk menambah stok dan jika sudah terisi maka data stok akan bertambah dan data yang dimasukkan di form insert yang sudah ditekan oleh admin akan otomatis masuk kedalam database.

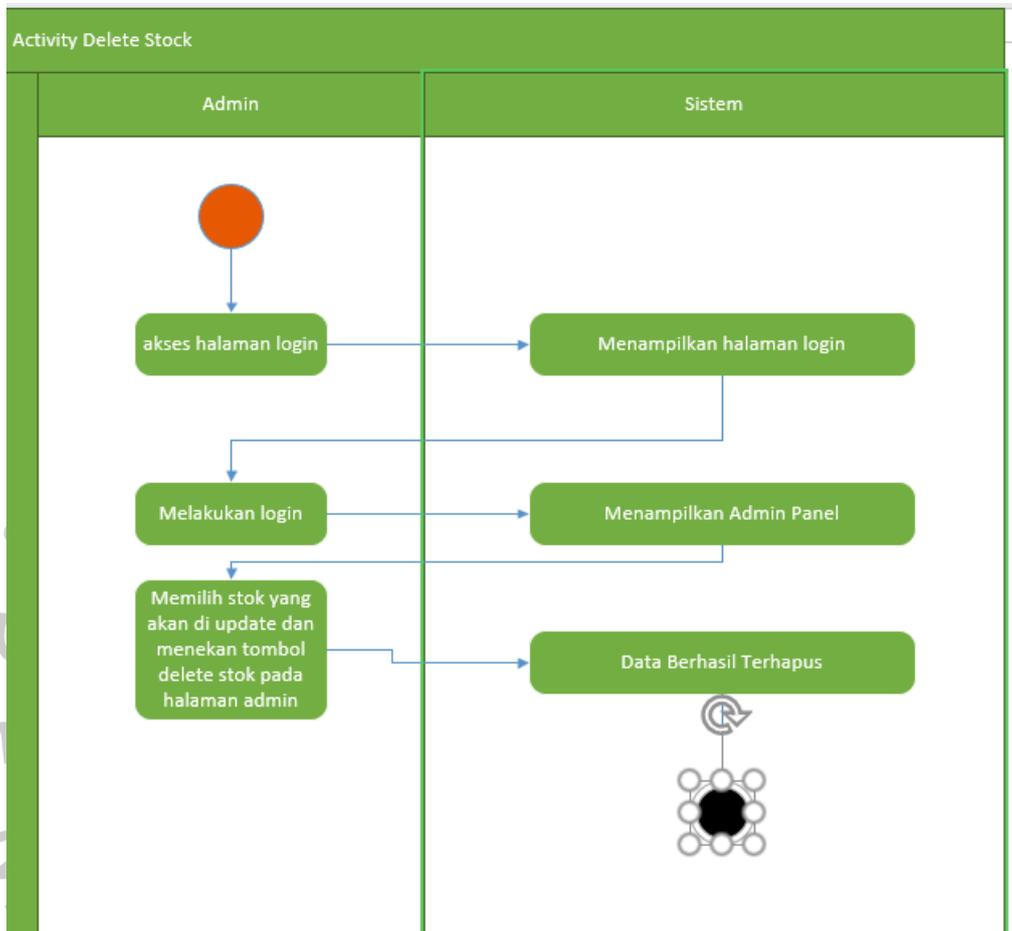
3. Activity Diagram Update Stok



Gambar 3.7 Activity Diagram Update Stock

Activity Diagram update stok bermula pada fase admin melakukan proses login didalam halaman login dan setelah autentikasi berhasil akan menampilkan halaman login lalu admin memilih data stok yang ingin di ubah dan menekan tombol update stok untuk menampilkan jendela form yang harus diisi admin untuk mengganti jumlah stok, jika sudah terisi maka data stok akan bertambah dan data yang dimasukan di form update yang sudah ditekan oleh admin akan otomatis mengganti data stok di dalam database.

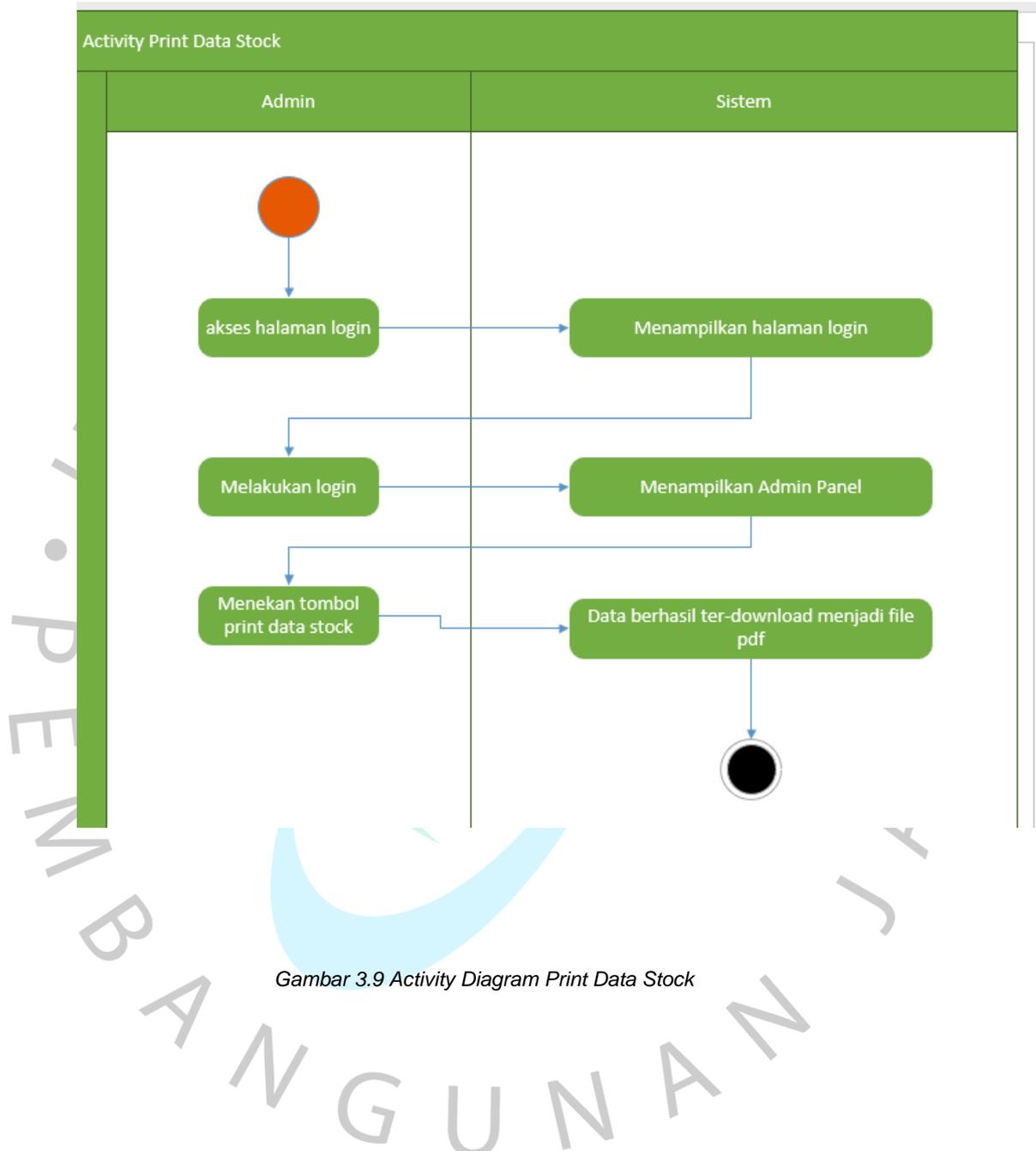
4. Activity Diagram Delete Stock



Gambar 3.8 Activity Diagram Delete Stock

Activity Diagram delete stok bermula pada fase admin melakukan proses login didalam halaman login dan setelah autentikasi berhasil akan menampilkan halaman login lalu admin memilih data stok yang ingin di hapus dan menekan tombol delete stok untuk menampilkan menghapus data stok, jika sudah terhapus maka data stok akan berkurang dan data yang dihapus akan otomatis terhapus di dalam database.

5. Activity Diagram Print Data Stock



Gambar 3.9 Activity Diagram Print Data Stock

Activity Diagram delete stok bermula pada fase admin melakukan proses login didalam halaman login dan setelah autentikasi berhasil akan menampilkan halaman login lalu admin menekan halaman print data stok dan akan ter-download menjadi file pdf.

3.2.7 Flowchart

Flowchart adalah representasi grafis dari alur atau urutan langkah-langkah dalam suatu proses atau program. Flowchart menggunakan simbol-simbol khusus yang menggambarkan setiap langkah atau tindakan, serta penghubung antar langkah-langkah tersebut. Flowchart digunakan untuk menggambarkan algoritma atau proses secara visual, sehingga memudahkan pemahaman dan analisis terhadap suatu proses.

Flowchart biasanya digunakan dalam pemrograman komputer, manajemen proyek, analisis bisnis, dan bidang lainnya yang melibatkan langkah-langkah atau proses yang kompleks. Dalam pemrograman, flowchart sering digunakan untuk merencanakan alur logika dari sebuah program sebelum diimplementasikan dalam bahasa pemrograman tertentu.

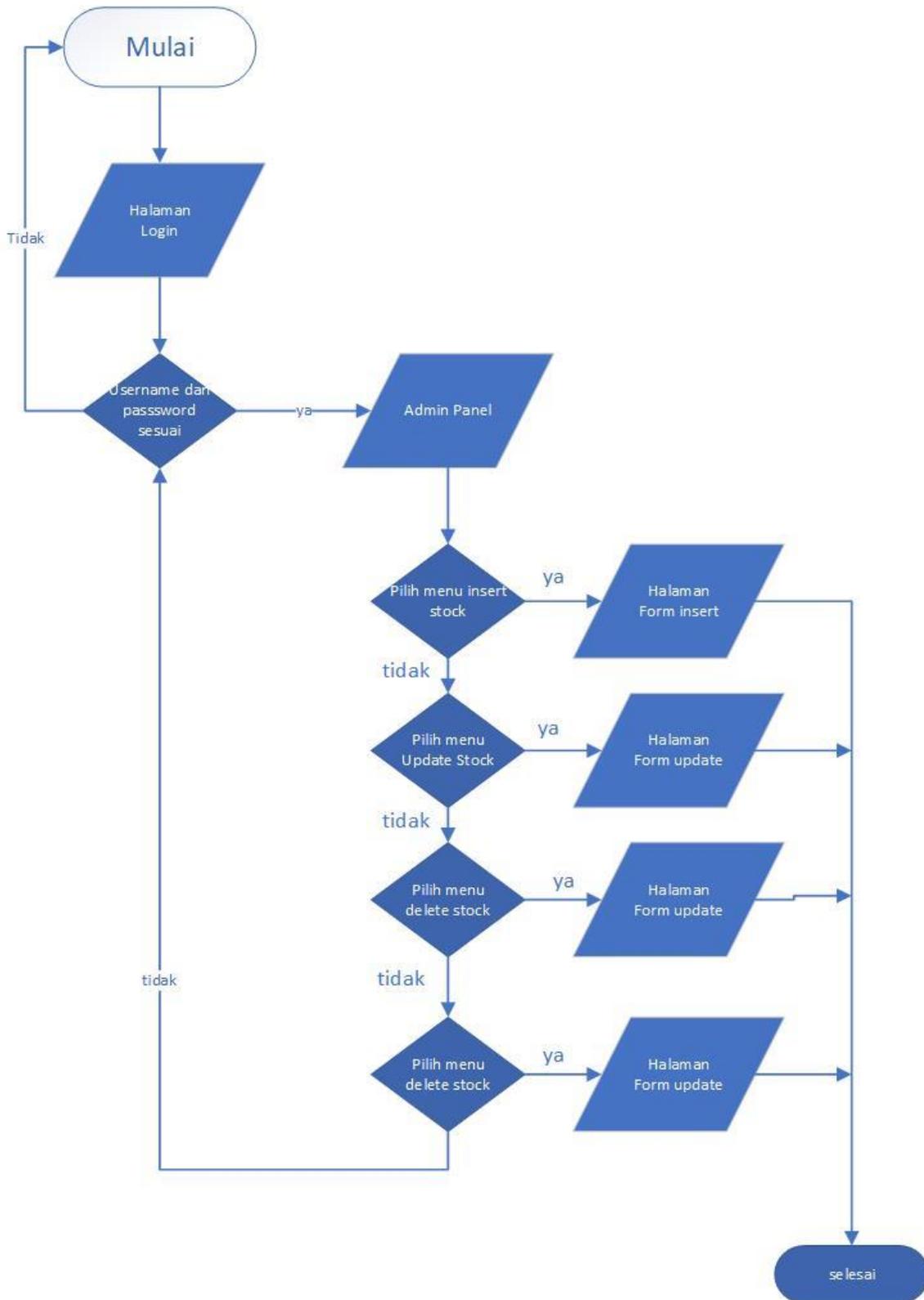
Berikut adalah beberapa simbol yang umum digunakan dalam flowchart:

Oval: Menunjukkan awal atau akhir dari suatu proses.

Kotak: Menunjukkan suatu tindakan atau langkah dalam proses.

Diamond: Menunjukkan suatu keputusan atau kondisi yang harus dievaluasi.

Panah: Menghubungkan langkah-langkah dalam urutan yang benar.



Gambar 3.10 Flowchart

Pada gambar diatas dimulai dari admin melakukan login kemudian setelah username dan password login diterima oleh sistem maka akan langsung diarahkan ke halaman admin untuk menampilkan menu stok yaitu

1. Data Stock

User diperlihatkan data stok yang tersedia berbentuk table yang menunjukkan field-field seperti nama barang, jenis barang, jumlah stock, tanggal masuk stock dan menu update dan delete stock juga akan muncul di bagian kanan baris stock.

2. Insert Stock

User menambahkan stok baru kedalam database

3. Update Stock

User dapat mengganti jumlah stok tersedia dan juga mengubah nama dan juga size pada kolom data stok

4. Delete Stock

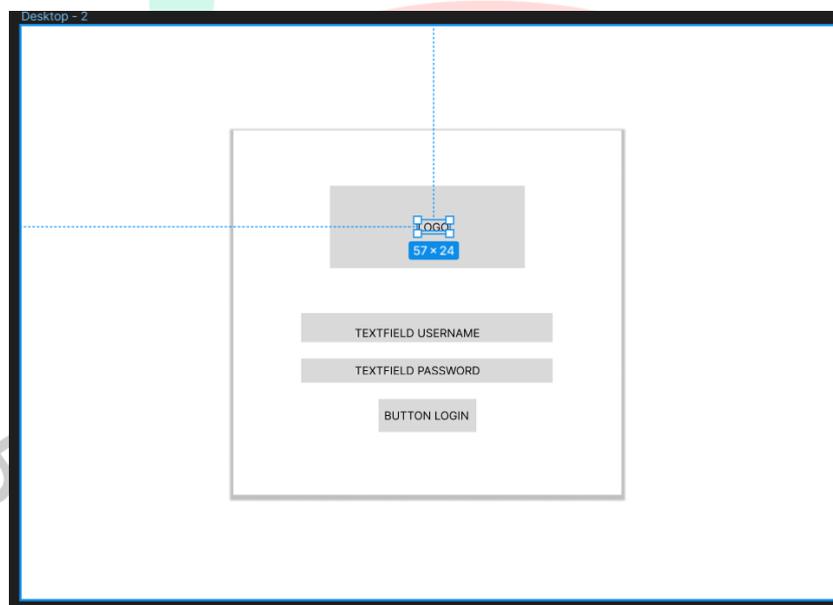
User dapat menghapus data stok yang sudah tidak akan diproduksi lagi

5. Print Data Stock

User dapat mengunduh data stok dalam bentuk pdf

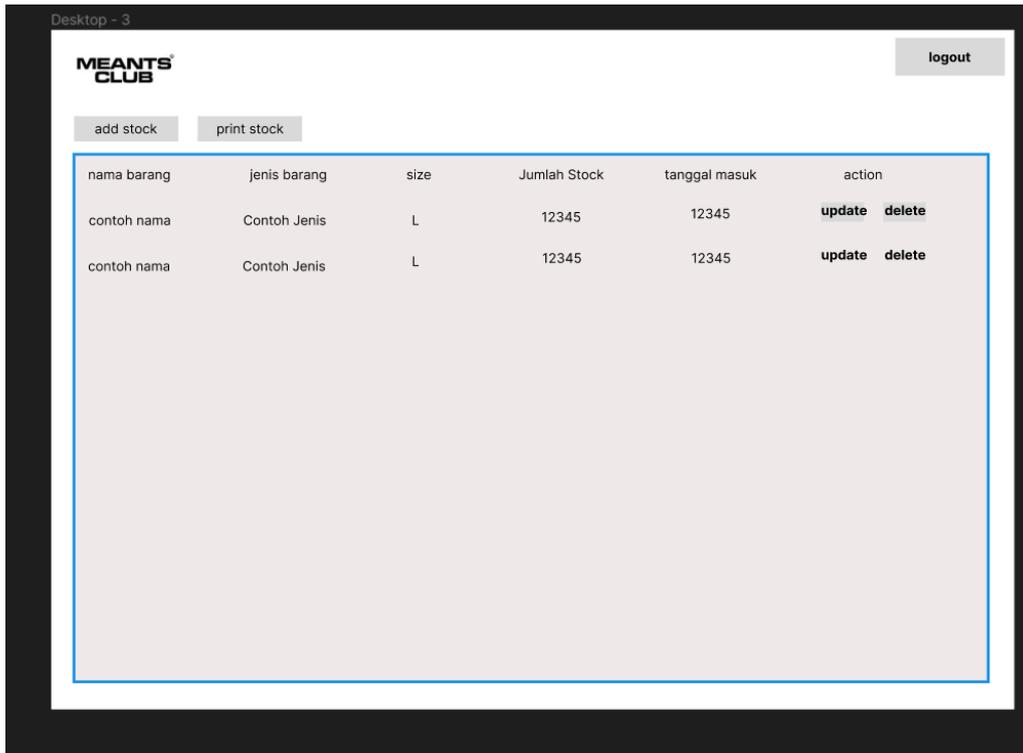
3.2.8 Rancangan Mockup Desain Aplikasi Menggunakan Figma

Berikut ini merupakan bentuk rancangan mockup Aplikasi menggunakan figma:



Gambar 3.11 Mockup Desain Halaman Login di figma

Halaman login dibuat simple karena Aplikasi ini hanya digunakan oleh pengguna internal dan tidak membutuhkan registrasi user karena user akan didaftarkan secara manual melalui database.

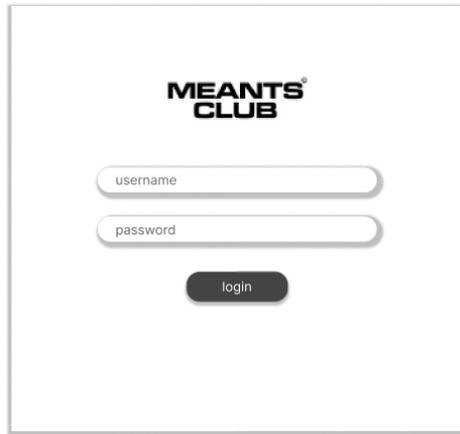


Gambar 3.12 Mockup Desain Halaman Admin di Figma

Halaman admin disini adalah halaman utama dimana semua operasi Aplikasi dilakukan disini gambar diatas hanyalah mockup sebagai acuan tata letak desain. Jenis font yang akan digunakan adalah font inter dengan size 20 regular dengan warna hitam. Warna background putih.

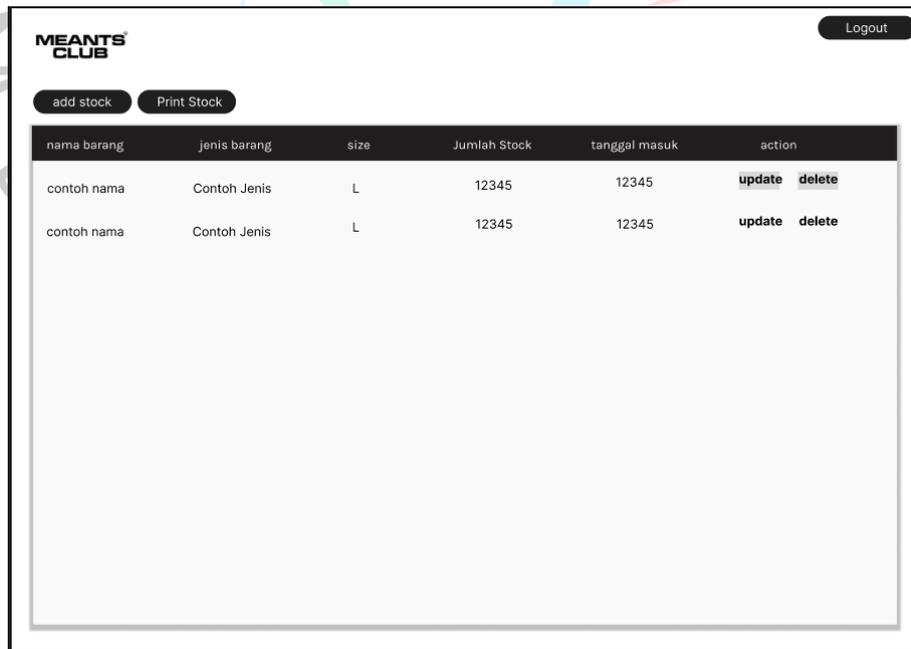
Warna dan desain yang simple seperti ini dipilih karena demi kemudahan operasi supaya tidak sulit untuk admin menemukan tombol yang harus digunakan.

Berikut adalah hasil desain UI:



The login form features the 'MEANTS CLUB' logo at the top center. Below the logo are two input fields: 'username' and 'password', each with a rounded rectangular border. A dark 'login' button is positioned centrally below the password field.

Gambar 3.13 UI Halaman login



The admin dashboard includes the 'MEANTS CLUB' logo and a 'Logout' button in the top right corner. Below the logo are two buttons: 'add stock' and 'Print Stock'. The main content is a table with the following data:

nama barang	jenis barang	size	Jumlah Stock	tanggal masuk	action
contoh nama	Contoh Jenis	L	12345	12345	update delete
contoh nama	Contoh Jenis	L	12345	12345	update delete

Gambar 3.14 UI Halaman Admin

3.2.9 DFD (Data Flow Diagram)

Diagram aliran data (data flow diagram) adalah jenis diagram yang digunakan untuk menggambarkan aliran data dalam suatu sistem informasi. Diagram ini menggambarkan bagaimana data diproses, disimpan, dan bergerak dari satu entitas ke entitas lainnya dalam sistem.

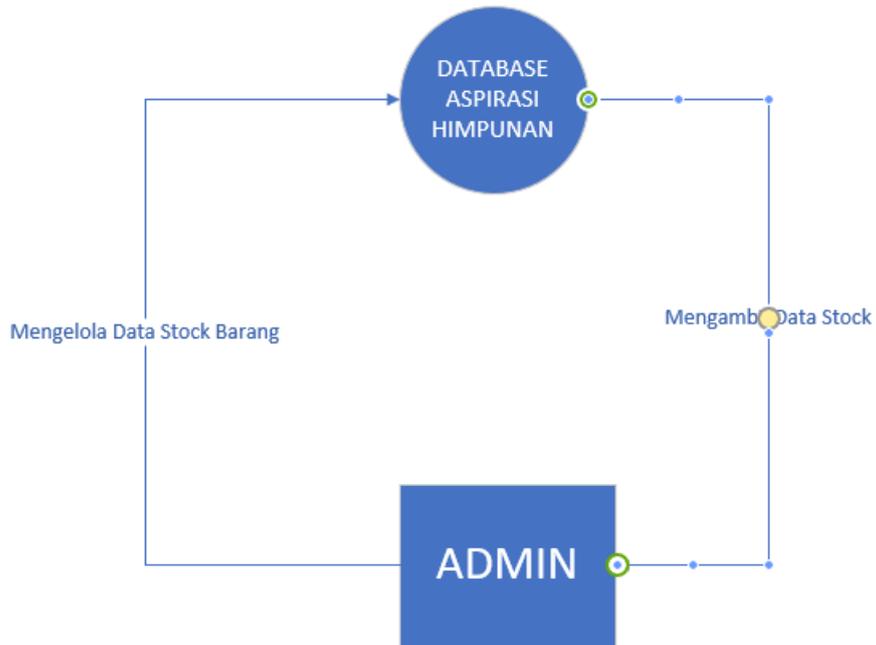
1. Data flow diagram (DFD) terdiri dari beberapa simbol dasar, termasuk:
2. Proses: Mewakili aktivitas yang memproses data. Proses ini dapat berupa perhitungan, transformasi, pengambilan, atau tindakan lainnya terhadap data.
3. Aliran Data: Mewakili aliran data antara entitas atau proses dalam sistem. Aliran data digambarkan sebagai panah yang mengindikasikan arah aliran data.
4. Entitas Eksternal: Mewakili sumber atau tujuan data yang berinteraksi dengan sistem. Entitas eksternal dapat berupa pengguna, sistem eksternal, atau perangkat lain yang berinteraksi dengan sistem.
5. Penyimpanan Data: Mewakili tempat penyimpanan data dalam sistem. Ini dapat berupa basis data, file, atau penyimpanan lainnya.

DFD membantu dalam memodelkan aliran data dalam sistem secara visual, memperjelas bagaimana data dikumpulkan, diproses, dan disimpan. Mereka memungkinkan analis sistem untuk memahami dan mengkomunikasikan secara efektif tentang bagaimana sistem bekerja dan bagaimana entitas dalam sistem berinteraksi satu sama lain.

DFD dapat digunakan dalam berbagai tahap pengembangan sistem, mulai dari analisis kebutuhan, desain sistem, hingga pemeliharaan dan pembaruan sistem. Mereka membantu dalam identifikasi kesalahan, redundansi, atau masalah dalam aliran data, serta dalam merancang solusi yang efisien dan efektif.

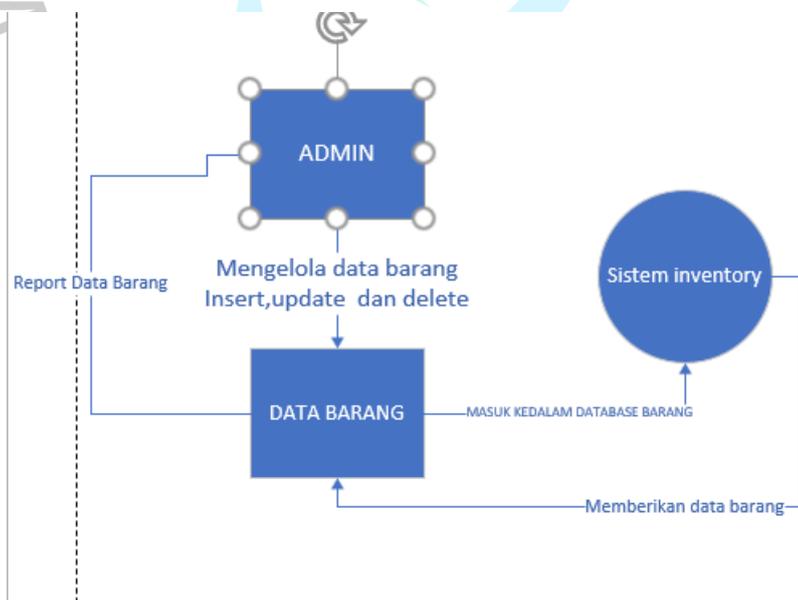
Berdasarkan kompleksitas nya DFD diurutkan berdasarkan level kerumitan dari 0 hingga 2 tapi disini penulis hanya akan menunjukkan DFD level 0 dan 1 saja karena secara kompleksitas aplikasi ini hanya mencakup admin dan database dan tidak ada entitas pengguna yang menjadi aktor dalam operasi aplikasi ini.

a) Level 0



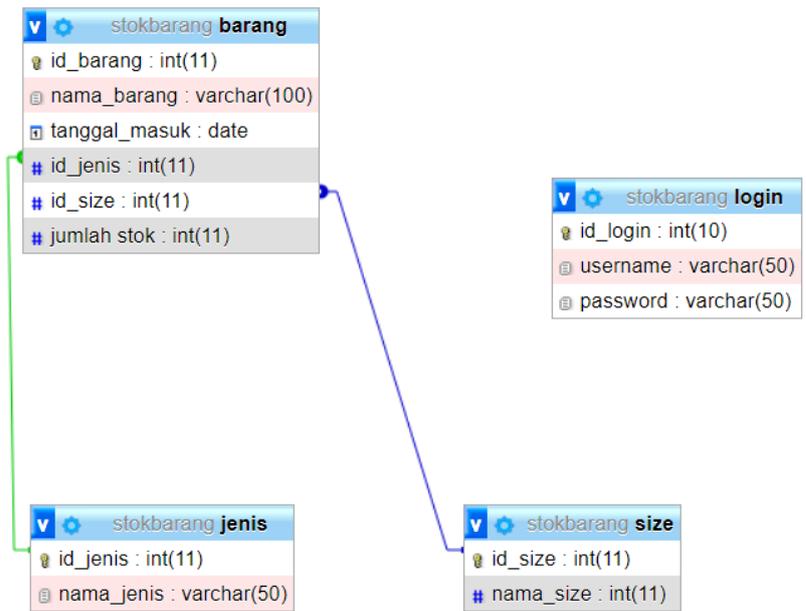
Gambar 3.15 DFD level 0

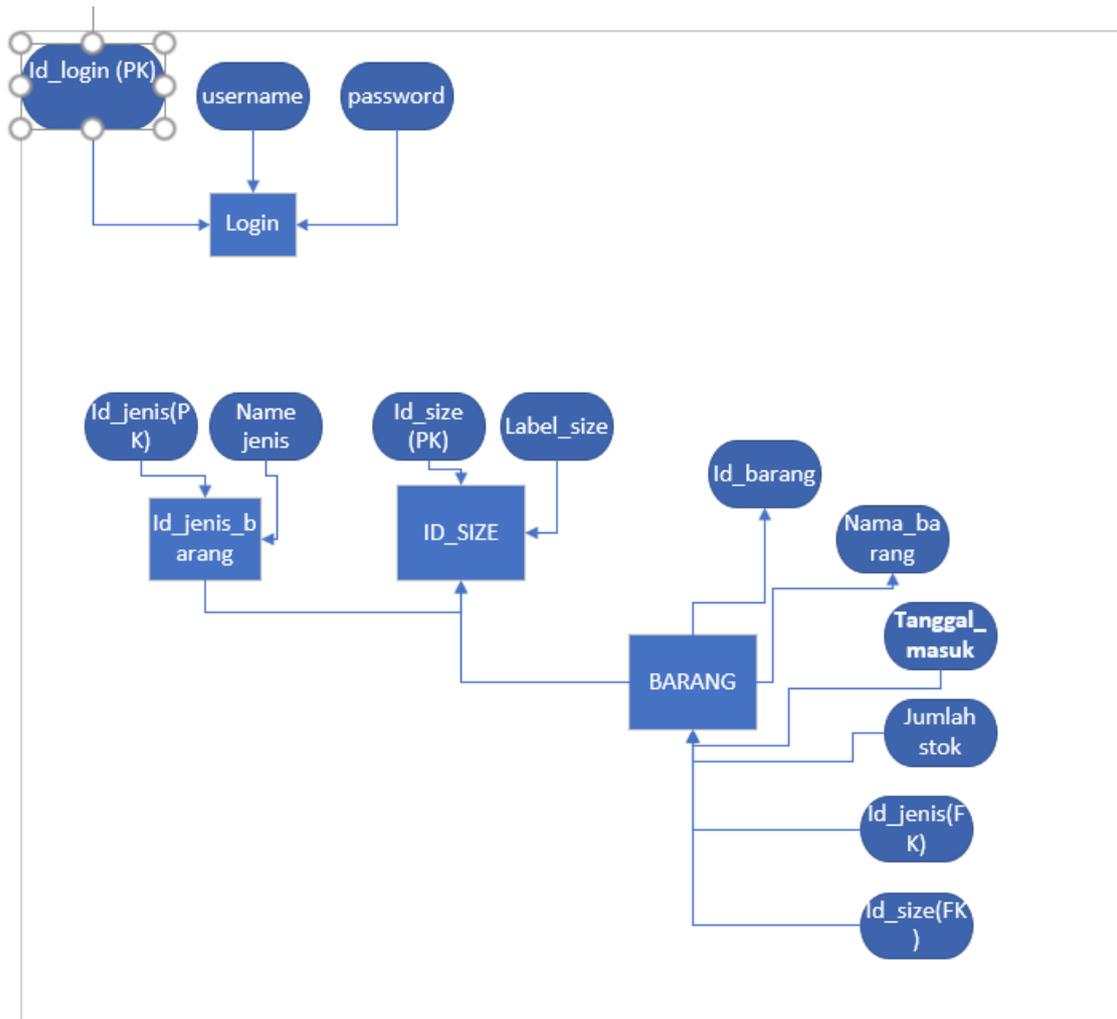
b) Level 1



Gambar 3.16 DFD level 1

3.2.10 Entity Relationship Diagram





Gambar 3.17 ERD

3.2.11 Rancangan Database

Database memiliki peran yang sangat penting dalam sebuah aplikasi. Berikut ini adalah beberapa alasan mengapa database penting dalam sebuah aplikasi:

- **Penyimpanan Data:** Database berfungsi sebagai tempat penyimpanan yang terstruktur untuk data aplikasi. Semua informasi yang dibutuhkan oleh aplikasi, seperti profil pengguna, konten, dan transaksi, disimpan dalam database. Dengan adanya database, data dapat diorganisir dengan baik, diakses dengan cepat, dan dapat diakses oleh banyak pengguna secara bersamaan.
- **Integritas Data:** Database menyediakan mekanisme untuk memastikan integritas data. Ini berarti data yang disimpan dalam database harus memenuhi aturan dan batasan yang ditentukan. Contohnya, jika aplikasi memiliki fitur pengguna yang tidak boleh memiliki nama yang sama, database dapat menerapkan aturan untuk memastikan bahwa tidak ada duplikat nama dalam basis data. Hal ini membantu menjaga konsistensi dan keandalan data.
- **Akses dan Manipulasi Data:** Database menyediakan bahasa kueri, seperti SQL (Structured Query Language), yang memungkinkan pengguna atau aplikasi untuk mengakses dan memanipulasi data dengan mudah. Bahasa kueri ini menyediakan kemampuan untuk mengambil data spesifik, melakukan pembaruan, penghapusan, atau penggabungan data. Dengan adanya database, pengguna aplikasi dapat dengan mudah mencari, menyortir, dan menganalisis data sesuai kebutuhan mereka.
- **Keamanan Data:** Database menyediakan fitur keamanan yang penting dalam melindungi data sensitif. Dengan menggunakan izin akses dan hak pengguna yang terkendali, database memungkinkan pengaturan tingkat akses yang tepat. Hal ini memastikan bahwa hanya pengguna yang memiliki izin yang sesuai yang dapat mengakses, memperbarui, atau menghapus data tertentu. Dalam database yang baik, juga dapat diterapkan metode enkripsi untuk melindungi data saat istirahat.
- **Skalabilitas dan Kinerja:** Database dirancang untuk mengelola dan menyimpan data dalam skala yang besar. Dengan menggunakan teknik seperti indeks, pemartisian, atau replikasi, database dapat dioptimalkan untuk mencapai kinerja tinggi dan skalabilitas. Ini penting ketika aplikasi memiliki pertumbuhan

pengguna yang signifikan atau ketika data yang disimpan semakin banyak dari waktu ke waktu.

- Pemulihan Data: Database sering dilengkapi dengan fitur pemulihan yang memungkinkan pemulihan data setelah terjadi kegagalan sistem atau bencana. Melalui pencadangan berkala dan teknik pemulihan, database dapat mengurangi risiko kehilangan data dan memastikan kelangsungan bisnis.

Berikut adalah rancangan database yang saya buat dan bagaimana saya membuat databasenya :

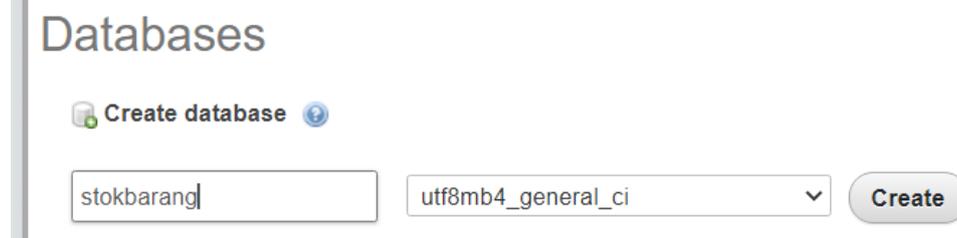
1. Membuat database baru

Disini saya menggunakan sql dan dijalankan melalui phpMyAdmin yang saya install di localhost.



Gambar 3.18 phpmyadmin

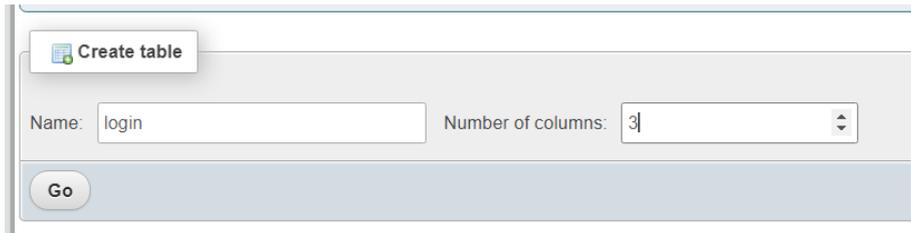
Lalu saya mengisi nama database dan menekan tombol create untuk membuat database baru



Gambar 3.19 Membuat database baru

2. Membuat tabel login

Tabel login dibutuhkan untuk menyimpan data username dan juga password yang dibutuhkan untuk autentikasi login kedalam aplikasi



Gambar 3.20 Membuat tabel login

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_login	int(10)			No	None			Change Drop More
2	username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	password	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

Gambar 3.21 Tampilan tabel login

3. Membuat tabel jenis barang

Tabel jenis barang ini dibutuhkan untuk memasukan jenis barang menjadi beberapa kategori barang yang bisa di update

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_jenis	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama_jenis	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

Gambar 3.22 Tampilan Tabel Jenis Barang

4. Membuat tabel size barang

Tabel size ini dibutuhkan untuk mengkategorikan barang menjadi beberapa size

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_size	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 nama_size	int(11)			No	None			Change Drop More

Gambar 3.23 Tampilan Tabel Jenis Barang1

5. Membuat tabel barang

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_barang	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 nama_barang	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 tanggal_masuk	date			No	None			Change Drop More
<input type="checkbox"/>	4 id_jenis	int(11)			No	None			Change Drop More
<input type="checkbox"/>	5 id_size	int(11)			No	None			Change Drop More
<input type="checkbox"/>	6 jumlah_stok	int(11)			No	None			Change Drop More

Gambar 3.24 Tampilan Tabel J Barang

Membuat tabel barang dengan id_barang sebagai primary key dan juga merelasikan tabel jenis dan size dengan cara memasukan id_jenis dan id_size sebagai foreign key.