

BAB III

METODOLOGI PENELITIAN

3.1 Subyek dan Obyek Penelitian

Penelitian ini merupakan penelitian dengan menggunakan metode kualitatif deskriptif. Beberapa langkah yang dilakukan dalam pengumpulan data penelitian ini adalah melakukan wawancara dengan karyawan yang memahami kondisi proses bisnis di Toko Emas Cahaya.

Subyek pada penelitian ini adalah sebuah Toko Emas Cahaya yang terletak di Pasar Kesugihan, Cilacap. Obyek dari penelitian yang dilakukan terdapat pada perancangan sistem informasi *inventory*.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat

Alat yang digunakan untuk melaksanakan penelitian ini, antara lain:

1. Laptop Asus ROG G531GD sebagai alat utama dalam menyusun proposal pada penelitian Tugas Akhir I dan II
2. *Smartphone* Android Asus Zenfone Max Pro M1 untuk media komunikasi dan informasi.
3. *Sublime Text* Versi 3.2.2 build 3211
4. XAMPP Versi 3.2.4 untuk mengaktifkan dan menghubungkan mySQL.
5. PHP versi 7.4 untuk megubah halaman menjadi halaman dinamis.
6. *Framework* Bootstrap Versi 5.1.3 untuk implementasi pengembangan webite.
7. *Microsoft Office Professional Plus* 2019 untuk mengolah data, menulis dan mengembangkan hasil penelitian.
8. *Google Chrome* Versi 97.0.4692.99 (64 bit) untuk mencari informasi referensi penelitian.

3.2.2 Bahan

Bahan penelitian yang digunakan dalam melakukan penelitian ini antara lain:

1. Data Primer

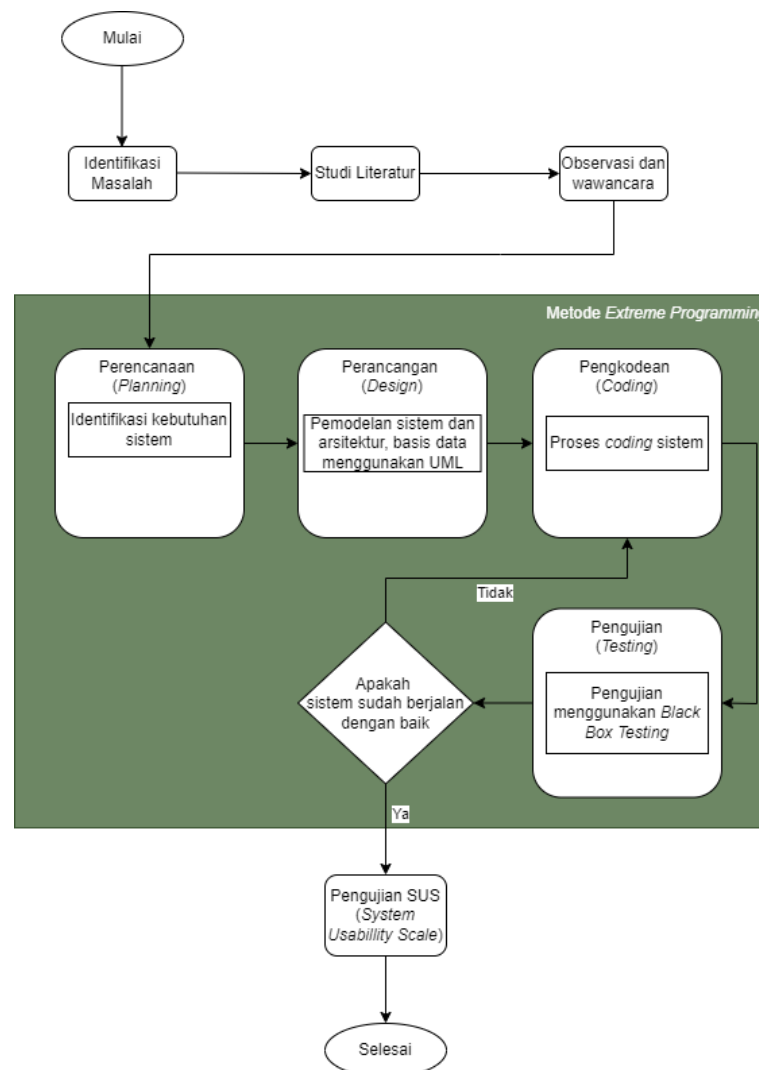
Pengumpulan data primer ini dilakukan dengan cara melakukan komunikasi langsung atau wawancara dengan pihak terkait. Data ini diperoleh dari saudara Saean sebagai salah satu karyawan Toko Emas Cahaya pada 27 Juni 2021 untuk memperoleh informasi yang lebih rinci terhadap masalah yang akan diteliti.

2. Data Sekunder

Data sekunder merupakan data yang diperoleh secara tidak langsung. Data sekunder dalam penelitian ini didapatkan dengan mencari literatur dari kepustakaan berupa artikel, jurnal, *e-book*, situs internet dan lainnya yang memiliki kaitan dengan penelitian ini.

3.3 Diagram Alir Penelitian

Penelitian yang dilakukan dimulai dari studi literatur yaitu mengkaji tinjauan pustaka dari berbagai jurnal yang berkaitan dengan topik penelitian. Selanjutnya melakukan observasi dan wawancara dengan narasumber. Memasuki tahap pertama metode Extreme Programming yaitu melakukan Perencanaan (*Planning*) yang berisi identifikasi kebutuhan. Kemudian pada tahap kedua melakukan Perancangan (*Design*) dengan merancang *Unified Modeling Language* (UML) diantaranya membuat *usecase diagram*, *activity diagram*, *sequence diagram* dan *class diagram*. Pada tahap ketiga melakukan Pengkodean (*Coding*) yaitu penerapan pemodelan yang telah dirancang dengan memakai bahasa pemrograman. Kemudian tahap terakhir yaitu Pengujian (*Testing*) yaitu melakukan pengujian sistem sehingga ditemukan kesalahan-kesalahan yang ada, dan dapat mengetahui bahwa sistem yang dibuat sudah sesuai dengan kebutuhan. Tahapan dari penelitian ini dapat dilihat pada gambar 3.1 sebagai berikut:



Gambar 3. 1 Alir Penelitian

3.3.1 Identifikasi Masalah

Tahap identifikasi masalah merupakan tahap awal dalam menentukan permasalahan yang akan dijadikan topik penelitian. Masalah yang diambil dalam penelitian ini merupakan topik yang belum memiliki solusi untuk mengelola stok barang dan pencatatan transaksi dari Toko Emas Cahaya. Permasalahan yang didapatkan adalah pencatatan stok barang dan transaksi penjualan masih bersifat manual menggunakan buku, sehingga memungkinkan dapat diatasi dengan teknologi informasi yang berbasis *website*.

3.3.2 Studi Literatur

Studi literatur adalah mengumpulkan data dan informasi sesuai dengan permasalahan penelitian yang dijadikan acuan yang terkait dengan perancangan sistem informasi *inventory* di Toko Emas Kesugihan.

3.3.3 Pengumpulan Data

Pengumpulan data dan informasi diperoleh melalui wawancara secara langsung terhadap salah satu karyawan dari Toko Emas Cahaya dengan mengajukan beberapa pertanyaan yang terkait dengan perancangan sistem informasi yang akan dibuat.

3.3.4 Implementasi metode *Extreme Programming*

Proses pengembangan dari metode *Extreme Programming* terdiri dari empat tahap, diantaranya:

1. Perencanaan (*Planning*)

Tahap awal pada pengembangan sistem informasi *inventory* mengidentifikasi masalah, mengidentifikasi kebutuhan sistem dan penentuan jadwal pelaksanaan pembangunan sistem. Tahapan *planning* berisi tentang *user stories*, *values*, *acceptance test criteria* dan *iteration plan*. Masing-masing elemen akan dibahas pada sub bab berikut.

a. *User Stories*

User Stories merupakan urutan dari beberapa cerita yang didapatkan dari hasil diskusi antara pengembang (*developer*) dan Pengguna (*user*). *User Stories* akan dijadikan acuan pembuatan fitur yang harus tersedia pada sistem. Berikut adalah kutipan cerita yang penulis dapatkan dari user.

Tabel 3. 1 Story

<p>“Saya memerlukan sebuah sistem untuk mencatat stok barang dan transaksi yang berbasis website. Sistem dapat mengelola data barang dengan keterangan kode barang, jenis barang, nama barang dan berat. Kemudian sistem bisa menghitung jumlah stok yang masuk dan keluar ketika ada pelanggan yang membeli maupun</p>

menjual emas. Sistem ini harus bisa mencatat transaksi masuk dan transaksi keluar. Kemudian, sistem ini juga harus bisa menghitung jumlah pemasukan dan pengeluaran dari hasil transaksi pada Toko Emas Cahaya.

Ketika *story* telah didapatkan, tahapan yang harus dilakukan selanjutnya adalah pembuatan *user stories*, untuk mendapatkan hasil analisis kebutuhan yang baik secara fungsional maupun non fungsional. Berikut adalah beberapa *user stories* yang telah dibuat berdasarkan *story* yang telah didapatkan.

Tabel 3. 2 *User Stories*

No.	<i>User Story</i>
US-01	Sistem dapat mengelola data barang dengan keterangan kode barang, jenis barang, nama barang dan berat.
US-02	sistem bisa menghitung jumlah stok yang masuk dan keluar ketika ada pelanggan yang membeli maupun menjual emas
US-03	Sistem ini harus bisa mencatat transaksi masuk dan transaksi keluar.
US-04	sistem ini juga harus bisa menghitung jumlah pemasukan dan pengeluaran dari hasil transaksi pada Toko Emas Cahaya.

Dari hasil penjabaran *story* tersebut, telah terbentuk beberapa *user stories* yang dapat dijadikan sebagai acuan fitur pokok yang akan dibangun didalam sistem. Mengingat *extreme programming* merupakan metode pengembangan perangkat lunak yang bersifat fleksibel, *user stories* dapat berubah sewaktu-waktu.

b. Values

Values merupakan tahapan menentukan prioritas pada *user stories* yang telah dibuat. Nilai yang akan diberikan terhadap *user stories* bergantung pada tingkat kesulitan dan kompleksnya fitur tersebut. *User stories* dengan value tertinggi akan di prioritaskan dan diimplementasikan segera mungkin. Berikut adalah hasil pemberian value pada *user stories* yang telah dibuat, yang kemudian diurutkan berdasarkan dari values tertinggi hingga terendah.

Tabel 3. 3 *User Story Values*

No.	<i>User Story</i>	<i>Values</i>
US-02	sistem bisa menghitung jumlah stok yang masuk dan keluar ketika ada pelanggan yang membeli maupun menjual emas.	4
US-01	Sistem dapat mengelola data barang dengan keterangan kode barang, jenis barang, nama barang dan berat.	3
US-04	Sistem ini juga harus bisa menghitung jumlah pemasukan dan pengeluaran dari hasil transaksi pada Toko Emas Cahaya.	2
US-03	Sistem ini harus bisa mencatat transaksi masuk dan transaksi keluar.	1

Mengacu pada tabel diatas, maka *user stories* yang memiliki tingkat kesulitan tertinggi yaitu **US-02**, selanjutnya diikuti **US-01**, **US-04**, **US-03**.

c. Acceptance Test Criteria

Berdasarkan pada hasil analisa yang telah dilakukan, berikut ini merupakan detail *Acceptance Test Criteria* dari seluruh *user stories* yang telah dibuat.

Tabel 3. 4 User Stories US-01

No.	<i>User Story</i>
US-01	Sistem dapat mengelola data barang dengan keterangan kode barang, jenis barang, nama barang dan berat.

Dari hasil penjabaran *user stories* US-01 serta analisa, maka didapatkan beberapa *acceptance criteria* sebagai berikut:

1. Terdapat empat field inti yaitu kode barang, jenis barang, nama barang dan berat.
2. Data barang di inputkan secara manual melalui sistem

Tabel 3. 5 User Stories US-02

No.	<i>User Story</i>
US-02	Sistem bisa menghitung jumlah stok yang masuk dan keluar ketika ada pelanggan yang membeli maupun menjual emas.

Dari hasil penjabaran *user stories* US-02 serta analisa, maka didapatkan beberapa *acceptance criteria* sebagai berikut:

1. Ketika menginput transaksi masuk maupun keluar, maka jumlah stok harus otomatis bertambah maupun berkurang.
2. Berelasi dengan menu data barang.

Tabel 3. 6 User Stories US-03

No.	<i>User Story</i>
US-03	Sistem ini harus bisa mencatat transaksi masuk dan transaksi keluar.

Dari hasil penjabaran *user stories* US-03 serta analisa, maka didapatkan beberapa *acceptance criteria* sebagai berikut:

1. Transaksi diinputkan manual kedalam sistem.
2. Berelasi dengan menu stok barang.

Tabel 3. 7 User Stories US-04

No.	User Story
US-04	Sistem ini juga harus bisa menghitung jumlah pemasukan dan pengeluaran dari hasil transaksi pada Toko Emas Cahaya.

Dari hasil penjabaran *user stories* US-04 serta analisa, maka didapatkan beberapa *acceptance criteria* sebagai berikut:

1. Berelasi dengan menu transaksi.
2. Seluruh aktor bisa melihat.

d. Iteration Plan

Iteration plan adalah suatu proses diskusi dan perencanaan fase iterasi pengembangan sistem mengenai sistem yang akan dibangun untuk menentukan berapa lama proses pengerjaan sistem. Berdasarkan pada value dan prioritas *user stories* yang sudah ditentukan, dapat dibuat rencana iterasi sebagai berikut:

Tabel 3. 8 Rencana Iterasi

Story	Time (weeks)	I-0	I-1	I-2
US-02	2	v		
US-01	1		v	
US-04	1			v
US-03	1			v

Tabel rencana iterasi diatas menunjukkan bahwa dalam pengembangan sistem ini terdapat 3 iterasi pengulangan, yang mana *user stories* US-02 berperan sebagai iterasi pertama, kemudian US-01 berperan sebagai iterasi kedua, US-04 dan US-03 berperan sebagai iterasi ketiga.

2. Perancangan (*Design*)

Setelah ditemukan masalah dan kebutuhan sistem, melakukan perancangan *Class Responsibility Collaborator Cards (CRC Cards)* pemodelan sistem dan arsitektur, pemodelan basis data dengan melibatkan *user* atau pelanggan.

a. *Class Responsibility Collaborator Cards (CRC Cards)*

CRC Cards merupakan diagram yang berfungsi untuk menunjukkan interaksi antar objek, beserta tanggung jawab yang bisa dilakukan objek tersebut. Berdasarkan hasil melakukan analisa pada *user stories* yang telah dibuat, selanjutnya yang dilakukan peneliti adalah membuat desain *CRC Cards*, terdapat beberapa *cards* yang terbentuk dari hasil menganalisa *user stories* tersebut. Berikut ini adalah desain *CRC Cards* yang telah dibuat beserta penjelasannya:

1. *CRC Cards US-01*

Dari hasil menganalisa *user stories* US-01, ditemukan satu objek dari kata ini yaitu **barang**, yang kemudian objek tersebut akan dijadikan sebagai nama *class*, artinya dari US-01 ini hanya didapatkan sebuah *cards* sebagai berikut.

Tabel 3. 9 *CRC Cards* Barang

Class: Barang	
Responsibility	Collaborator
<ul style="list-style-type: none"> - Menampilkan kode karang - Menampilkan jenis barang - Menampilkan nama barang - Menampilkan berat - Menampilkan keterangan 	

2. *CRC-Cards US-02*

Dari hasil menganalisa *user stories* US-02, ditemukan satu objek dari kata ini yaitu **Stok Barang**, yang kemudian objek tersebut akan dijadikan

sebagai nama *class*, artinya dari US-02 ini hanya didapatkan sebuah *cards* sebagai berikut.

Tabel 3. 10 CRC Cards Stok Barang

Class: Stok Barang	
Responsibility:	Collaborator:
- Menampilkan kode barang	- kode barang
- Menampilkan jenis barang	- jenis barang
- Menampilkan nama barang	- nama barang
- Menampilkan berat	- berat
- Menampilkan harga	
- Menampilkan jumlah stok	

Dari *cards class* tersebut, terdapat empat buah collabolorator yaitu cards kode barang, jenis barang, nama barang dan berat.

3. CRC-Cards US-03

Dari hasil menganalisa *user stories* US-03, ditemukan dua objek dari kata ini yaitu **Transaksi Masuk dan Transaksi Keluar**, yang kemudian objek tersebut akan dijadikan sebagai nama *class*, artinya dari US-03 ini hanya didapatkan sebuah *cards* sebagai berikut.

Tabel 3. 11 CRC Cards Transaksi Masuk

Class: Transaksi Masuk	
Responsibility:	Collaborator:
- Menampilkan tanggal	- kode barang
- Menampilkan id transaksi	- berat
- Menampilkan kode barang	
- Menampilkan berat	
- Menampilkan jumlah	
- Menampilkan nama penjual	
- Menampilkan alamat penjual	
- Menampilkan keterangan	

Tabel 3. 12 CRC Cards Transaksi Keluar

Class: Transaksi Keluar	
Responsibility:	Collaborator:
<ul style="list-style-type: none"> - Menampilkan id transaksi - Menampilkan kode barang - Menampilkan berat - Menampilkan jumlah - Menampilkan nama pembeli - Menampilkan alamat pembeli - Menampilkan keterangan 	<ul style="list-style-type: none"> - kode barang - berat

Dari *cards class* tersebut, terdapat dua buah *collaborator* yaitu *cards* kode barang dan berat.

4. CRC-Cards US-04

Dari hasil menganalisa *user stories* US-04, ditemukan dua objek dari kata ini yaitu **Pemasukan dan Pengeluaran**, yang kemudian objek tersebut akan dijadikan sebagai nama *class*, artinya dari US-04 ini hanya didapatkan sebuah *cards* sebagai berikut.

Tabel 3. 13 CRC Cards Pemasukan

Class: Pemasukan	
Responsibility:	Collaborator:
<ul style="list-style-type: none"> - Menampilkan id transaksi - Menampilkan nama barang - Menampilkan pembeli - Menampilkan alamat - Menampilkan harga - Menampilkan berat - Menampilkan jumlah 	<ul style="list-style-type: none"> - id transaksi - nama barang - pembeli - alamat - berat - jumlah - harga

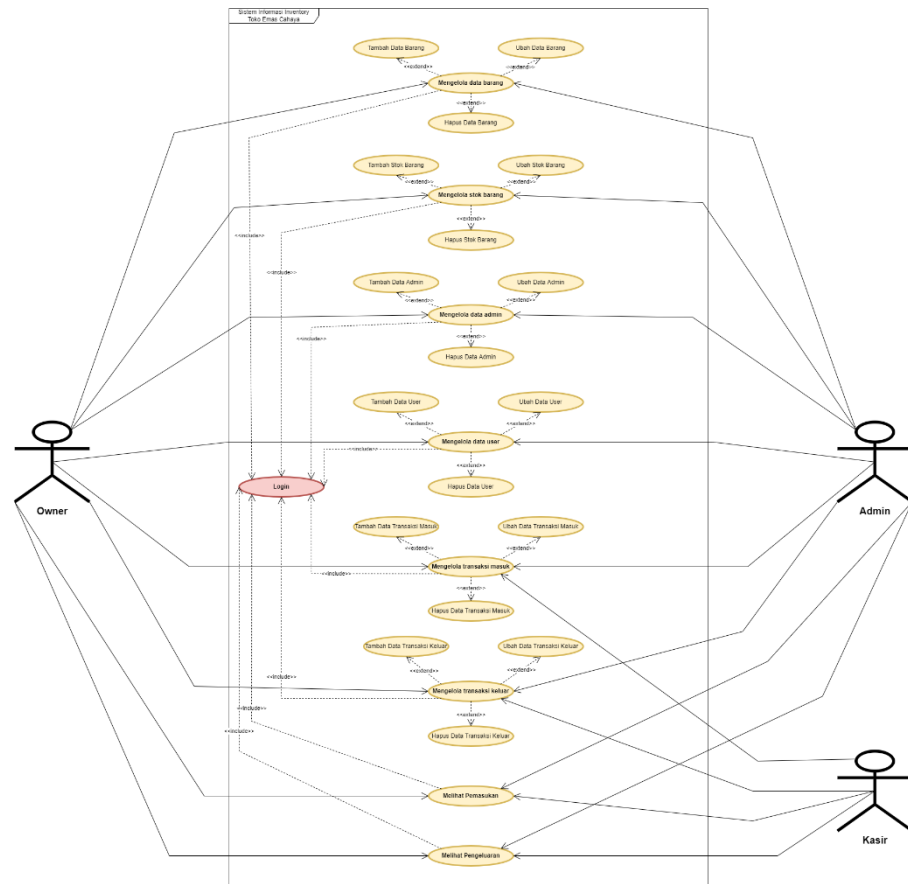
- Menampilkan total pembayaran	
--------------------------------	--

Tabel 3. 14 CRC Cards Pengeluaran

Class: Pengeluaran	
Responsibility:	Collaborator:
- Menampilkan id transaksi	- id transaksi
- Menampilkan nama barang	- nama barang
- Menampilkan penjual	- penjual
- Menampilkan alamat	- alamat
- Menampilkan harga	- berat
- Menampilkan berat	- jumlah
- Menampilkan jumlah	- harga
- Menampilkan total pembayaran	

Dari *cards class* tersebut, terdapat lima buah *collaborator* yaitu *cards* id transaksi, nama barang, berat, jumlah dan harga.

e. *Use case diagram*



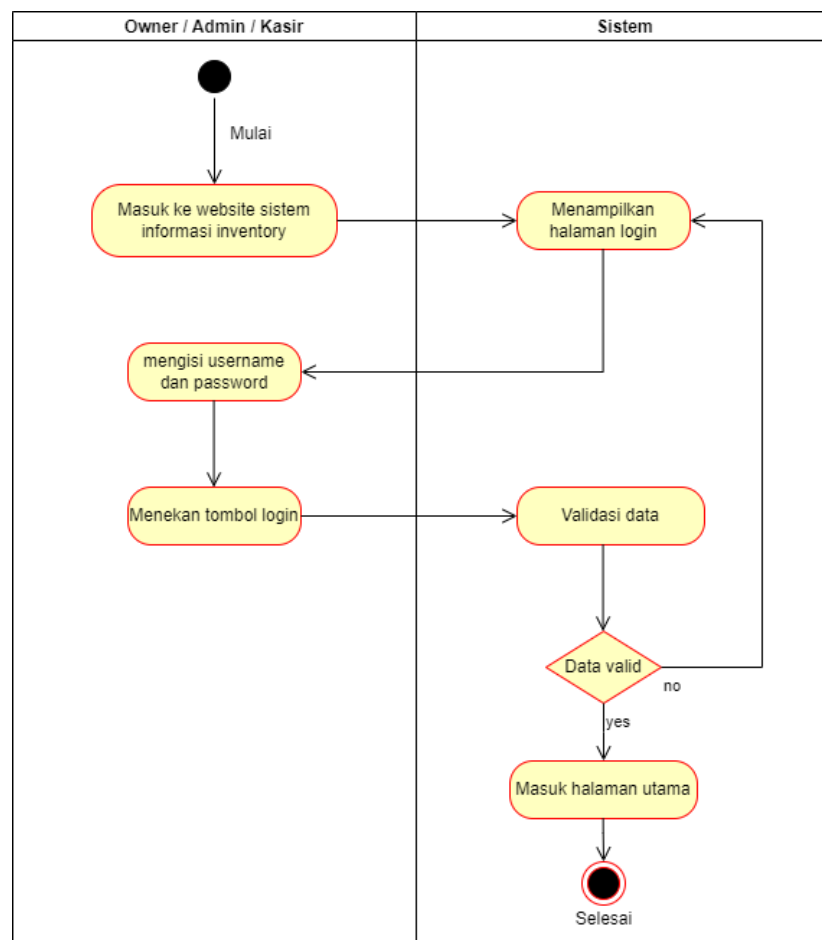
Gambar 3. 2 Use Case Sistem Informasi Inventory Toko Emas Cahaya

Gambar 3.2 merupakan gambar *use case diagram website* Sistem Informasi *Inventory* Toko Emas Cahaya dimana terdapat tiga aktor yaitu owner, admin dan kasir. Owner dan admin berperan sebagai pengelola sistem yang memiliki akses penuh dalam sistem. Owner dan admin memiliki beberapa atribut antara lain mengelola data admin, mengelola data user, mengelola data barang, mengelola data stok, mengelola data transaksi masuk, mengelola data transaksi keluar, mengelola laporan pemasukan dan mengelola pengeluaran. Kemudian, aktor kasir memiliki beberapa atribut antara lain kelola transaksi masuk, kelola transaksi keluar, kelola pemasukan dan kelola pengeluaran.

Sebelum admin dapat mengakses semua menu-menu tersebut, admin harus melakukan *login* terlebih dahulu kepada sistem.

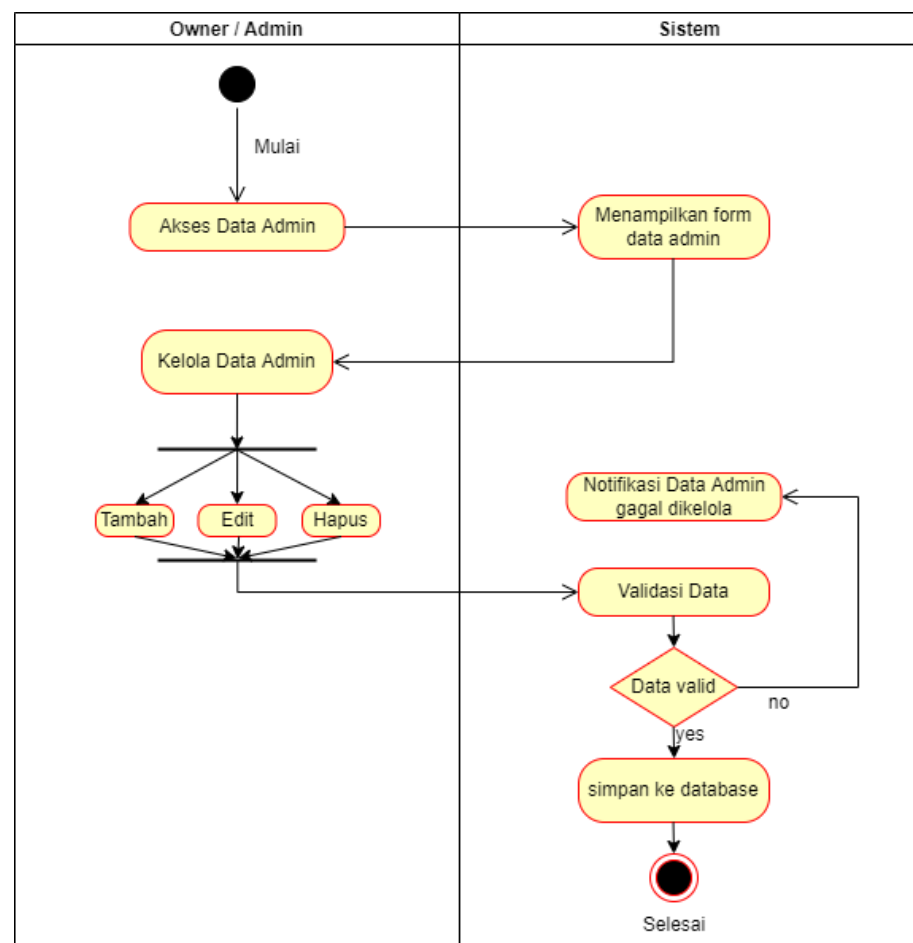
f. Activity Diagram

Penyusunan *activity diagram* didasarkan pada *use case diagram* pada tahap sebelumnya. Pemodelan *activity diagram* dijelaskan sebagai berikut dengan membaginya berdasarkan aktor yang terdapat dalam sistem. Penelitian ini terdapat 9 *activity diagram* yang terdiri dari *activity diagram login*, *activity diagram data admin*, *activity diagram data user*, *activity diagram data barang*, *activity diagram stok barang*, *activity diagram transaksi masuk*, *activity diagram transaksi keluar*, *activity diagram pemasukan* dan *activity diagram pengeluaran*.



Gambar 3. 3 Activity Diagram Login

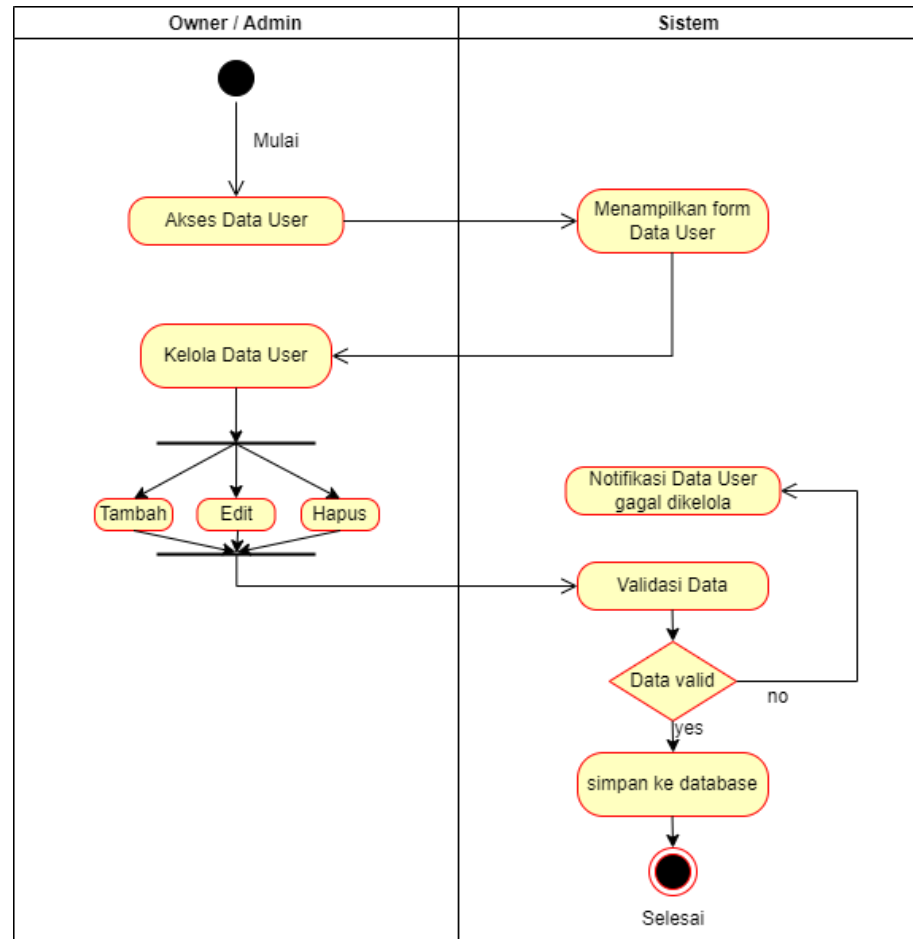
Berdasarkan pada gambar 3.3 maka dapat diberikan penjelasan yaitu Owner / Admin / Kasir masuk ke *website* sistem informasi *inventory* kemudian sistem menampilkan halaman *login*. Owner / Admin / Kasir mengisi *username* dan *password* dan menekan tombol *login*, sistem akan melakukan validasi data jika valid maka masuk ke halaman utama sistem jika tidak valid kembali ke halaman *login*.



Gambar 3. 4 Activity Diagram Kelola Data Admin

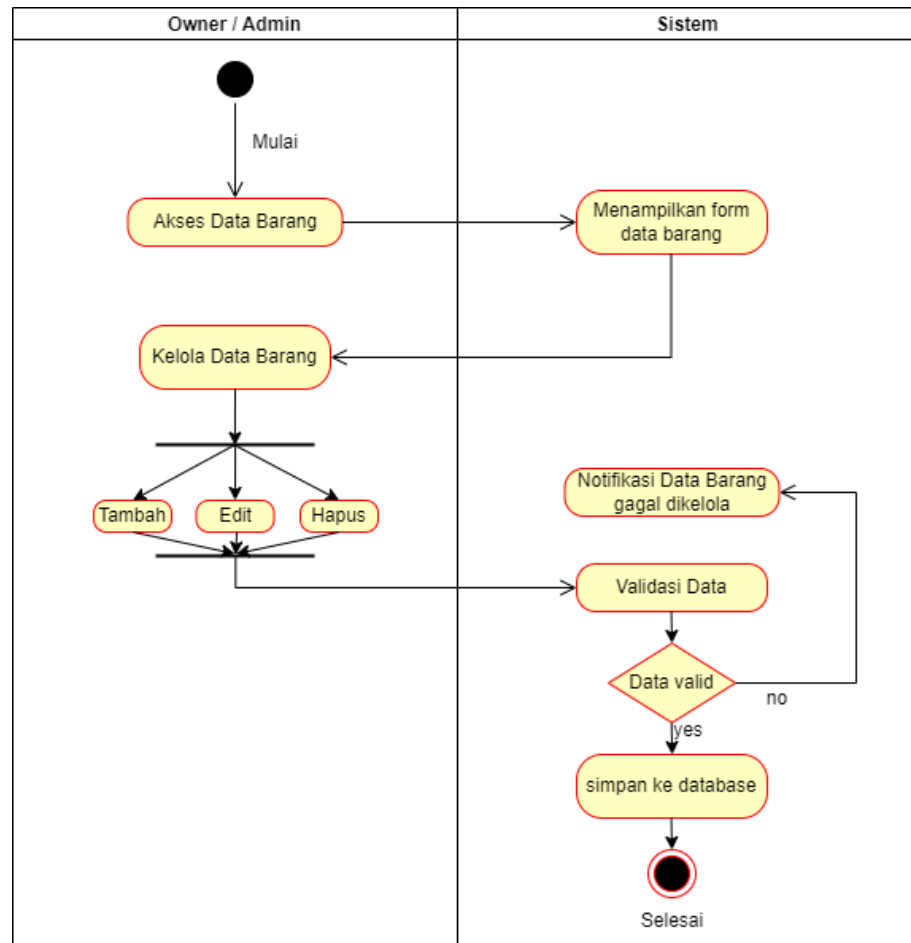
Berdasarkan gambar 3.4 menjelaskan aktivitas mengelola data admin yang dimana dimulai dengan mengakses data admin, kemudian kelola data admin yang terdiri dari menambah, mengubah dan menghapus. Setelah melakukan kelola data admin, maka sistem melakukan validasi data yang dimana pada bagian validasi data dilakukan pengecekan dan

memastikan semua *form* telah diisi oleh pengguna. Jika data *valid*, sistem akan menyimpan data ke *database* dan selesai. Jika tidak *valid*, sistem akan menampilkan notifikasi data gagal dikelola.



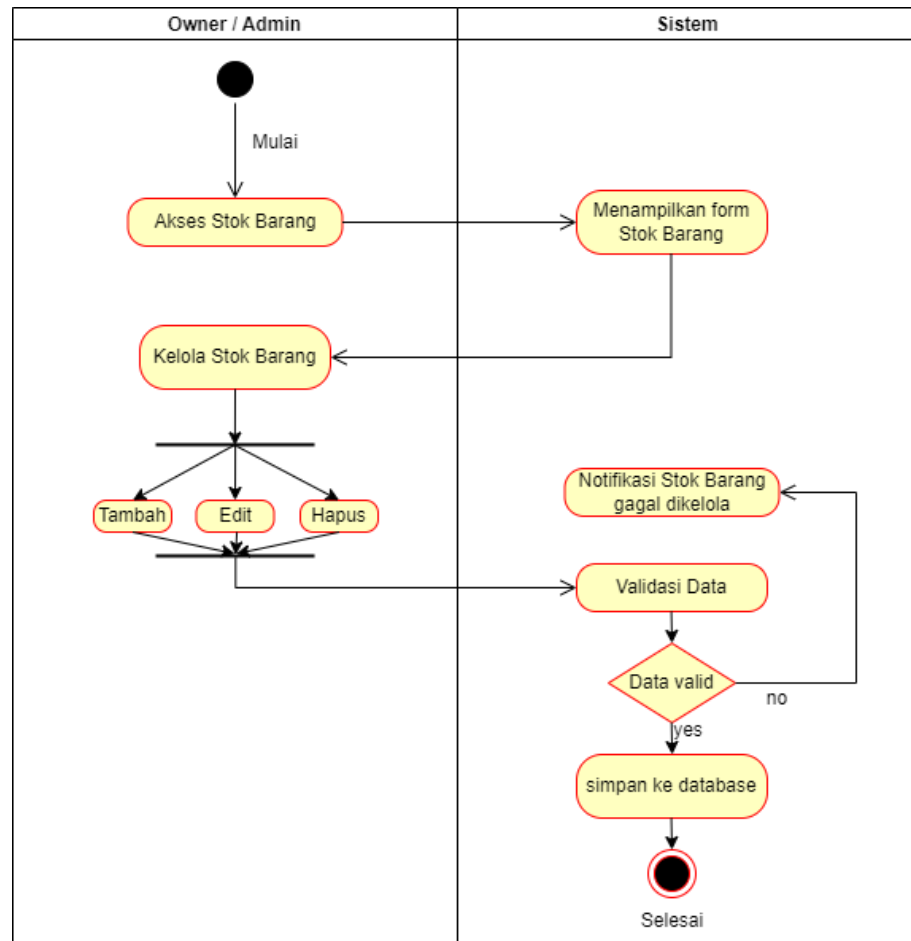
Gambar 3.5 Activity Diagram Kelola Data User

Berdasarkan gambar 3.5 menjelaskan aktivitas mengelola data user yang dimana dimulai dengan mengakses data user, kemudian kelola data user yang terdiri dari menambah, mengubah dan menghapus. Setelah melakukan kelola data user, maka sistem melakukan validasi data yang dimana pada bagian validasi data dilakukan pengecekan dan memastikan semua *form* telah diisi oleh pengguna. Jika data *valid*, sistem akan menyimpan data ke *database* dan selesai. Jika tidak *valid*, sistem akan menampilkan notifikasi data gagal dikelola.



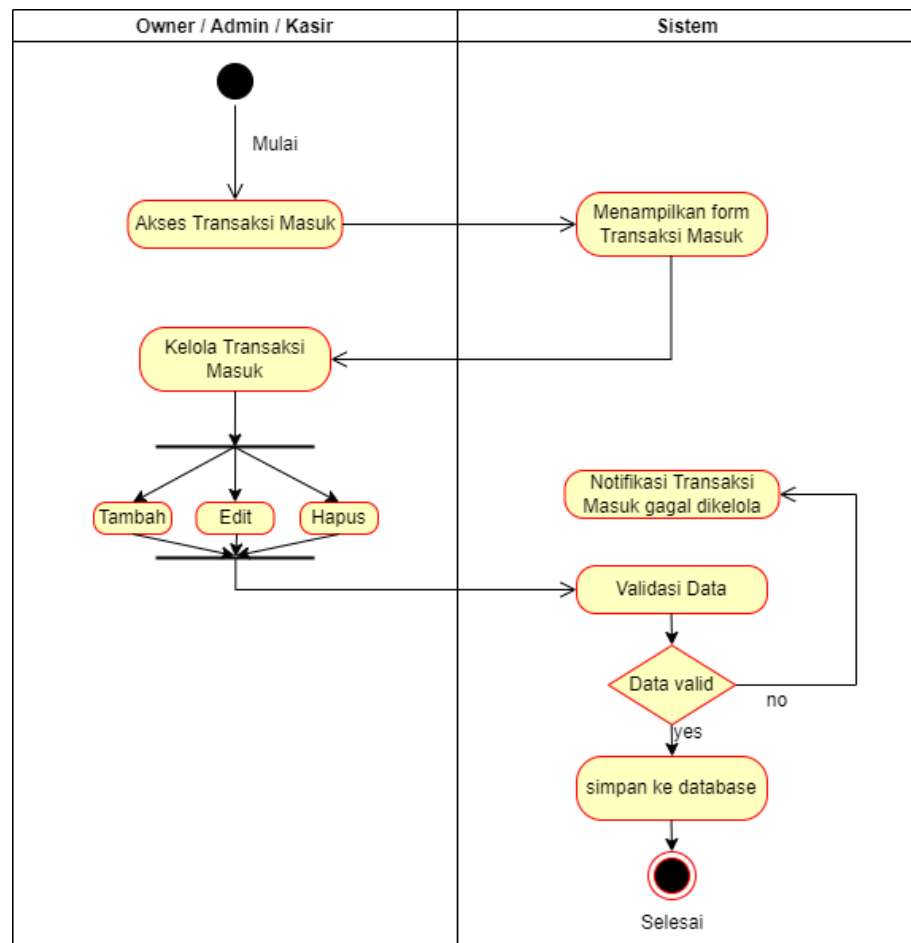
Gambar 3. 6 Activity Diagram Data Barang

Berdasarkan gambar 3.6 menjelaskan aktivitas mengelola data barang yang dimana dimulai dengan mengakses data barang, kelola data barang terdiri dari menambah, mengubah dan menghapus. Setelah melakukan kelola data barang, maka sistem melakukan validasi data yang dimana pada bagian validasi data dilakukan pengecekan dan memastikan semua *form* telah diisi oleh pengguna. Jika data *valid*, sistem akan menyimpan data ke *database* dan selesai. Jika tidak *valid*, sistem akan menampilkan notifikasi data gagal dikelola.



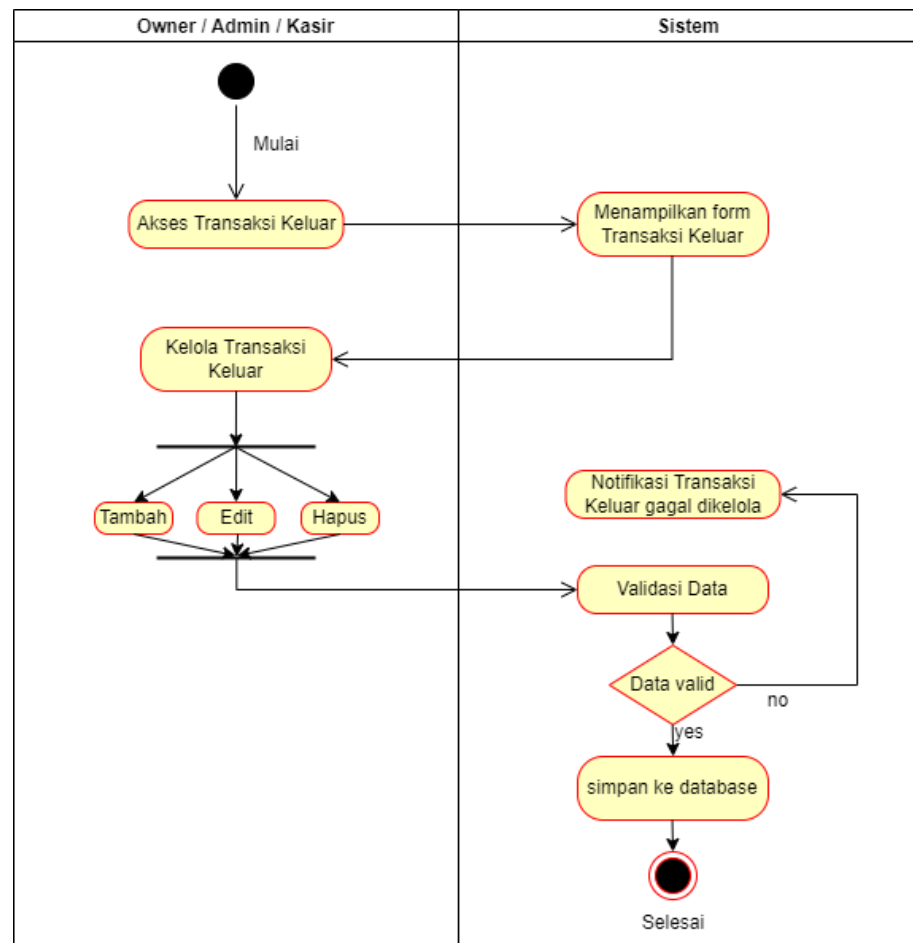
Gambar 3. 7 Activity Diagram Stok Barang

Berdasarkan gambar 3.7 menjelaskan aktivitas mengelola stok barang yang dimana dimulai dengan mengakses stok barang, kelola stok barang terdiri dari menambah, mengubah dan menghapus. Setelah melakukan kelola stok barang, maka sistem melakukan validasi data yang dimana pada bagian validasi data dilakukan pengecekan dan memastikan semua *form* telah diisi oleh pengguna. Jika data *valid*, sistem akan menyimpan data ke *database* dan selesai. Jika tidak *valid*, sistem akan menampilkan notifikasi data gagal dikelola.



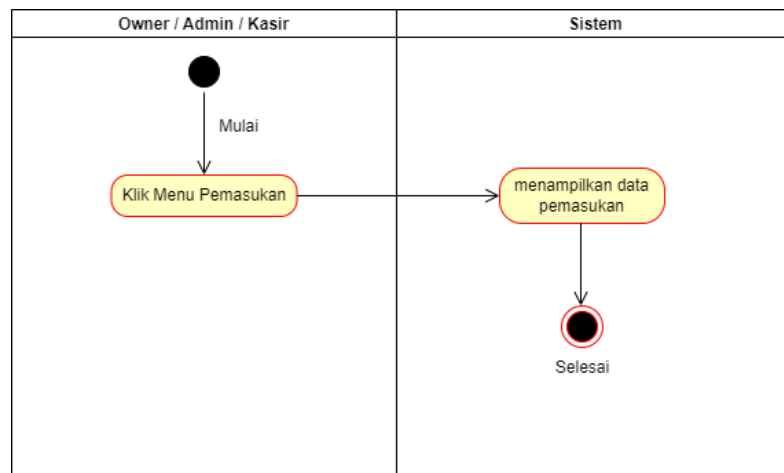
Gambar 3. 8 Activity Diagram Transaksi Masuk

Berdasarkan gambar 3.8 menjelaskan aktivitas mengelola transaksi masuk yang dimana dimulai dengan mengakses transaksi masuk, kelola transaksi masuk terdiri dari menambah, mengubah dan menghapus. Setelah melakukan kelola transaksi masuk, maka sistem melakukan validasi data yang dimana pada bagian validasi data dilakukan pengecekan dan memastikan semua *form* telah diisi oleh pengguna. Jika data *valid*, sistem akan menyimpan data ke *database* dan selesai. Jika tidak *valid*, sistem akan menampilkan notifikasi data gagal dikelola.



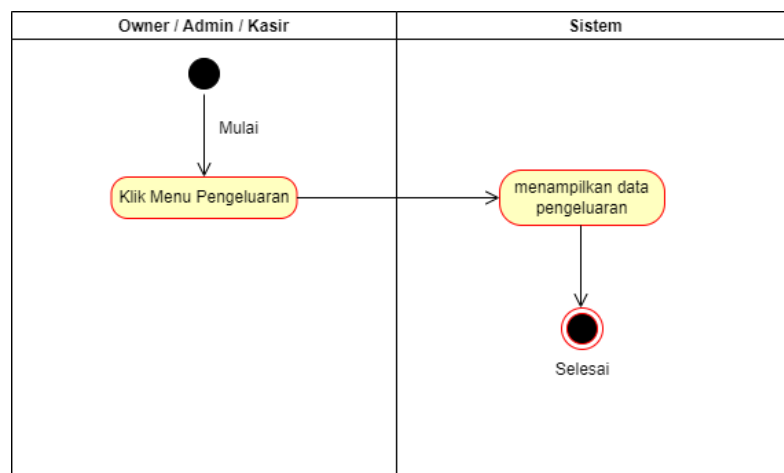
Gambar 3. 9 Activity Diagram Transaksi Keluar

Berdasarkan gambar 3.9 menjelaskan aktivitas mengelola transaksi keluar yang dimana dimulai dengan mengakses transaksi keluar, kelola transaksi keluar terdiri dari menambah, mengubah dan menghapus. Setelah melakukan kelola transaksi keluar, maka sistem melakukan validasi data yang dimana pada bagian validasi data dilakukan pengecekan dan memastikan semua *form* telah diisi oleh pengguna. Jika data *valid*, sistem akan menyimpan data ke *database* dan selesai. Jika tidak *valid*, sistem akan menampilkan notifikasi data gagal dikelola.



Gambar 3. 10 Activity Diagram Pemasukan

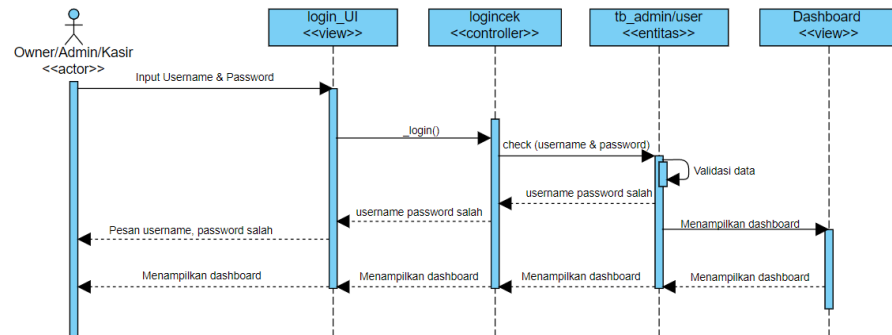
Berdasarkan gambar 3.10 dapat diambil penjelasan bahwa owner/admin/kasir melakukan klik pada menu pemesanan kemudian sistem dapat menampilkan data pemasukan.



Gambar 3. 11 Activity Diagram Pengeluaran

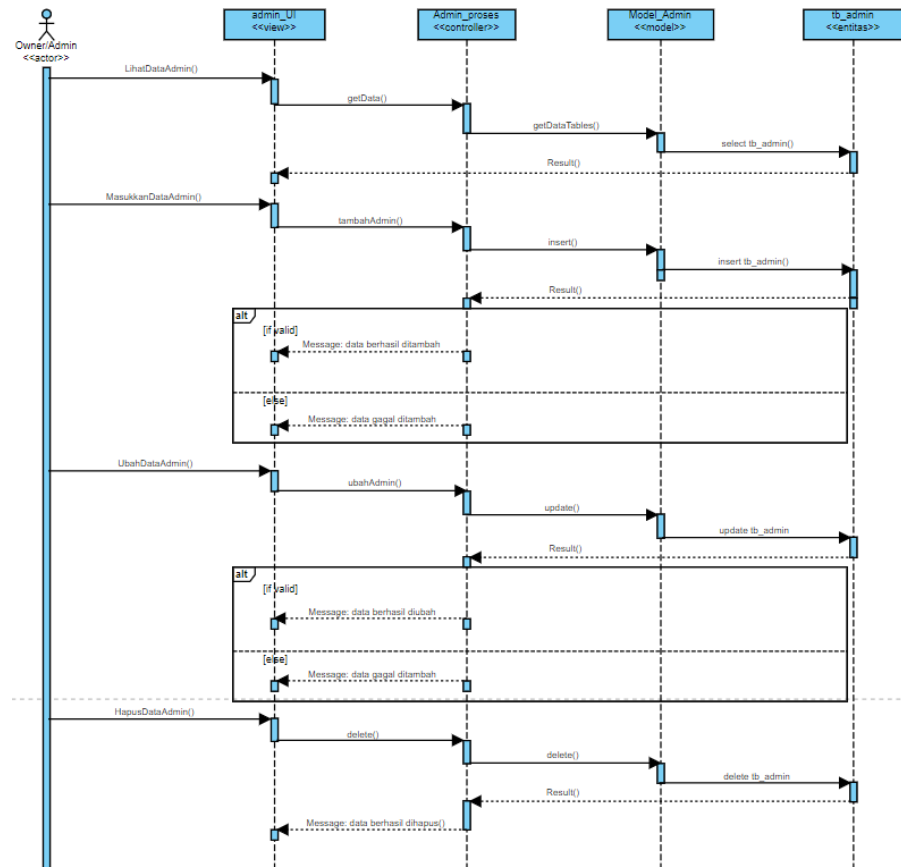
Berdasarkan gambar 3.11 dapat diambil penjelasan bahwa owner/admin/kasir melakukan klik pada menu pengeluaran kemudian sistem dapat menampilkan data pengeluaran.

g. Sequence Diagram



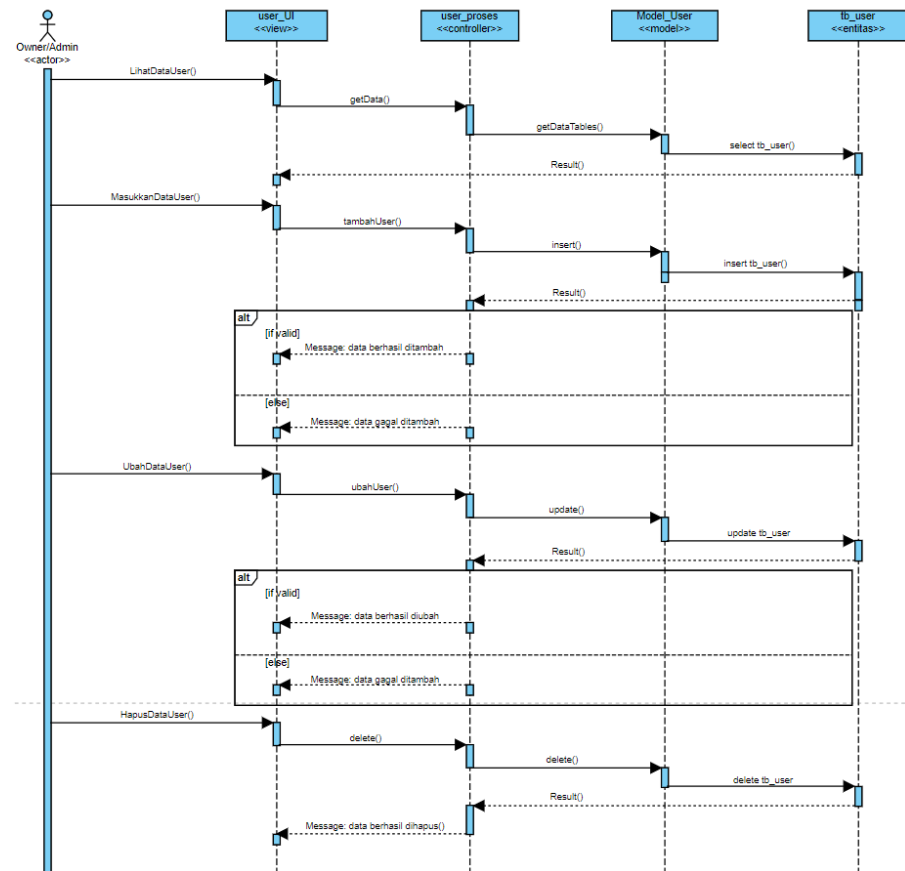
Gambar 3. 12 Sequence Diagram Login

Berdasarkan gambar 3.12 dapat diambil penjelasan bahwa aktor melakukan *input username* dan *password* pada *form login*, sistem akan melakukan memproses *username* dan *password* pada *database user* data untuk melakukan validasi. Jika, *username* dan *password* benar maka akan menampilkan halaman *dashboard*.



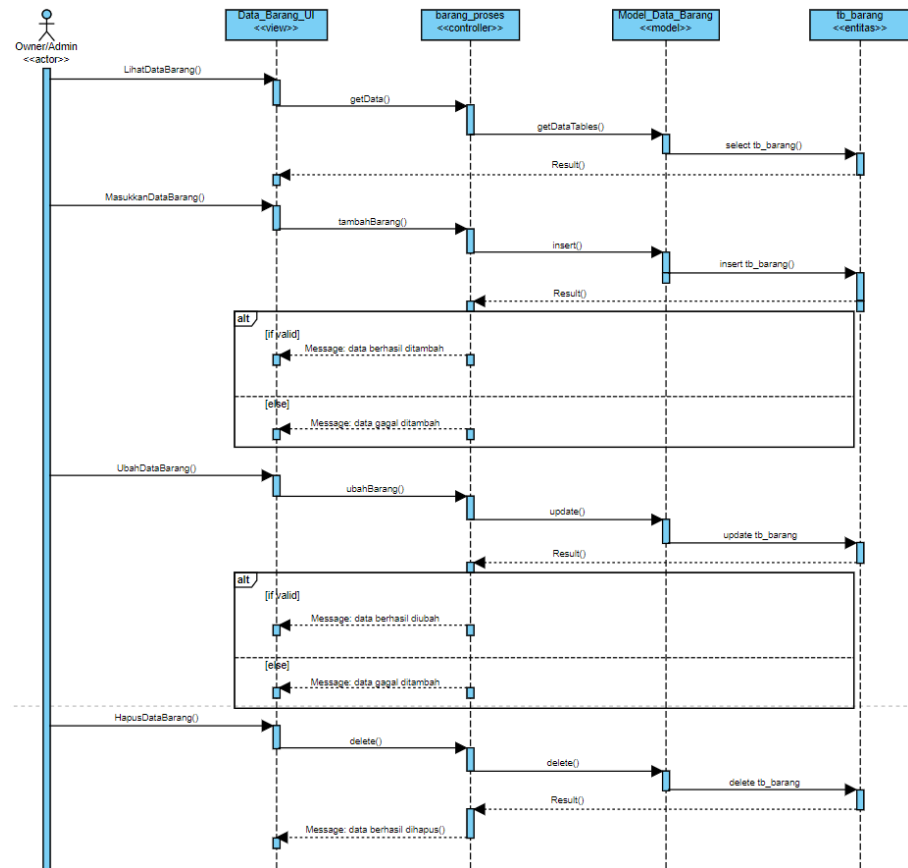
Gambar 3. 13 Sequence Diagram Kelola Data Admin

Gambar 3.13 menjelaskan mengelola data admin kemudian yang dimulai dengan mengakses menu data admin, kelola data admin yang terdiri dari tambah, ubah dan hapus. Setelah melakukan kelola data admin, sistem melakukan validasi data dimana jika data *valid* maka sistem akan menyimpan data admin ke *database*. Jika tidak *valid* maka sistem akan menampilkan notifikasi data gagal dikelola.



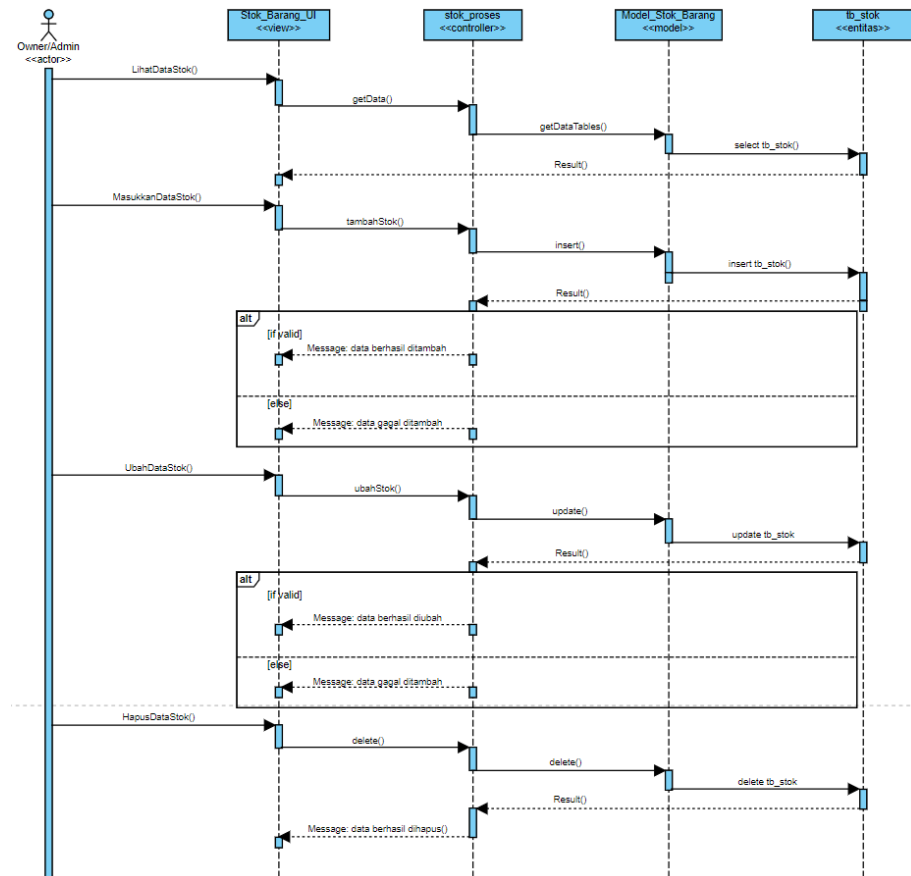
Gambar 3. 14 Sequence Diagram Kelola Data User

Gambar 3.14 menjelaskan mengelola data user kemudian yang dimulai dengan mengakses menu data user, kelola data user yang terdiri dari tambah, ubah dan hapus. Setelah melakukan kelola data user, sistem melakukan validasi data dimana jika data *valid* maka sistem akan menyimpan data user ke *database*. Jika tidak *valid* maka sistem akan menampilkan notifikasi data gagal dikelola.



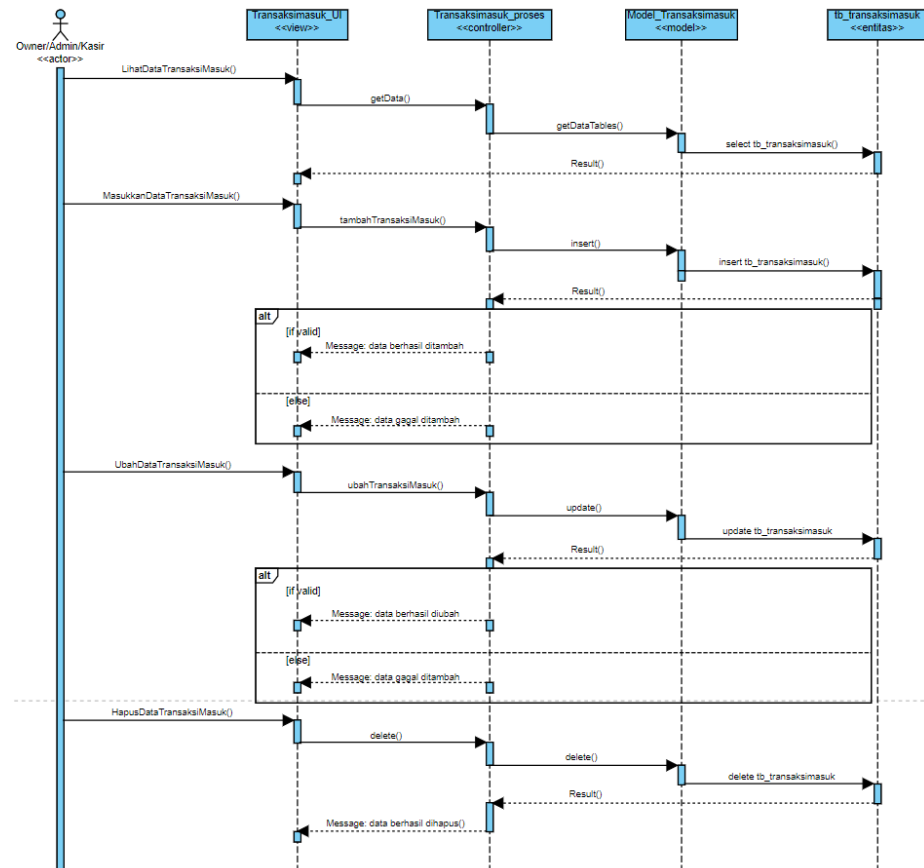
Gambar 3. 15 Sequence Diagram Data Barang

Gambar 3.15 menjelaskan bahwa aktor memilih menu data barang. Kelola data barang terdapat tambah, ubah dan hapus. Sistem melakukan validasi data, jika *valid* maka data barang tersimpan di *database* dan menampilkan notifikasi sukses kepada aktor. Jika tidak *valid*, sistem akan menampilkan notifikasi data barang gagal dikelola dan kembali ke halaman kelola data barang.



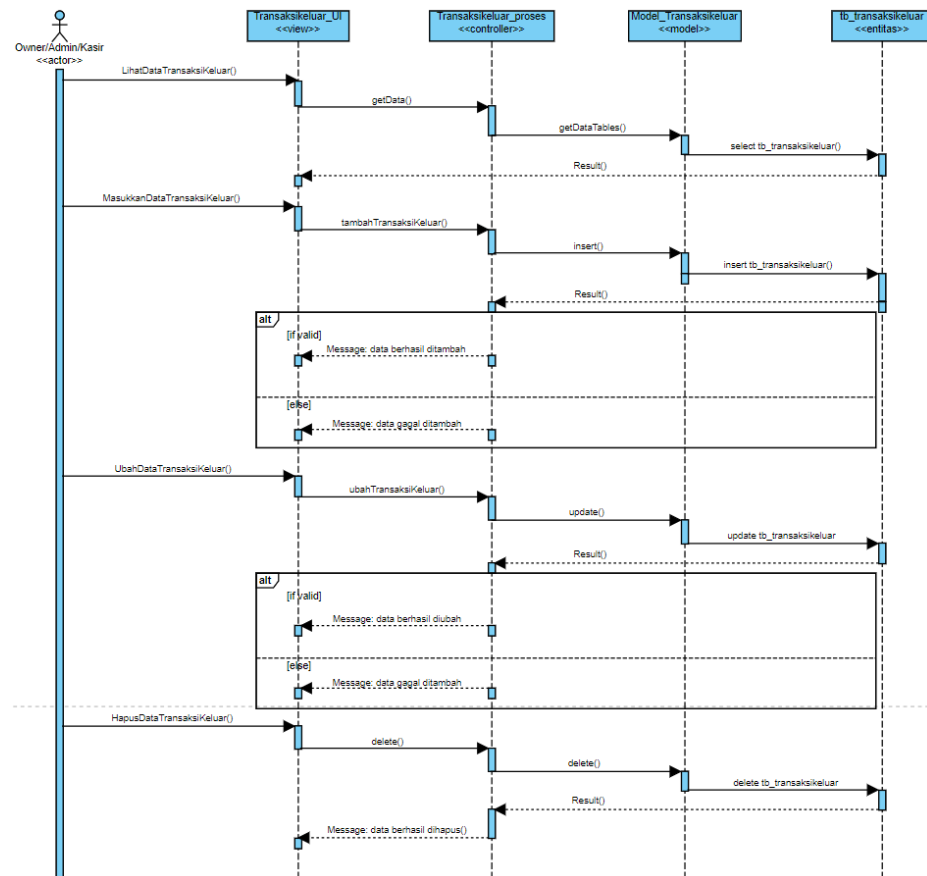
Gambar 3. 16 Sequence Diagram Stok Barang

Gambar 3.16 menjelaskan bahwa aktor memilih menu data stok barang. Kelola data stok barang terdapat tambah, ubah dan hapus. Sistem melakukan validasi data, jika *valid* maka stok barang tersimpan di *database* dan menampilkan notifikasi sukses kepada aktor. Jika tidak *valid*, sistem akan menampilkan notifikasi data stok barang gagal dikelola dan kembali ke halaman kelola stok barang.



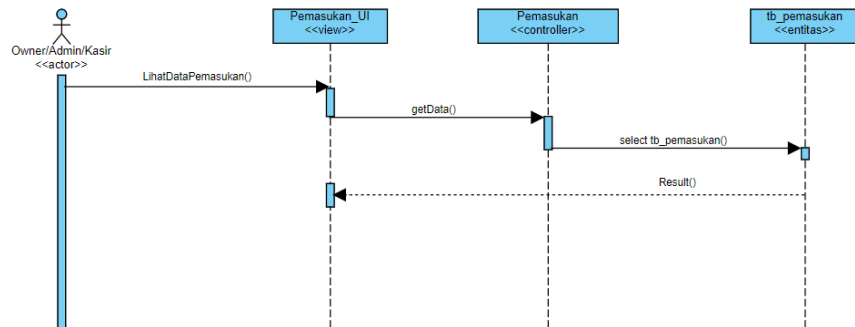
Gambar 3. 17 Sequence Diagram Transaksi Masuk

Gambar 3.17 menjelaskan bahwa aktor memilih menu data transaksi masuk. Kelola data transaksi masuk terdapat tambah, ubah dan hapus. Sistem melakukan validasi data, jika *valid* maka data transaksi masuk tersimpan di *database* dan menampilkan notifikasi sukses kepada aktor. Jika tidak *valid*, sistem akan menampilkan notifikasi data transaksi masuk gagal dikelola dan kembali ke halaman kelola transaksi masuk.



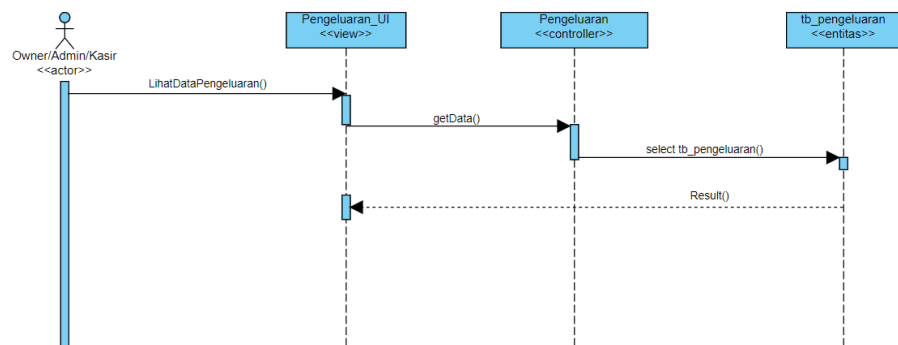
Gambar 3. 18 Sequence Diagram Transaksi Keluar

Berdasarkan gambar 3.18 menjelaskan bahwa aktor memilih menu data transaksi keluar. Kelola data transaksi keluar terdapat tambah, ubah dan hapus. Sistem melakukan validasi data, jika *valid* maka data transaksi keluar tersimpan di *database* dan menampilkan notifikasi sukses kepada aktor. Jika tidak *valid*, sistem akan menampilkan notifikasi data transaksi keluar gagal dikelola dan kembali ke halaman kelola transaksi keluar.



Gambar 3. 19 Sequence Diagram Pemasukan

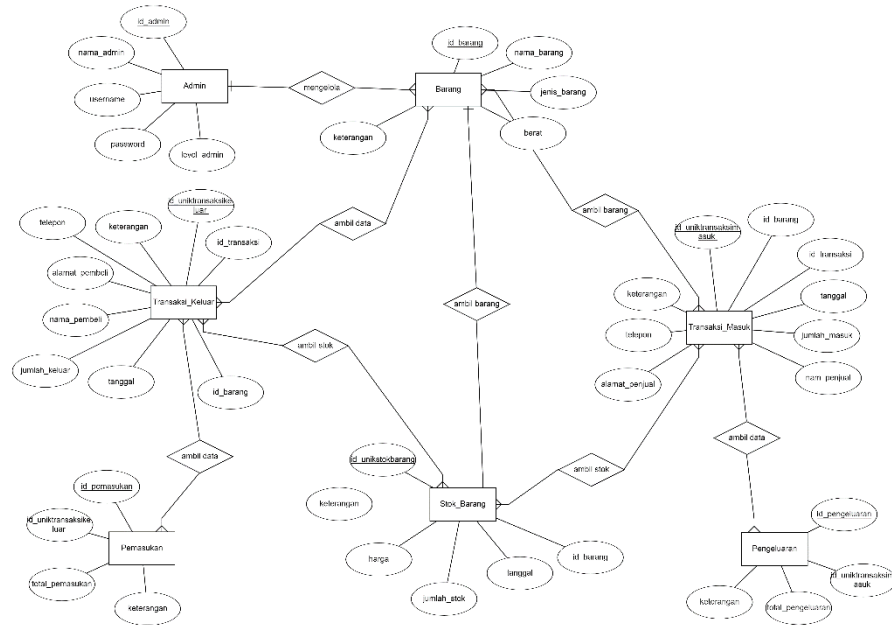
Berdasarkan gambar 3.19 menjelaskan untuk melihat pemasukan. Dimulai dengan mengakses menu pemasukan, kemudian sistem akan menampilkan data pemasukan.



Gambar 3. 20 Sequence Diagram Pengeluaran

Berdasarkan gambar 3.20 menjelaskan untuk melihat pengeluaran. Dimulai dengan mengakses menu pengeluaran, kemudian sistem akan menampilkan data pengeluaran.

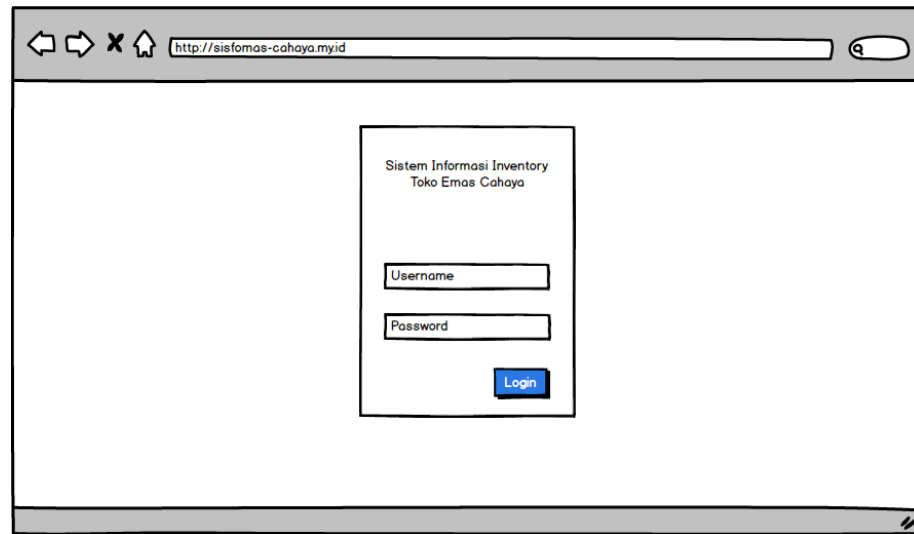
h. Entity Relationship Diagram (ERD)



Gambar 3. 21 Entity Relationship Diagram (ERD)

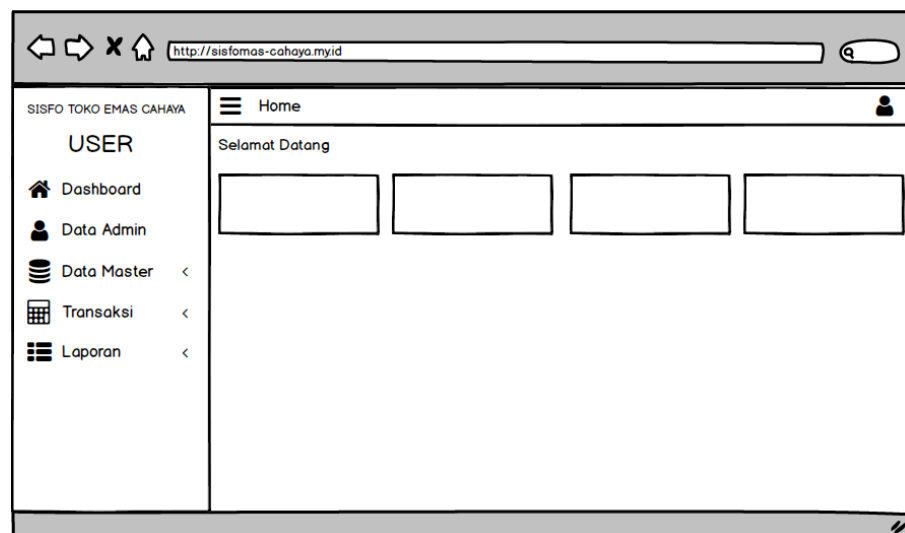
Berdasarkan gambar 3.21 pada *entity relationship diagram* (ERD) dijelaskan bahwa admin mengelola barang dengan relasi *one to many*, yang berarti admin dapat mengelola banyak barang. Kemudian stok barang berelasi dengan barang, untuk mengambil data barang dengan *id_barang*. Setelah itu, transaksi masuk dan keluar berelasi dengan barang dan stok barang dengan relasi *many to many*. Yang terakhir pemasukan dan pengeluaran berelasi dengan transaksi masuk dan keluar, karena untuk mengambil data transaksi.

i. *Mockup* Sistem



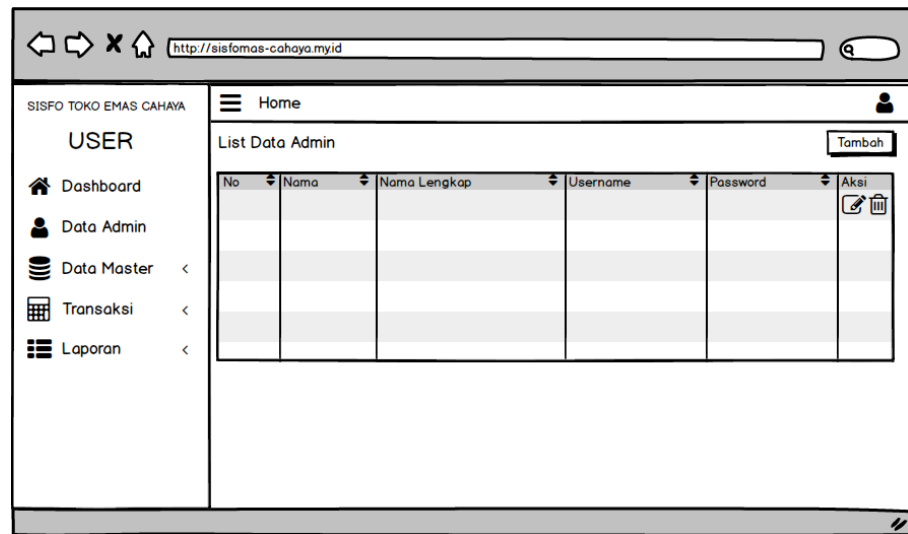
Gambar 3. 22 *Mockup* Form Login

Gambar 3.22 merupakan tampilan *mockup* dari Sistem Informasi *Inventory* Toko Emas Cahaya. Tampilan tersebut merupakan halaman awal dari *website*, yaitu halaman *login*.



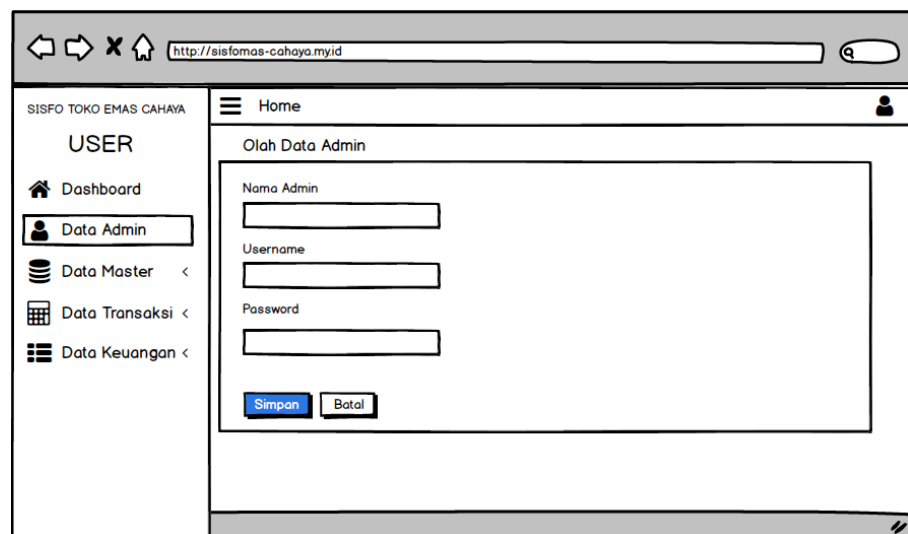
Gambar 3. 23 *Mockup* Menu Dashboard

Gambar 3.23 merupakan tampilan *mockup* menu *dashboard* yang merupakan tampilan awal setelah masuk ke dalam sistem.



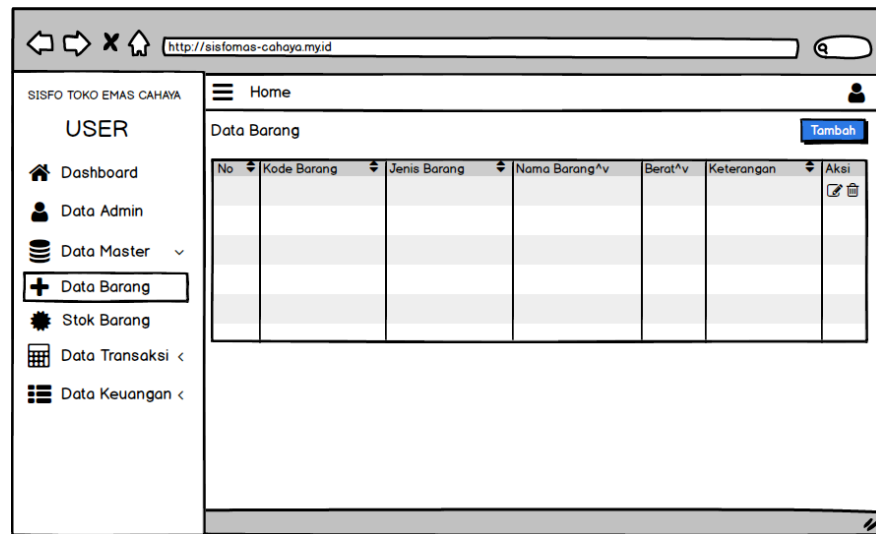
Gambar 3. 24 Mockup Menu Data Admin

Gambar 3.24 merupakan tampilan *mockup* menu data admin. Pada halaman olah data admin digunakan untuk mengelola data admin.



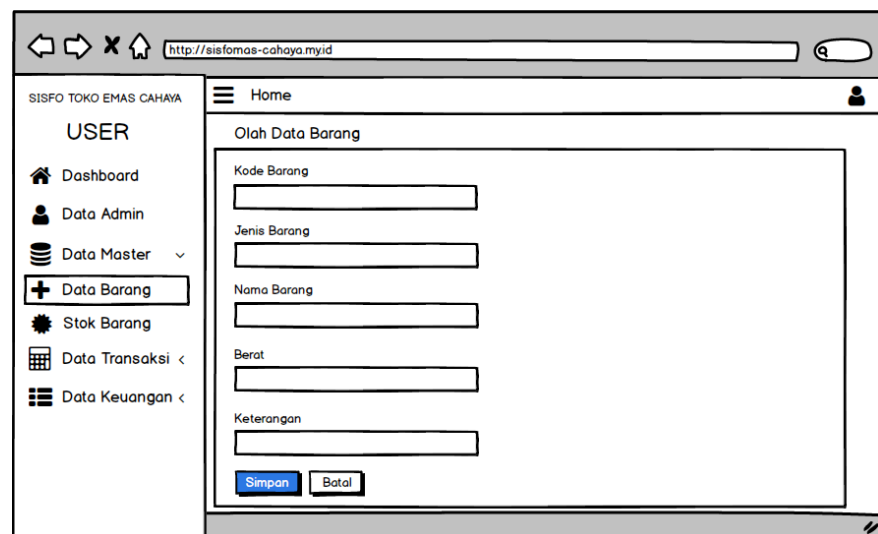
Gambar 3. 25 Mockup Menu Data Admin Ketika Tambah Data Admin

Gambar 3.25 merupakan tampilan *mockup* Menu data admin ketika tambah Data Admin. Pada halaman ini digunakan untuk menambahkan data Admin.



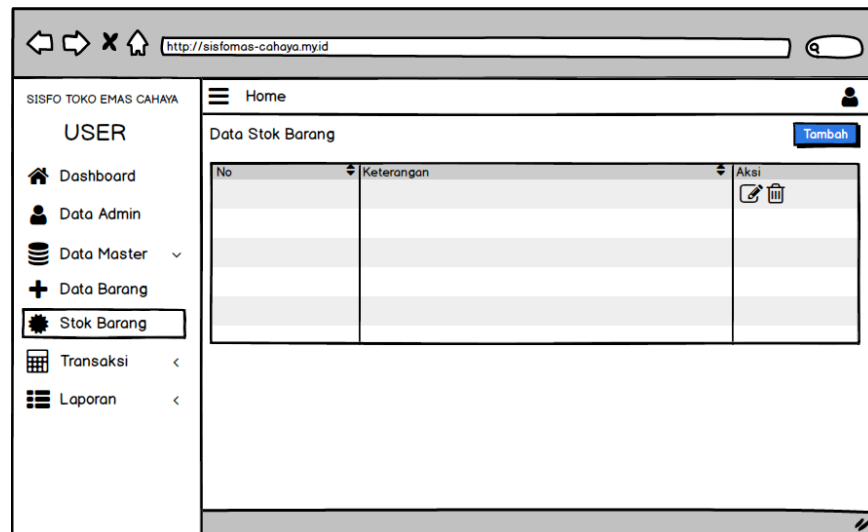
Gambar 3. 26 Mockup Data Barang

Gambar 3.26 merupakan tampilan *mockup* data barang. Pada halaman ini digunakan untuk mengelola Data Barang.



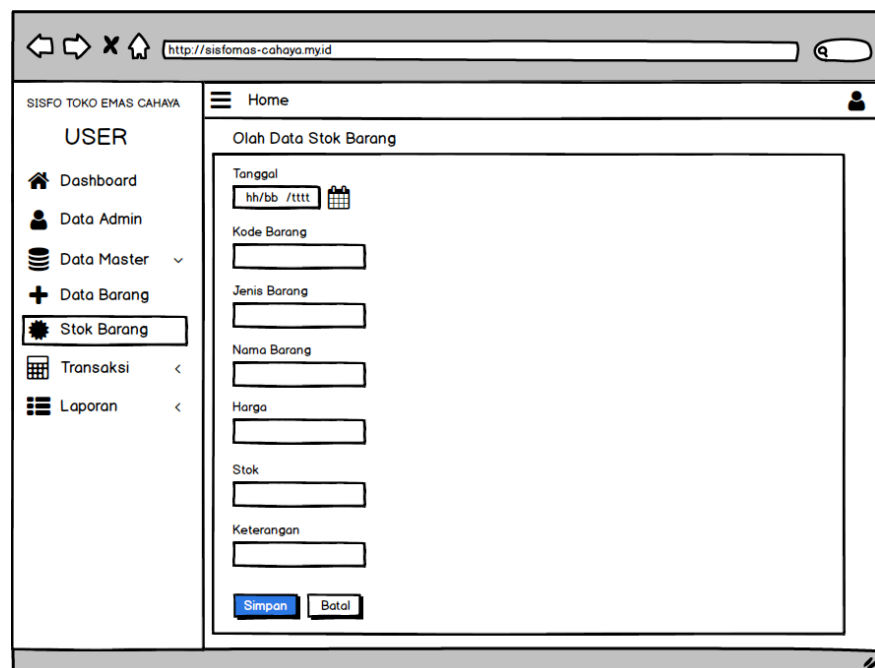
Gambar 3. 27 Mockup Menu Data Barang Ketika Menambahkan Barang

Gambar 3.27 merupakan tampilan *mockup* menu data barang ketika menambahkan barang. Pada halaman olah data barang digunakan untuk menambahkan barang.



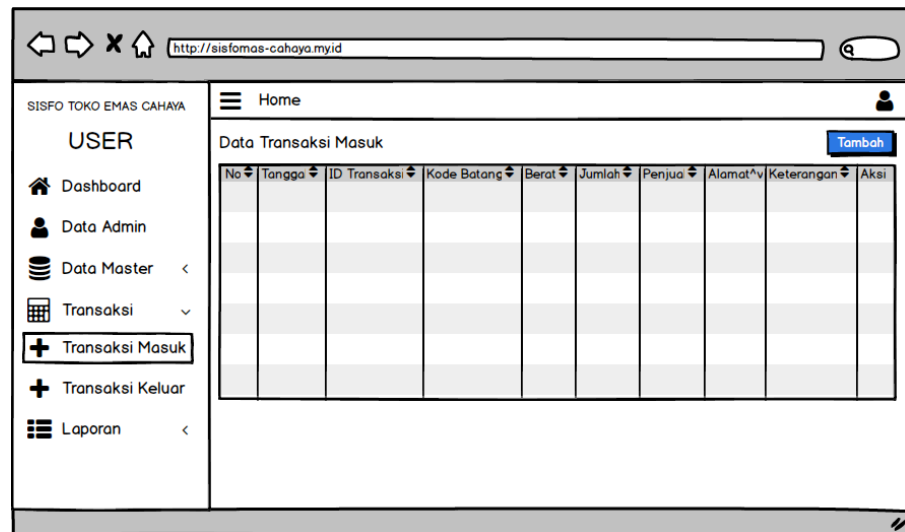
Gambar 3. 28 Mockup Data Stok Barang

Gambar 3.28 merupakan tampilan *mockup* data stok barang. Pada halaman ini digunakan untuk mengelola stok barang yang tersedia.



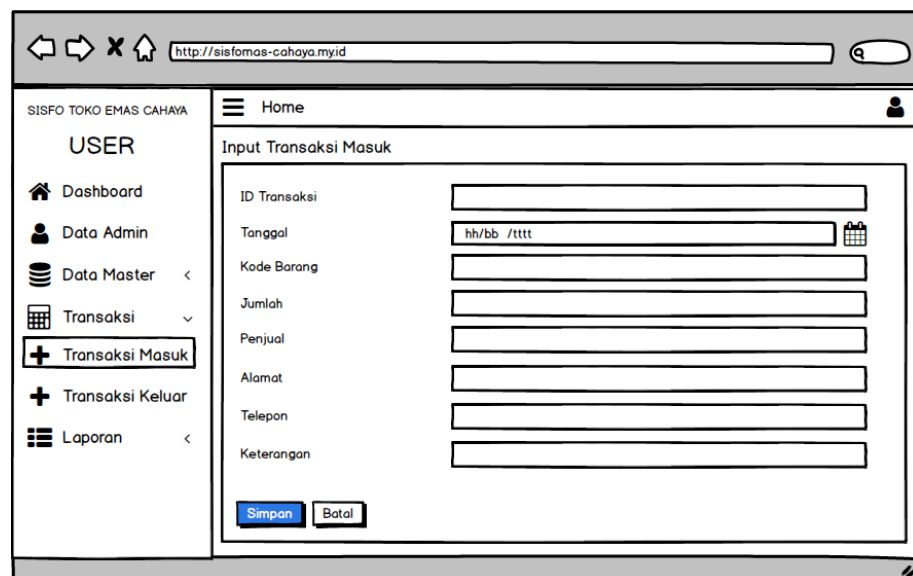
Gambar 3. 29 Mockup Data Olah Data Stok Barang

Gambar 3.29 merupakan tampilan *mockup* olah data stok barang. Pada halaman ini digunakan untuk menambah stok ke dalam sistem.



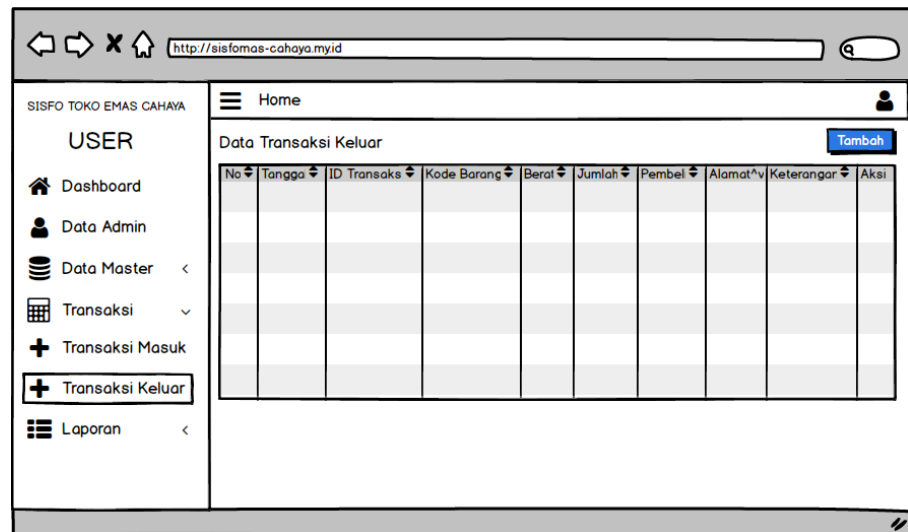
Gambar 3. 30 Mockup Menu Transaksi Masuk

Gambar 3.30 merupakan tampilan *mockup* menu transaksi masuk. Pada halaman ini digunakan untuk mengelola transaksi.



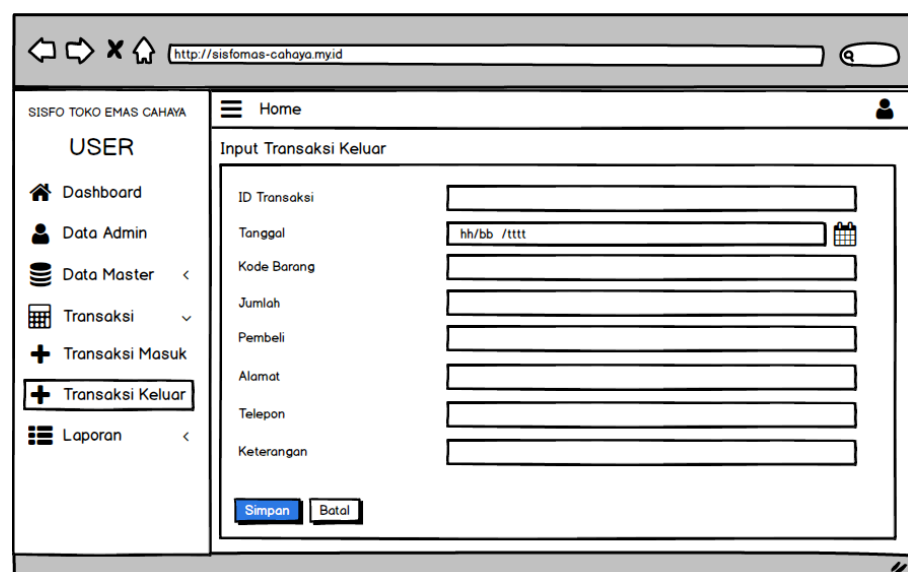
Gambar 3. 31 Mockup Input Transaksi Masuk

Gambar 3.31 merupakan tampilan *mockup input* transaksi masuk. Halaman ini digunakan untuk menginput transaksi ke dalam sistem.



Gambar 3. 32 Mockup Menu Transaksi Keluar

Gambar 3.32 merupakan tampilan *mockup* menu transaksi keluar. Pada halaman ini digunakan untuk mengelola transaksi.



Gambar 3. 33 mockup Input Transaksi Keluar

Gambar 3.33 merupakan tampilan *mockup* Input Transaksi Keluar. Halaman ini digunakan untuk melakukan *input* transaksi ke dalam sistem.

3. Pengkodean (*Coding*)

Tahap saat melakukan pengkodean pembangunan sistem informasi *inventory* sesuai dengan rancangan yang sudah disepakati dengan pelanggan berdasarkan pemodelan yang telah dibuat sebelumnya.

4. Pengujian (*Testing*)

Pada tahap pengujian pada sistem, menggunakan metode Black Box Testing untuk mengetahui apakah sistem yang dibangun sudah berjalan dengan baik.

3.3.5 Melakukan *System Usability Scale (SUS)*

Pada tahap ini, melakukan *survey* terhadap *user* atau pengguna yang akan menggunakan sistem ini yaitu terdiri dari empat orang karyawan toko Emas Cahaya Kesugihan yang berguna untuk mengetahui apakah sistem informasi *inventory* sudah berjalan sesuai dengan kebutuhan.