

## BAB IV HASIL DAN ANALISIS PENELITIAN

### 4.1 Analisis Sistem Berjalan

Analisis perancangan sistem adalah proses memecah sistem menjadi komponen-komponen dan memeriksa bagaimana komponen-komponen tersebut dapat bekerja dan berinteraksi satu sama lain. Analisis perancangan sistem sendiri didasarkan pada hasil analisis masalah dan kebutuhan. Dalam penelitian ini, penulis membuat rancangan berupa desain modul berdasarkan hasil analisis masalah dan kebutuhan yang disebutkan pada Bab III. Pada penelitian ini penulis menyarankan untuk membuat modul pemesanan pengiriman barang dan verifikasi pengiriman barang sistem *cash on delivery* pada aplikasi *e-Logistic* berbasis web. Perancangan aplikasi ini bertujuan untuk

- memudahkan pemilik toko *online*, pengguna jasa pengiriman barang serta konsumen dalam membuat pemesanan pengiriman barang secara *online* serta melakukan *tracking* order dan juga verifikasi ketersediaan pengguna dalam penerimaan barang dengan sistem *cash on delivery*

Pemesanan pengiriman barang secara *online* pada aplikasi yang dirancang bertujuan untuk melakukan digitalisasi dalam pengiriman barang, sehingga jasa pengiriman barang dapat memiliki data yang sesuai dengan data yang diberikan oleh pengirim barang. Selain itu, dengan pemberlakuan digitalisasi tersebut *customer* dapat dipermudah dalam pengiriman barang dengan mengisi data melalui formulir pengajuan pengiriman barang melalui aplikasi secara digital tanpa menggunakan formulir berupa kertas.

Selanjutnya dengan adanya *tracking* order secara *real time* dapat mempermudah penerima barang untuk mengecek dimana barang mereka berada dan juga estimasi waktu untuk barang tersebut sampai ke pemilik barang. Hal tersebut mampu mempermudah pemberian informasi yang tepat ke pemilik barang. Selain itu, dengan adanya verifikasi ketersediaan pengguna dalam penerimaan barang dengan sistem *cash on delivery* tersebut, user dapat menentukan jika barang mereka sudah di kantor cabang terdekat apakah mereka siap untuk menerima barang tersebut atau tidak. Hal tersebut dapat

mempermudah pengiriman barang dan memberikan keuntungan bagi pemilik barang, kurir serta jasa pengiriman barang tersebut. Pemilik barang akan dapat menerima barang saat pemilik barang sedang berada di rumah, sehingga pemilik dapat menyiapkan nominal yang harus di bayarkan dan pemilik barang dapat mengetahui kapan tepatnya barang tersebut dikirim ke rumah pemilik barang, selain itu juga pemilik barang dapat melakukan *request* ataupun *reschedule* untuk pengiriman barang ketika pemilik barang bersedia menerima barangnya. Kurir akan semakin di mudahkan dengan metode tersebut karena kurir dapat mengantarkan barang tersebut langsung ke pemilik ketika mereka ada di rumah, sehingga kurir tidak perlu pergi mengantarkan barang tersebut berulang kali ketika pemilik barang tersebut sedang tidak di rumah. Karena hal tersebut, maka jasa pengiriman barang tidak perlu mengeluarkan *cost* berlebih dalam pengiriman barang tersebut, karena pihak pemilik barang serta cabang

- terdekad sudah setuju bahwa pemilik barang sudah siap menerima barangnya.

Hal tersebut dapat mengurangi *cost* berlebihnya ketika kurir harus pergi berkali-kali ke lokasi pemilik barang ketika pemilik barang tidak siap menerima barangnya tersebut.

## 4.2 Perancangan Diagram Sistem Usulan

Penulis menggunakan OOAD sebagai metode pengembangan sistemnya. Dalam menggambarkan proses analisis yang akan penulis lakukan, penulis menggunakan bahasa *Unified Modeling Language* (UML), yang terdiri dari diagram – diagram yang akan menggambarkan proses sistem berjalan.

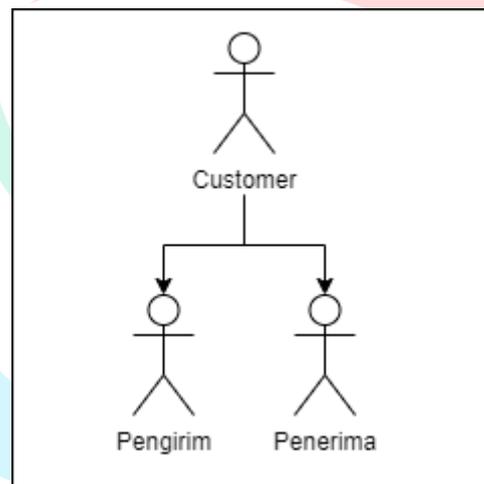
Selain itu, metode OOAD sangat cocok digunakan dalam melakukan dalam pengembangan aplikasi dalam penelitian ini karena:

1. Analis dan *programmer* tidak dibatasi oleh kendala implementasi sistem, sehingga mereka dapat merumuskan proyek dan memvalidasinya menggunakan lingkungan *runtime* yang berbeda.
2. OOAD lebih mudah digunakan dalam melakukan pengembangan sistem.
3. Lebih mudah menemukan hal-hal untuk diubah.
4. Dapat memberikan informasi yang jelas tentang konteks *system*.

5. Dapat mengurangi biaya *maintenance*.

#### 4.2.1 Use Case Diagram

Use case diagram merupakan bentuk utama dari sistem persyaratan perangkat lunak untuk sistem baru yang ingin dibuat. Use case dapat menentukan perilaku yang diharapkan (apa), dan bukan metode yang tepat untuk mewujudkannya (bagaimana). Use case setelah ditentukan dapat dilambangkan dengan representasi tekstual dan visual. Dalam sistem yang dikembangkan oleh penulis, terdapat 3 *roles* yang ada, yaitu Admin, pengirim dan penerima. *Costumer* merupakan generalisasi aktor dari Pengirim dan Penerima. Berikut ini merupakan Use Case Diagram yang penulis usulkan untuk pada pembuatan aplikasi *e-Logistic* berbasis web:



Gambar 4.1 Generalisasi User

Selanjutnya merupakan diagram Use Case yang menggambarkan kumpulan aktivitas yang dapat dilakukan oleh aktor yang terdapat pada Gambar 4.2. Berikut merupakan penjelasan dari setiap aktivitas tersebut:

##### 1. Login

Seluruh aktor diharuskan untuk *login* terlebih dahulu untuk menjalankan aplikasi. Aktor dapat *login* dengan memasukkan nama pengguna dan *password* mereka di *form login* yang disediakan.

## 2. Mengirim Barang

User dapat mengirim barang ketika mereka sudah melakukan login. Lalu dalam mengirim barang, user hanya perlu memasukan informasi dari pengiriman barang tersebut.

## 3. Kelola Pengiriman Barang

Admin dapat melakukan kelola pengiriman barang jika terdapat data yang salah atau pengiriman data yang invalid dan mengkonfirmasi pemesanan barang.

## 4. *Tracking* Barang

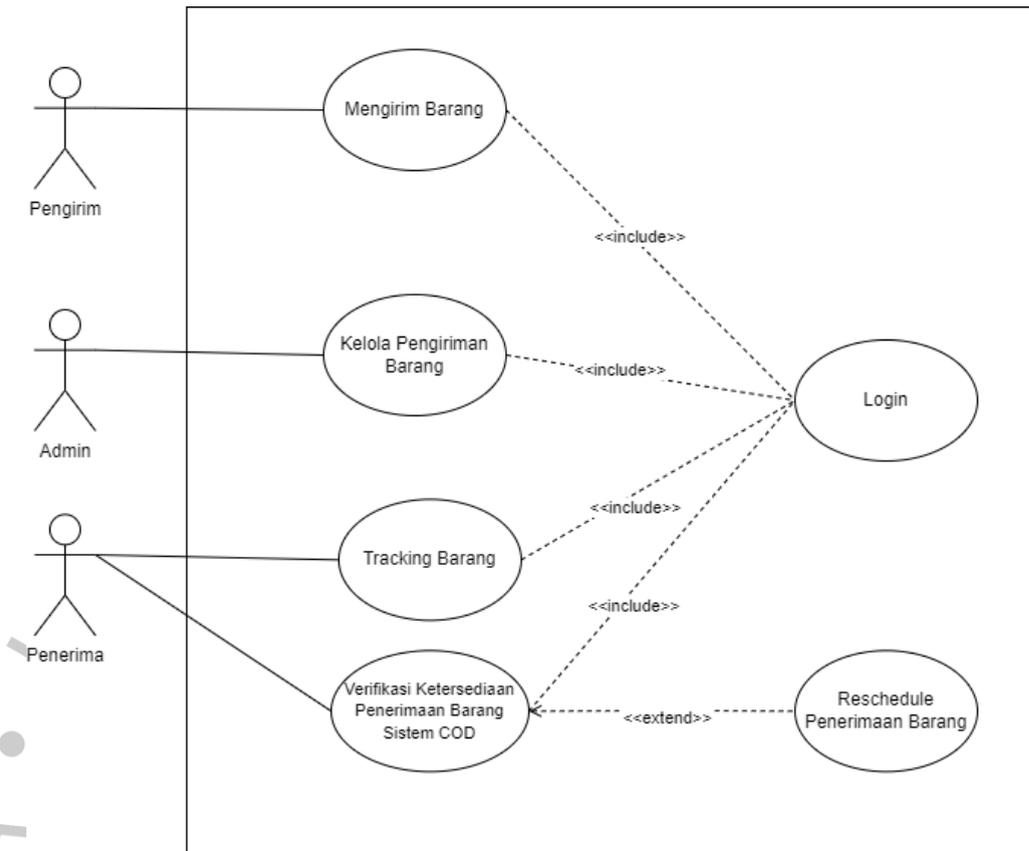
Penerima dapat melakukan *tracking* barang setelah *login* di aplikasi tersebut. Penerima dapat melihat lokasi barang secara *real time* di dalam aplikasi *e-logistic*

## 5. Verifikasi Ketersediaan Penerimaan Barang Sistem COD

Dalam penerimaan barang, user dapat melakukan konfirmasi pengiriman barang ketika user tersebut di rumah.

## 6. *Reschedule* Pengiriman Barang

Penerima dapat melakukan *reschedule* penerimaan barang apabila penerima tidak berada di rumah. Sehingga, barang akan di antarkan ke penerima ketika dapat menerima barang.



**Gambar 4.2** Use Case Aplikasi e-logistic

#### 4.2.2 Spesifikasi Use Case

Spesifikasi *use case* adalah deskripsi alur atau langkah dari setiap *use case* yang dijelaskan dalam skenario. Berikut adalah spesifikasi *use case* yang dikembangkan oleh penulis dalam aplikasi e-logistic yang dapat dilihat pada **Tabel 4.1** sampai dengan **Tabel 4.6**.

**Tabel 4.1** Spesifikasi *Use Case Login*

<b>Name</b>	Login	
<b>Actor</b>	Pengirim, Admin, Penerima	
<b>Trigger</b>	Aktor ingin mengakses website	
<b>Pre-Condition</b>	Aktor sudah memiliki akun	
<b>Post-Condition</b>	Menampilkan halaman utama	
<b>Success Scenario</b>	<b>Aktor</b>	<b>Sistem</b>
	1. Melakukan Login dengan memasukkan Username dan Password	
		2. Melakukan validasi Username dan Password

		3. Menampilkan Dashboard
<b>Alternatives flows</b>	Login gagal jika username atau password salah. Dan sistem akan menampilkan pesan Username atau Password Salah”	

**Tabel 4.2** Spesifikasi *Use Mengirim Barang*

<b>Name</b>	Mengirim Barang	
<b>Actor</b>	Pengirim	
<b>Trigger</b>	Pengirim ingin mengirimkan barang kepada penerima	
<b>Pre-Condition</b>	Pengirim datang ke kantor cabang	
<b>Post-Condition</b>	Barang diinput oleh admin	
<b>Success Scenario</b>	Aktor	Sistem
	1. < Include : Login >	
	2. Pengirim membuka halaman form pengiriman barang	
		3. Sistem menampilkan halaman form pengiriman barang
<b>Alternatives flows</b>	-	

**Tabel 4.3** Spesifikasi *Use Case Kelola Pengiriman Barang*

<b>Name</b>	Kelola Pengiriman Barang	
<b>Actor</b>	Admin	
<b>Trigger</b>	Pengirim ingin mengirim barang	
<b>Pre-Condition</b>	Admin Login ke website	
<b>Post-Condition</b>	Data pengiriman barang berhasil diinput	
<b>Success Scenario</b>	Aktor	Sistem
	1. < Include : Login >	
	2. Menerima data pengiriman barang.	
	3. Memproses data dan memberikan estimasi biaya pengiriman, nomor resi, dan invoice.	
		4. Sistem menampilkan data ke pengirim.
<b>Alternatives flows</b>	-	

**Tabel 4.4** Spesifikasi *Use Case Tracking Barang*

<b>Name</b>	Tracking Barang	
<b>Actor</b>	Penerima	
<b>Trigger</b>	Aktor ingin mengetahui dimana barangnya	
<b>Pre-Condition</b>	Aktor sudah melakukan login	
<b>Post-Condition</b>	Aktor berhasil mengetahui dimana barangnya	
<b>Success Scenario</b>	Aktor	Sistem
	1. < Include : Login >	
	2. Memasukan nomor resi	
		3. Sistem menampilkan lokasi serta pop up maps.
<b>Alternatives flows</b>		

**Tabel 4.5** Spesifikasi *Use Case* Verifikasi Ketersediaan Penerimaan Barang Sistem COD

<b>Name</b>	Verifikasi Ketersediaan Penerimaan Barang Sistem COD	
<b>Actor</b>	Penerima	
<b>Trigger</b>	Aktor ingin memvalidasi bahwa aktor dapat menerima barang	
<b>Pre-Condition</b>	Aktor sudah melakukan login	
<b>Post-Condition</b>	Aktor berhasil memvalidasi bahwa aktor dapat menerima barang	
<b>Success Scenario</b>	<b>Aktor</b>	<b>Sistem</b>
	1. < Include : Login >	
	3. Aktor memvalidasi bahwa aktor dapat menerima barang.	2. Sistem menampilkan barang yang siap dikirim ke penerima.
<b>Alternatives flows</b>		

**Tabel 4.6** Spesifikasi *Use Case* Reschedule Penerimaan Barang

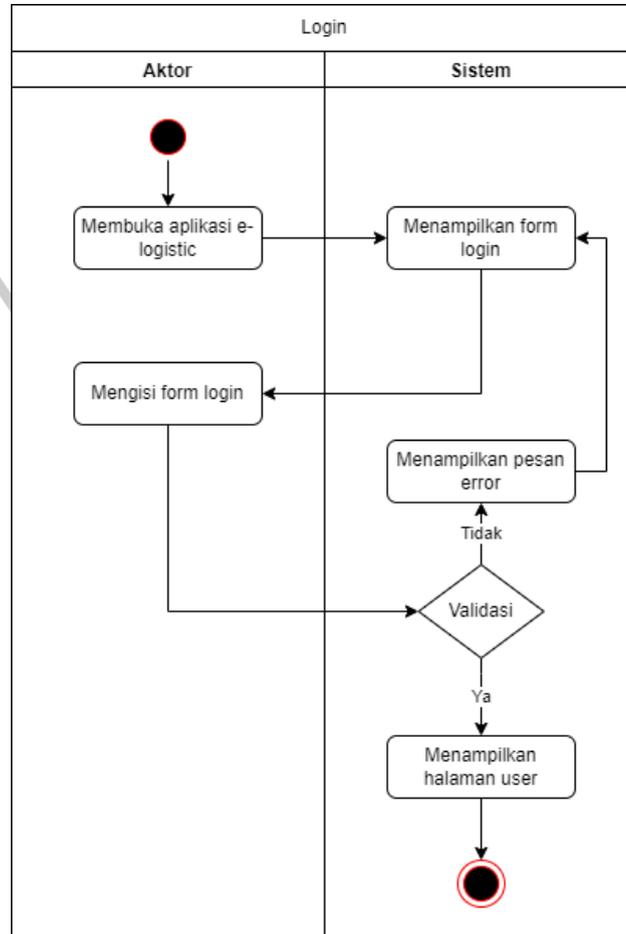
<b>Name</b>	Reschedule Penerimaan Barang	
<b>Actor</b>	Penerima	
<b>Trigger</b>	Aktor ingin mengubah waktu penerimaan barang	
<b>Pre-Condition</b>	Aktor sudah melakukan login	
<b>Post-Condition</b>	Aktor berhasil mengubah waktu penerimaan barang	
<b>Success Scenario</b>	<b>Aktor</b>	<b>Sistem</b>
	1. < Include : Login >	
	3. Aktor mengubah waktu penerimaan barang	2. Sistem menampilkan barang yang siap dikirim ke penerima.
<b>Alternatives flows</b>		

### 4.2.3 Activity Diagram

*Activity* Diagram adalah diagram perilaku penting lainnya dalam diagram UML untuk menggambarkan aspek dinamis dari sistem. *Activity* Diagram pada dasarnya adalah versi lanjutan dari *flow chart* diagram yang memodelkan *flow* dari satu aktivitas ke aktivitas lainnya.

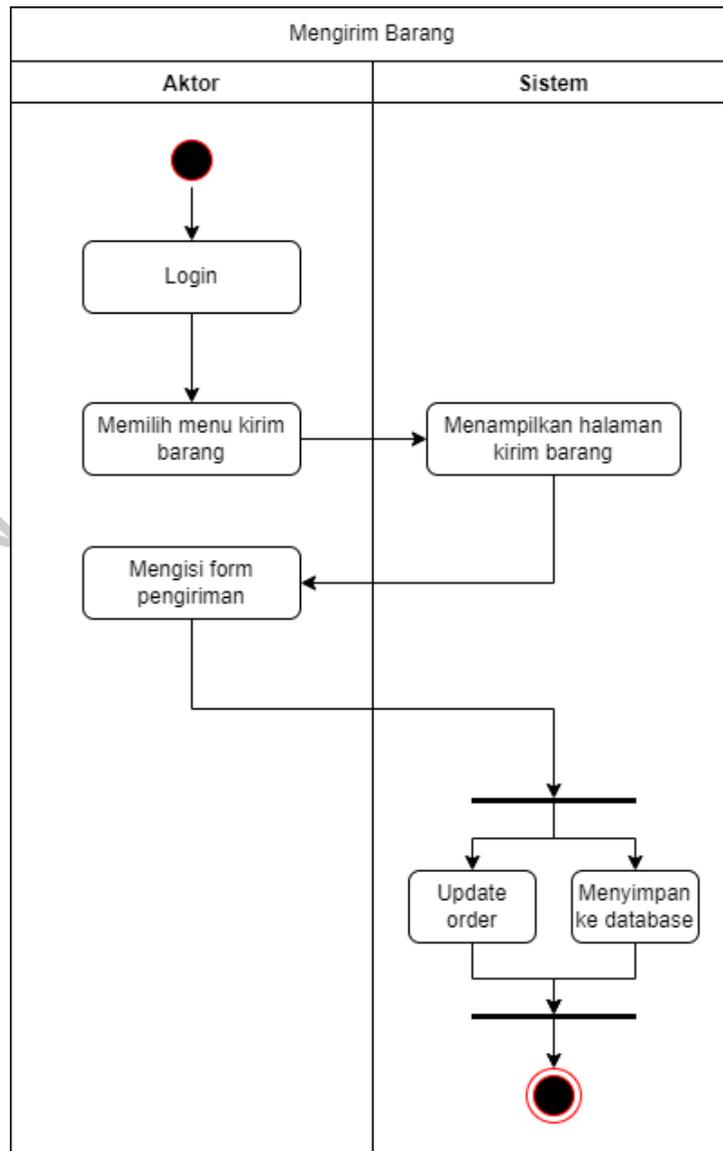
Proses *Login* dimulai ketika aktor membuka aplikasi dan sistem menampilkan *form Login*. Aktor diminta untuk memasukkan *Username* dan *password* akun yang sudah terdaftar dan sistem akan memvalidasi. Jika validasi gagal sistem akan memberikan pesan eror dan menampilkan *form Login*, dan jika validasi berhasil maka sistem akan menampilkan *dashboard* bagi admin, dan jika customer akan

menampilkan halaman user. *Activity Diagram Login* dapat dilihat pada **Gambar 4.3**.



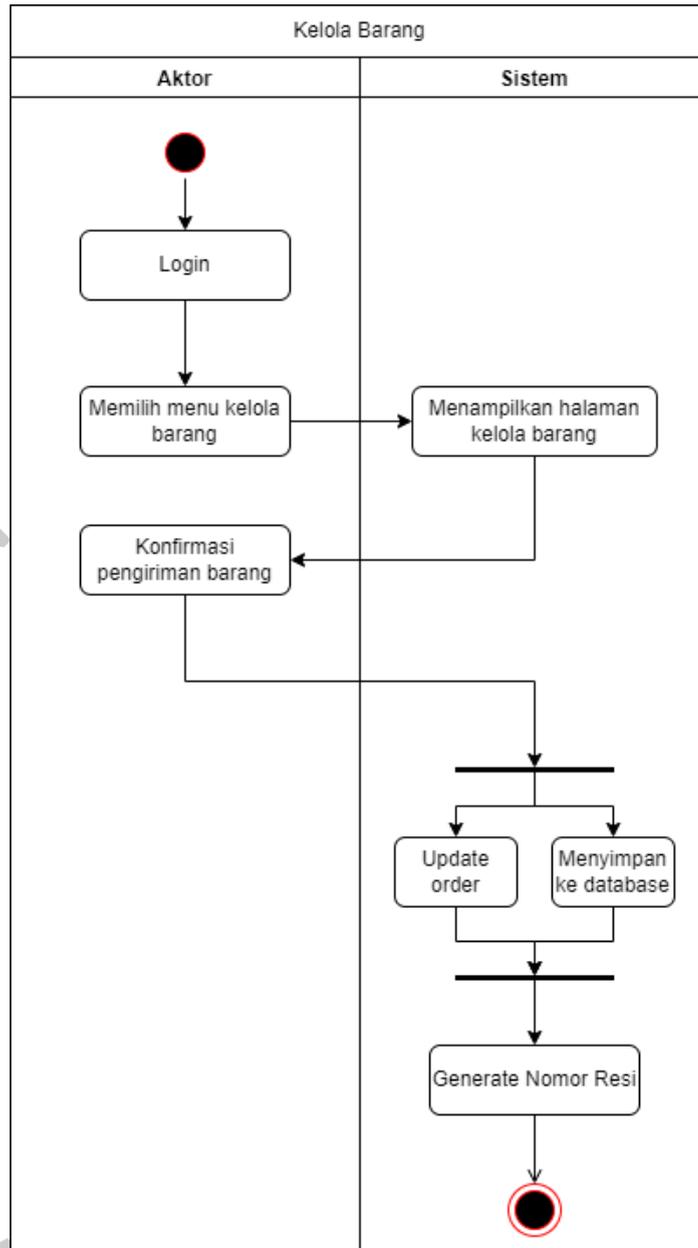
**Gambar 4.3** Activity Diagram Login

Proses Mengirim Barang dimulai ketika aktor telah melakukan *Login*. Setelah sukses melakukan *login*, Sistem akan secara otomatis menampilkan halaman user. Pada halaman tersebut user dapat memilih menu kirim barang, dan sistem akan memunculkan formulir pengiriman barang. *Customer* akan mengisi data berupa nama pengirim, email, lokasi pengiriman, tipe pengiriman, dan juga berat barang. Setelah *customer* mengisi *form* tersebut, data akan di simpan di dalam *database*. *Activity Diagram* mengirim barang dapat dilihat pada **Gambar 4.4**.



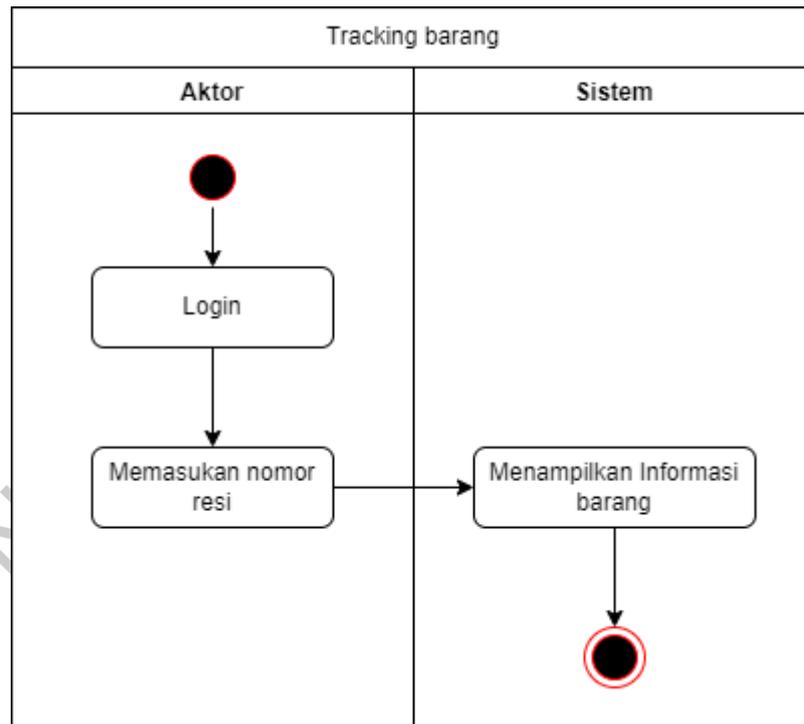
**Gambar 4.4** Activity Diagram Mengirim Barang

Proses Kelola Barang dilakukan oleh admin ketika admin telah melakukan *Login*. Setelah admin sukses melakukan *login*, sistem akan menampilkan *dashboard* kelola barang, dalam *dashboard* tersebut admin dapat melakukan konfirmasi pengiriman barang yang telah diisi oleh *customer*. Setelah user admin mengkonfirmasi pengiriman barang tersebut, sistem akan memberikan nomor resi yang nantinya akan diberikan kepada *customer* yang mengirim barang. Activity Diagram kelola barang dapat dilihat pada **Gambar 4.5**.



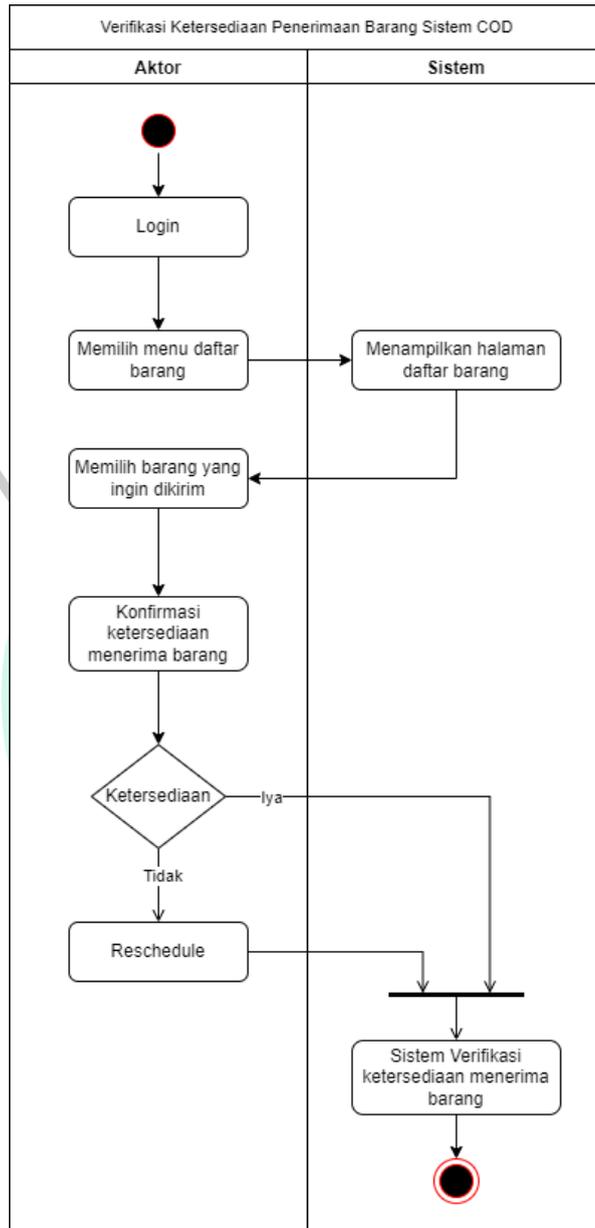
**Gambar 4.5** Activity Diagram Kelola Barang

Proses *Tracking* Barang dilakukan oleh *Customer*. Sebelum melakukan *tracking* barang, *customer* perlu melakukan *login* terlebih dahulu. Setelah user *login*, user dapat mengisi *field* untuk mengisi nomor resi, jika data yang dimasukkan benar maka informasi barang akan muncul, selain itu juga *customer* dapat melihat informasi barang dengan berupa titik koordinat *maps*. Activity Diagram *tracking* barang dapat dilihat pada **Gambar 4.6**.



**Gambar 4.6** Activity Diagram Tracking Barang

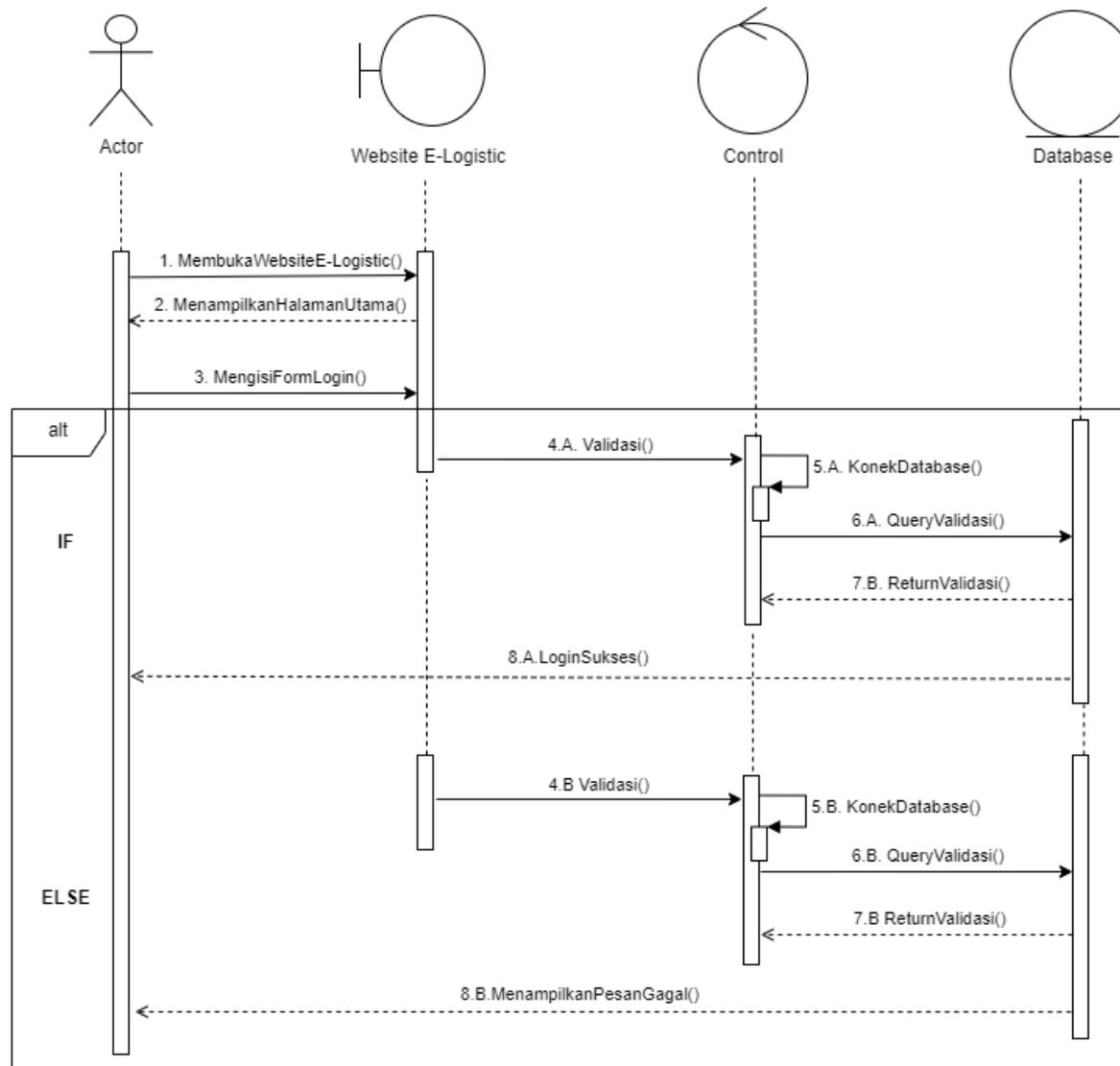
Proses Verifikasi Ketersediaan Penerimaan Barang Sistem COD dilakukan oleh *Customer* yang telah melakukan *login*. User dapat memverifikasi ketersediaan tersebut ketika barang yang dikirim sudah sampai di cabang terdekat di rumah dan barang tersebut siap untuk dikirimkan ke *customer*. *Customer* dapat melakukan *reschedule* jika tidak dapat menerima barang. Activity Diagram *tracking* barang dapat dilihat pada **Gambar 4.8**.



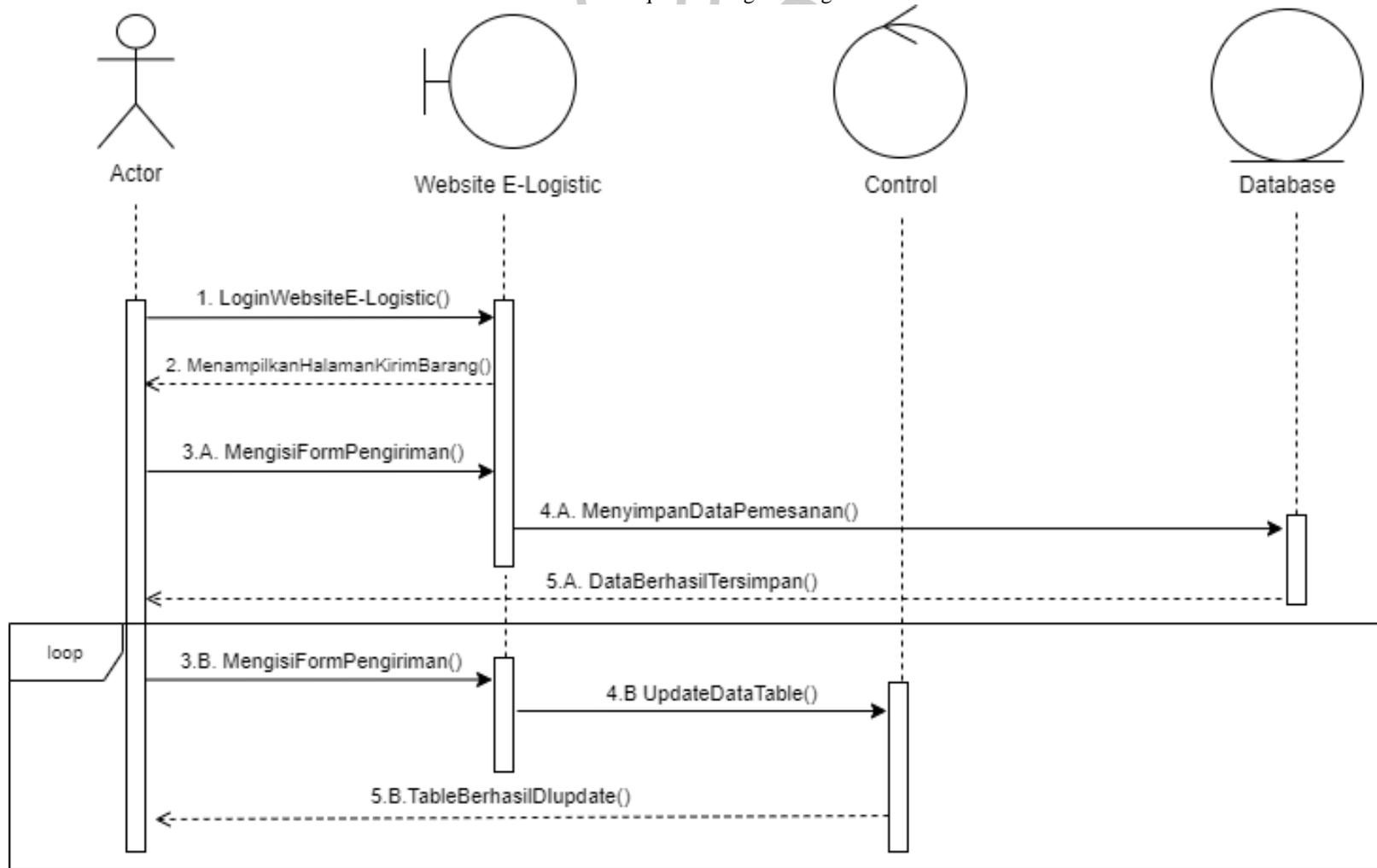
Gambar 4.7 Verifikasi Ketersediaan Penerimaan Barang Sistem COD

#### 4.2.4 Sequence Diagram

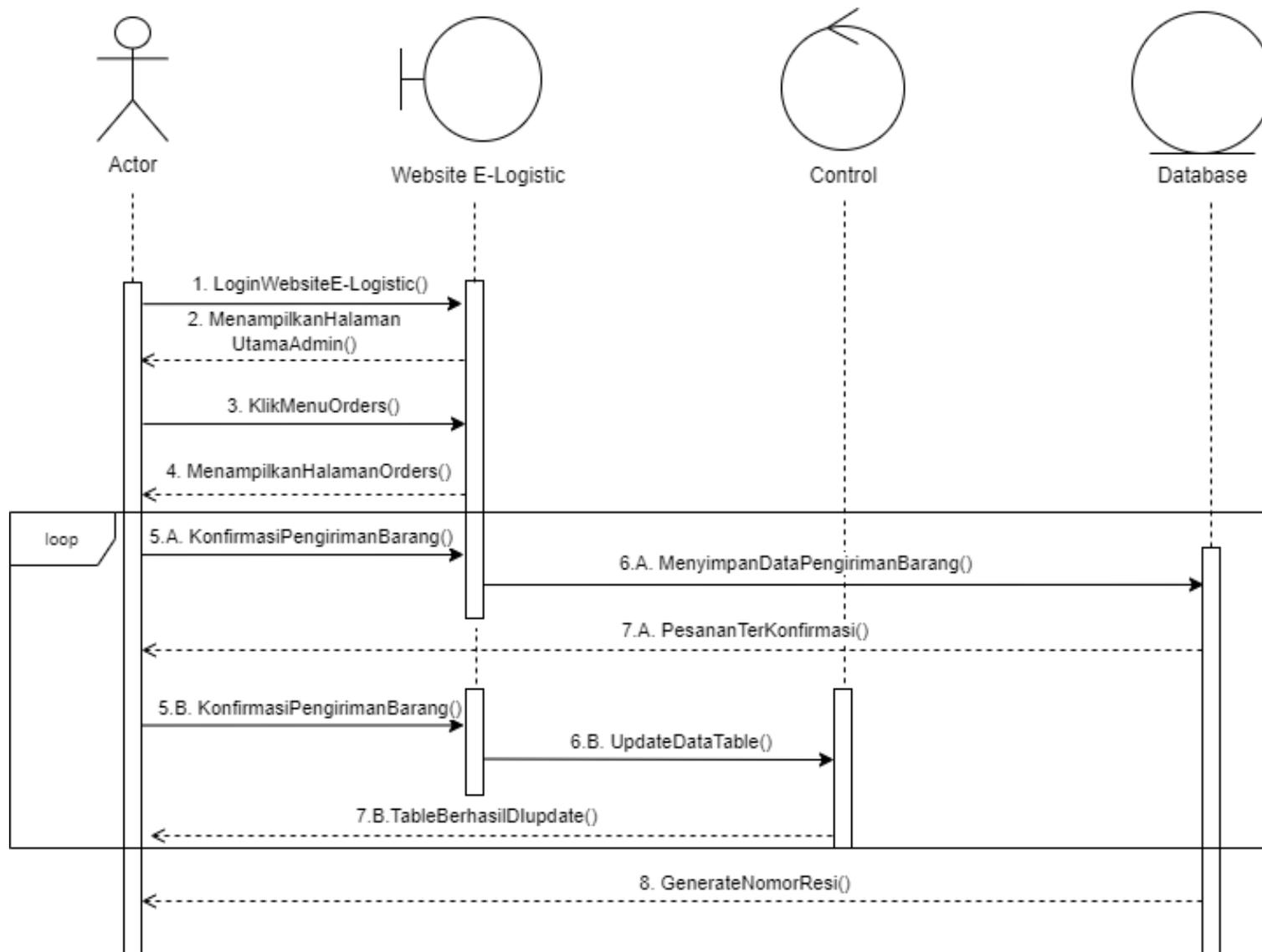
*Sequence Diagram* merupakan sebuah interaksi secara rinci tentang bagaimana suatu tindakan bekerja. *Sequence Diagram* mengambil hubungan antara suatu objek dalam konteks kolaboratif. *Sequence Diagram* berorientasi waktu dan menggunakan sumbu vertikal diagram untuk secara visual mewakili urutan interaksi untuk menunjukkan kapan dan dimana pesan dikirim.



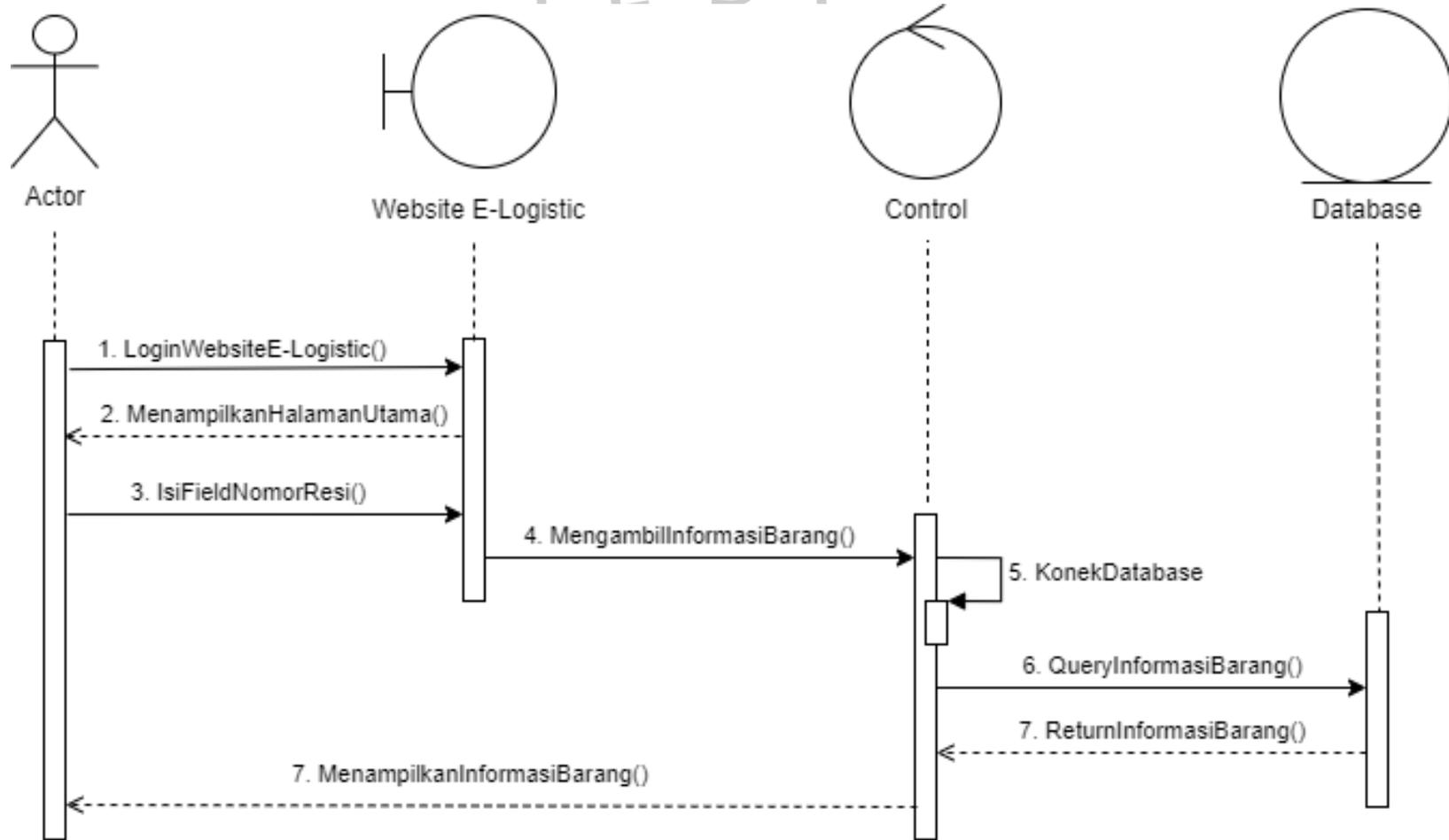
Gambar 4.8 Sequence Diagram Login



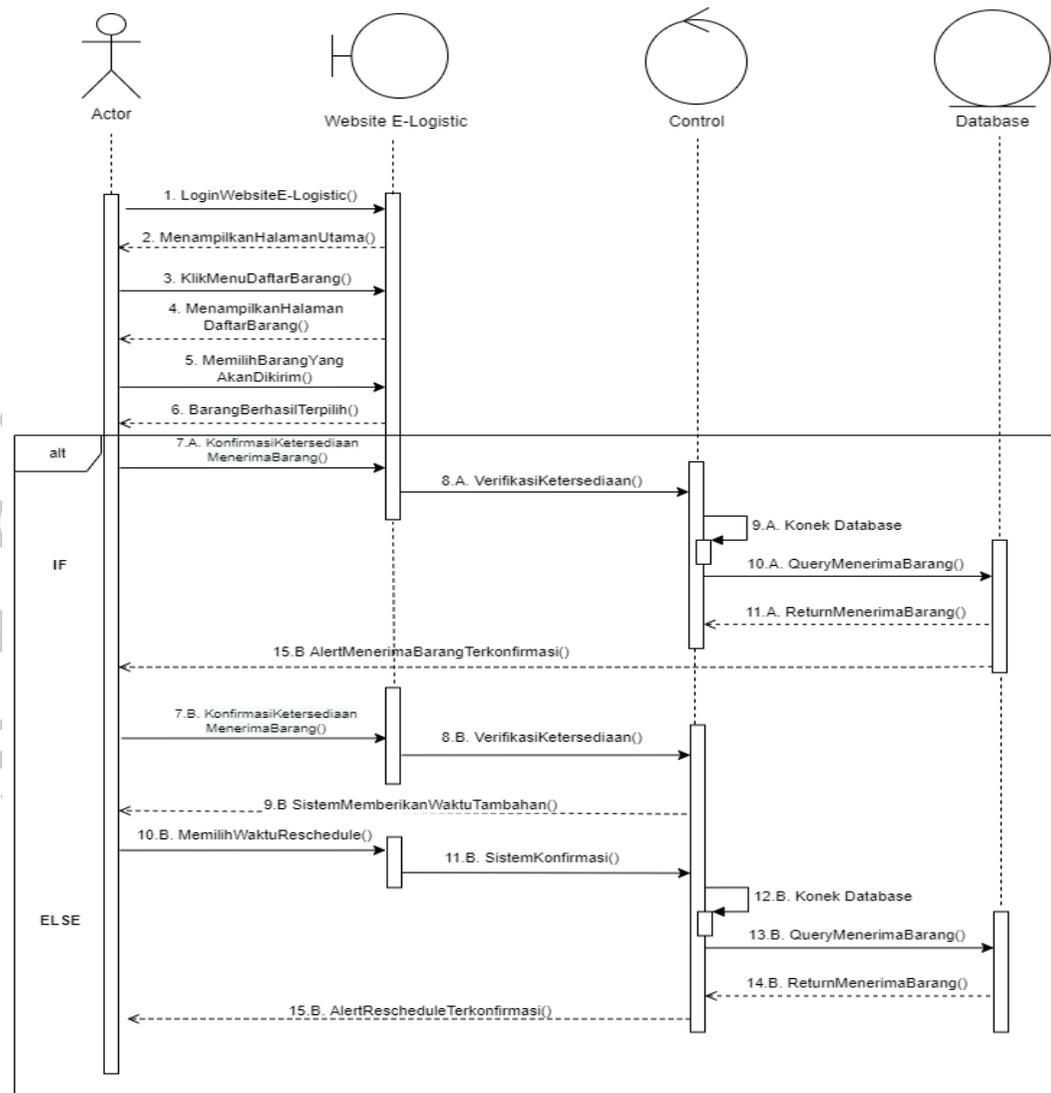
Gambar 4.9 Sequence Diagram Mengirim Barang



Gambar 4.10 Sequence Diagram Kelola Barang



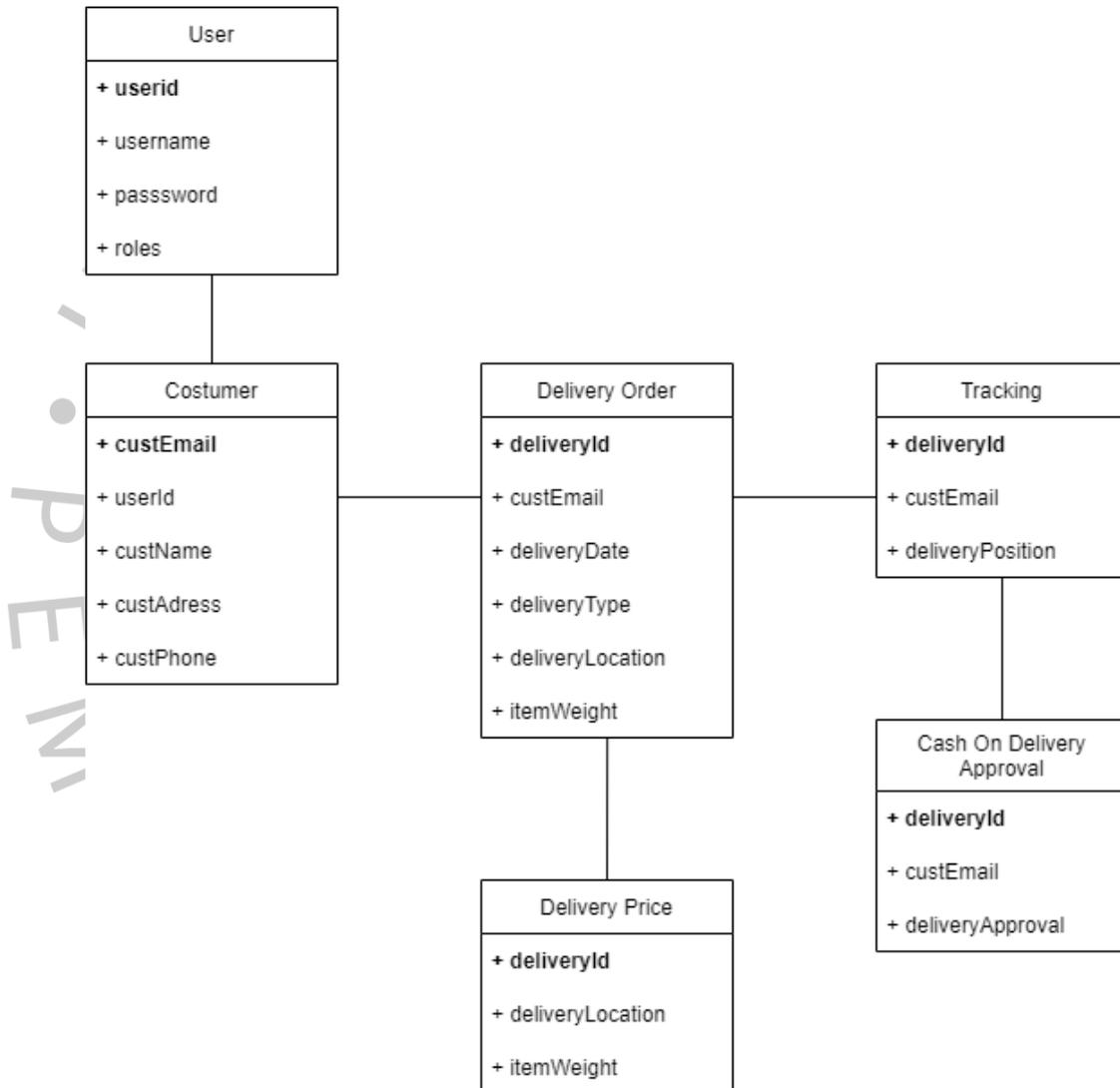
Gambar 4.11 Sequence Diagram Tracking Barang



**Gambar 4.12** Verifikasi Ketersediaan Penerimaan Barang Sistem COD

#### 4.2.5 Class Diagram

*Class Diagram* dalam *Unified Modeling Language (UML)* adalah jenis diagram struktur statis yang menggambarkan struktur suatu sistem dengan menunjukkan kelas sistem, atributnya, operasi (atau metode), dan hubungan antar objek.



**Gambar 4.13** Class Diagram E-Logistic

#### 4.2.6 Kamus Data

Kamus data adalah kumpulan nama, definisi, dan properti elemen data yang digunakan atau dikumpulkan sebagai bagian dari *database*, sistem informasi, atau proyek penelitian. Ini menjelaskan arti dan tujuan item data dalam konteks proyek dan memberikan panduan tentang

interpretasi, makna, dan presentasi yang dapat diterima. Berikut ini merupakan kamus data yang dirancang oleh penulis yang dapat dilihat pada Tabel 4.7.

**Table 4.7** Kamus Data E-Logistic

<b>E- Logistic</b>				
<b>User</b> = {userid, username, password, roles }				
<b>Customer</b> = {custEmail, userId, custName, custAdress, CustPhone }				
<b>Delivery</b>	<b>Order</b>	=	{deliveryId, custEmail, deliveryDate, deliveryType, deliveryLocation, itemWeight }	
<b>Delivery Price</b> = {deliveryId, deliveryLocation, itemWeight }				
<b>Tracking</b> = {deliveryId, custEmail, deliveryPosition }				
<b>Cash On Delivery Approval</b> = {deliveryId, custEmail, deliveryApproval }				

#### 4.2.7 Spesifikasi Basis Data

Spesifikasi basis data merupakan gambaran dari wujud data dalam suatu sistem atau aplikasi. Spesifikasi *database* menjelaskan proses penyimpanan data yang dilakukan dalam perangkat lunak *database*. Format spesifikasi *database* itu sendiri berbentuk sebuah tabel yang menyediakan data yang digunakan. Informasi data yang ditampilkan meliputi nama, jenis, panjang, dan *field* mana yang menjadi sebuah *primary key*. Berikut merupakan hasil rancangan spesifikasi basis data yang dibuat oleh penulis yang dapat dilihat pada Tabel 4.8 sampai dengan 4.13.

**Table 4.8** Spesifikasi Basis Data Tabel User

Tabel User				
	Field Name	Data Type	Size	Description
PK	userId	Int	11	Id User
	username	Varchar	10	Username
	password	Varchar	10	Password User
	roles	Varchar	10	Roles User

**Table 4.9** Spesifikasi Basis Data Tabel Costumer

Tabel Customer				
	Field Name	Data Type	Size	Description
PK	custEmail	Varchar	20	Email Customer
FK	userId	Int	11	Id User
	custName	Varchar	25	Nama Customer
	custAdress	Text	-	Alamat Customer
	custPhone	Varchar	13	Nomor Telepon Customer

**Table 4.10** Spesifikasi Basis Data Tabel Delivery Order

Delivery Order				
	Field Name	Data Type	Size	Description
PK	deliveryId	Int	11	Id Pengiriman
FK	custEmail	Varchar	20	Email Customer
	deliveryDate	Date	-	Tanggal Pengiriman
	deliveryType	Varchar	15	Tipe Pengiriman
	deliveryLocation	Text	-	Lokasi Pengiriman
	itemWeight	Int	11	Berat Barang

**Table 4.11** Spesifikasi Basis Data Tabel Delivery Price

Delivery Price				
	Field Name	Data Type	Size	Description
PK	deliveryId	Int	11	Id Pengiriman
FK	deliveryLocation	Text	-	Lokasi Pengiriman
FK	itemWeight	Int	11	Berat Barang

**Table 4.12** Spesifikasi Basis Data Tabel Tracking

Tracking				
	Field Name	Data Type	Size	Description
PK	deliveryId	Int	11	Id Pengiriman
FK	custEmail	Varchar	20	Email Customer
	deliveryPosition	Text	-	Lokasi Barang

**Table 4.13** Spesifikasi Basis Data Tabel Approval

Cash On Delivery Approval				
	Field Name	Data Type	Size	Description
PK	deliveryId	Int	11	Id Pengiriman
FK	custEmail	Varchar	20	Email Customer
	deliveryApproval	Varchar	10	Approval Pengiriman Barang

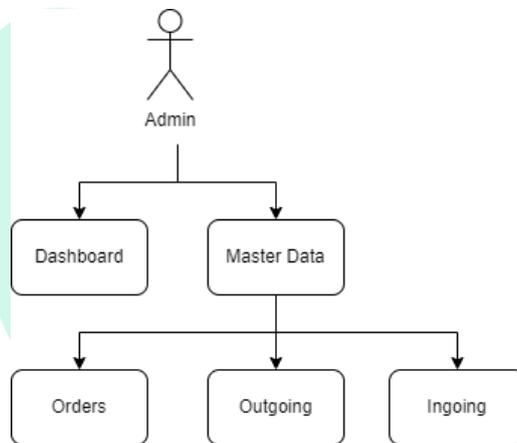
### 4.3 Perancangan Antar Muka

Setelah menganalisis desain sistem, penulis juga mengembangkan desain *User Interface* atau desain antarmuka pengguna untuk aplikasi e-logistik. Perancangan antar muka pengguna yang dibuat oleh penulis meliputi struktur menu, rancangan atau tampilan *layout* layar, rancangan masukan (*input*), rancangan keluaran (*output*), dan rancangan implementasi. Desain *User Interface* atau desain antar muka merupakan sebuah proses mendefinisikan bagaimana sistem akan berinteraksi dengan entitas eksternal (seperti pelanggan, pemasok, dan sistem lainnya). Tujuan dari desain antarmuka pengguna adalah untuk membuat pengalaman pengguna sesederhana dan seefisien mungkin dalam hal mencapai tujuan.

#### 4.3.1 Struktur Menu

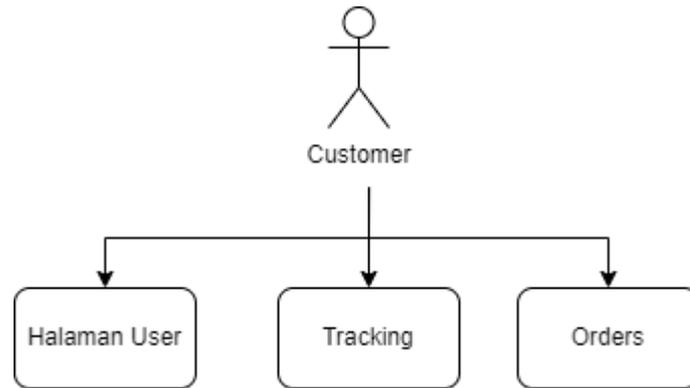
Aplikasi E-Logistic dapat diakses oleh dua jenis aktor, yaitu: Admin, dan Costumer. Setiap aktor memiliki hak akses yang berbeda-beda. Perbedaan hak akses tersebut dapat dilihat dari struktur menu pada setiap aktor. Admin dapat mengakses menu *Dashboard*, dan Master Data,. Pada menu Master Data, Admin dapat mengakses sub-menu *Orders*, *Outgoing* dan *Ingoing*. Menu *Orders* merupakan menu yang menampilkan pesanan pengiriman barang yang telah di pesan oleh

*customer* melalui aplikasi, dan juga pemesanan manual dan input pemesanan barang. *Outgoing* merupakan halaman untuk *input* barang yang ingin dikirimkan keluar dari kantor cabang. Untuk menambahkan data pada menu *outgoing* admin dapat memasukkan nomor resi dari pengiriman barang tersebut. Selain itu admin juga dapat mengakses menu *ingoing*, yaitu merupakan order yang masuk dari cabang lain ke cabang terakhir sebelum barang di kirim dari cabang ke rumah pemilik barang. Gambaran struktur menu dari aktor admin dapat dilihat pada **Gambar 4.13**.



**Gambar 4.13** Struktur Menu Admin

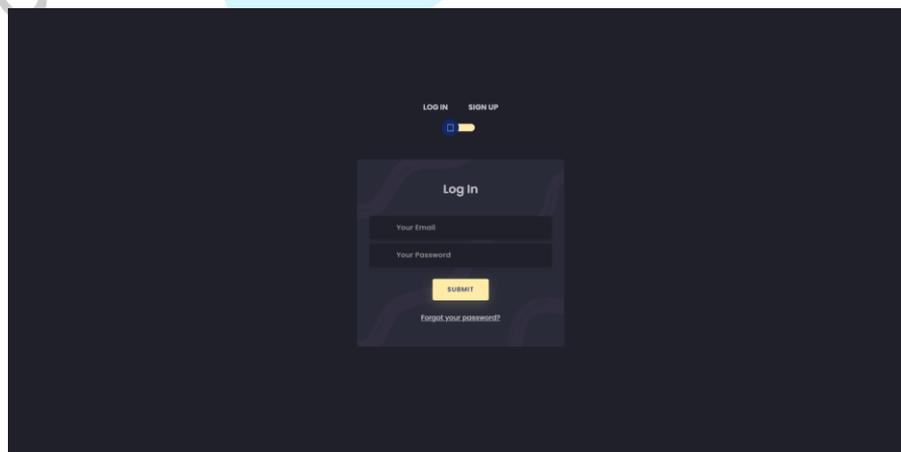
Sementara itu, pada **Gambar 4.14** dapat dilihat struktur menu dari aktor *Customer*. *Customer* dapat mengakses menu-menu seperti halaman user, tracking dan order. Halaman user merupakan halaman depan dari aplikasi tersebut. *Tracking* merupakan menu untuk *customer* memasukkan nomor resi pengiriman barang untuk mengetahui dimana barang tersebut berada. Sementara itu *orders* merupakan halaman untuk *customer* mengirim barang dengan cara memasukkan data pengiriman barangnya.



**Gambar 4.14** Struktur Menu *Customer*

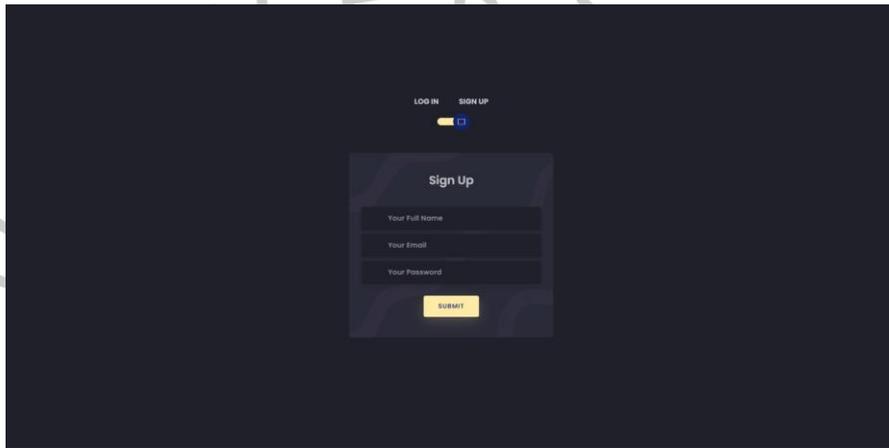
### 4.3.2 Rancangan *Layout* atau Tampilan

Rancangan *layout* atau tampilan dibuat untuk menggambarkan desain tampilan dari sebuah sistem atau aplikasi. Penulis membuat rancangan desain *layout* berdasarkan dari hasil analisis perancangan sistem yang telah dibuat sebelumnya. Pada aplikasi *E-logistic*, penulis merancang desain tampilan untuk halaman *User*, halaman *Login & Register*, halaman *Dashboard*, dan halaman *Tracking & Order*. Pada halaman *Login & Register*, aktor memasukkan *username* dan *password* akun yang telah terdaftar untuk dapat masuk ke dalam aplikasi sesuai dengan role yang diberikan. Selain itu juga *customer* dapat mendaftarkan akun barunya di halaman register Tampilan halaman *Login* dapat dilihat pada **Gambar 4.15**.



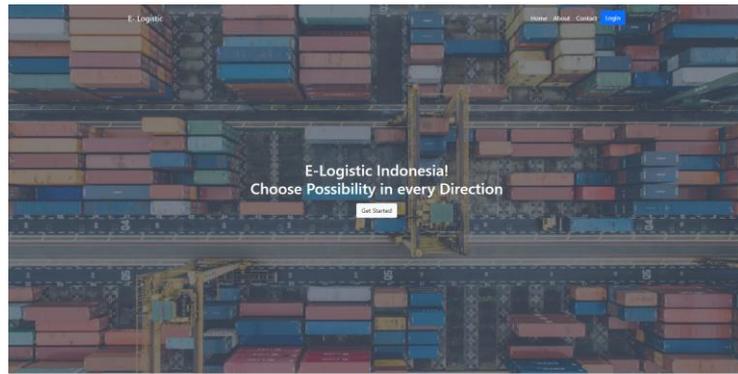
**Gambar 4.15** Halaman *Login*

Pada halaman *register*, user yang dapat mendaftar hanyalah *costumer* saja, *costumer* harus memiliki akun untuk menggunakan aplikasi berupa *tracking* dan sistem *cash on delivery*, karena user dapat melihat informasi barang pesanan secara lengkap dan memiliki data yang bersifat *confidential*. Halaman register di tampilkan pada **Gambar 4.16.**



**Gambar 4.16** Halaman *Register*

Selanjutnya merupakan *homepage* dari aplikasi *e-logistic* tersebut. *homepage* berisi tentang *about e-logistic*, *contact us*, dan juga *button* untuk menuju halaman *login* & *register* yang ditampilkakan pada **Gambar 4.17.**



#### About

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras conamilla turpis in velit blandit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Vestibulum puritiae in utam nibh aliquam. Quisque ultrices accor nisi, quis imperdiet augue accor id. Fusce ultricesper. Suspendisse a nibh eu metus dignissim ultricesper. Nunc ut nisi massa.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras conamilla turpis in velit blandit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Vestibulum puritiae in utam nibh aliquam. Quisque ultrices accor nisi, quis imperdiet augue accor id. Fusce ultricesper. Suspendisse a nibh eu metus dignissim ultricesper. Nunc ut nisi massa.



### Love to Hear From You

It is a long established fact that a reader will be distracted by the readable content of a page when looking at the layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using Content.

Company HeadOffice

Address: 4425 Senterberry Circle Knoxville

Phone: +1 778 460 9127

Website: www.uryyo

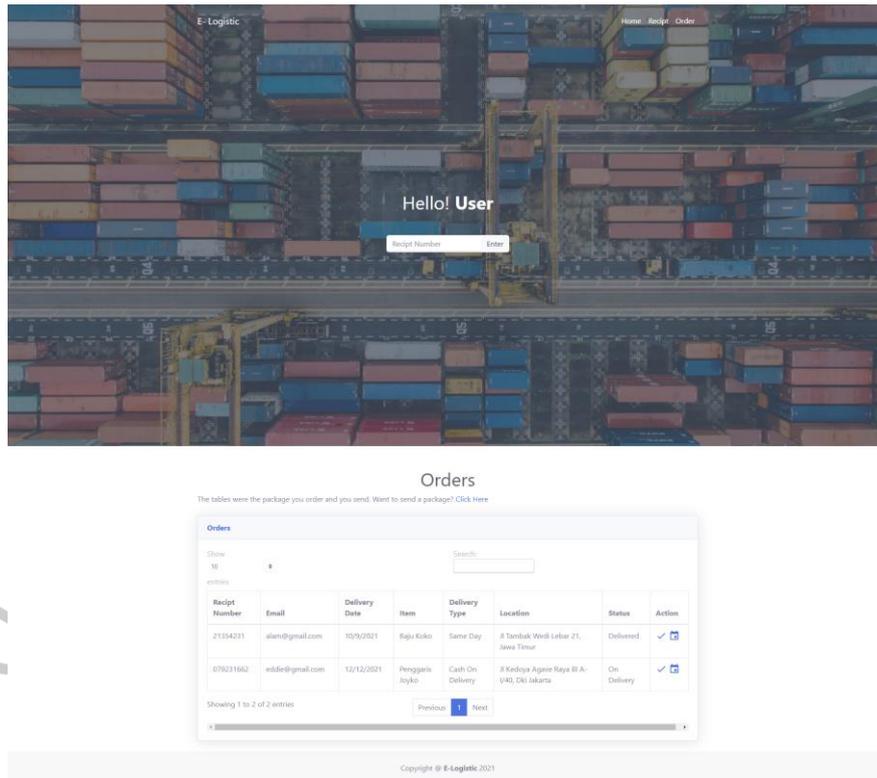
Program: March to Sat: 08:30 AM - 10:30 PM

Send Message

Copyright © E-Logistic 2021

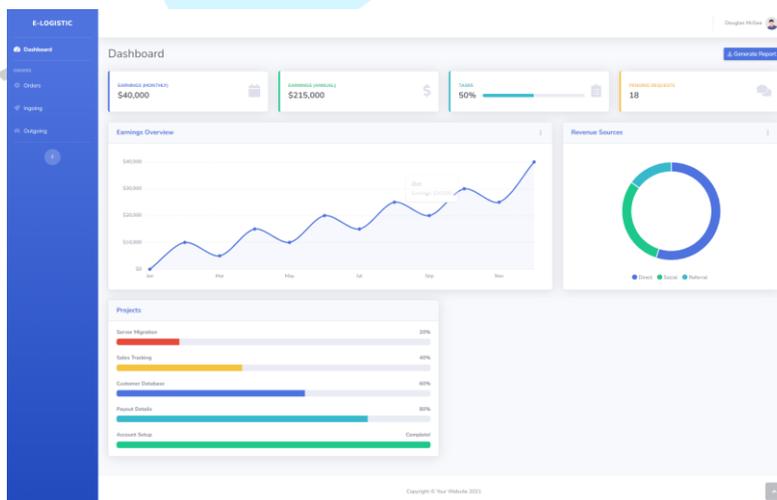
**Gambar 4.17** Halaman *Homepage*

Berikutnya merupakan halaman *costumer*. Pada halaman *costumer* terdapat *field* untuk melakukan *tracking* barang dan juga data barang yang di kirimkan ke rumah pemilik barang. Selain itu juga terdapat modul *popup* untuk memesan pengiriman barang. Selain itu juga *costumer* dapat mengkonfirmasi bahwa barang dengan *system cash on delivery* siap dikirim di rumah ataupun dapat meminta untuk *reschedule* yang di tampilkan pada **Gambar 4.18**.



**Gambar 4.18** Halaman *Costumer*

Berikutnya merupakan halaman *dashboard* admin, pada halaman tersebut terdapat informasi – informasi terkait berapa barang yang dikirim pada hari tersebut, total penghasilan selama sebulan, dan juga grafik – grafik informasi pengiriman barang yang di tampilkan pada **Gambar 4.19**.

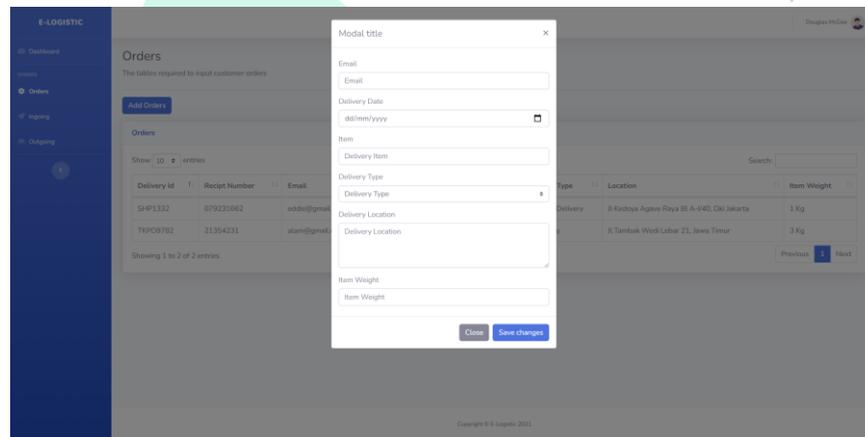


**Gambar 4.19** Halaman *Dashboard*

### 4.3.3 Rancangan Masukan (*Input*)

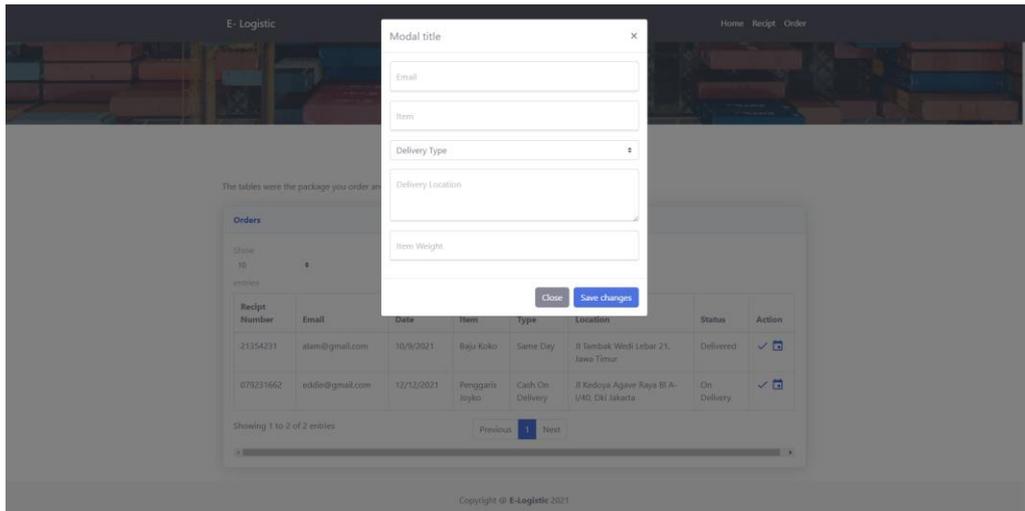
Perancangan pengenalan *input* merupakan awal dari suatu proses pengolahan data berupa bahan baku yaitu komoditas dan objek berupa angka atau tabel. Rancangan input harus benar-benar bagus untuk meyakinkan pengguna.

Rancangan masukan pertama pada aplikasi *logistic* merupakan masukan untuk melakukan order. Order dapat dilakukan secara *online* oleh *costumer*, dan juga dapat di *input* oleh admin melalui pemesanan *offline* di cabang. Rancangan masukan order melalui admin terdapat pada **Gambar 4.20**.



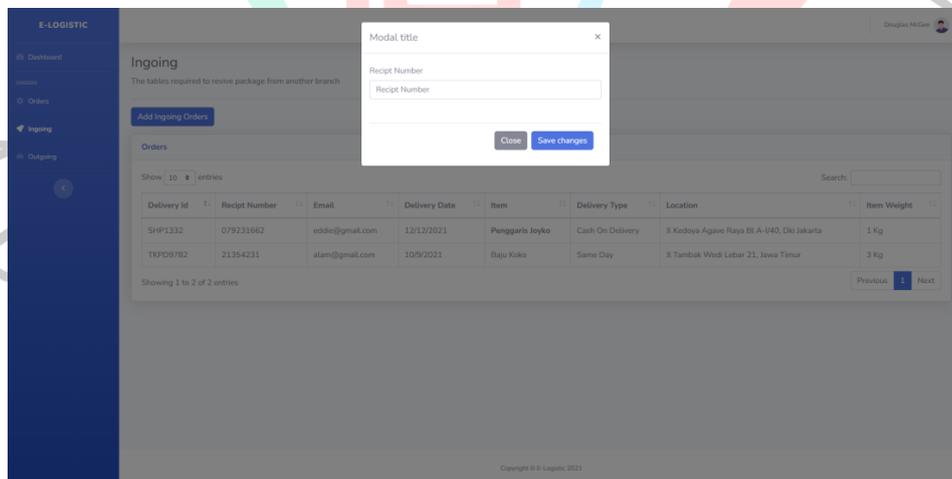
**Gambar 4.20** Order Melalui Admin

Selanjutnya merupakan rancangan pemesanan pengiriman barang melalui *website* yang dilakukan oleh aktor *customer*, sebelum memesan pengiriman barang, *costumer* harus memiliki akun terlebih dahulu. Selain itu juga *costumer* harus mengisi beberapa data pemesanan barang berupa email, alamat pengiriman, tipe pengiriman (*cod*, *same day*, *regular*), berat barang pada **Gambar 4.21**.



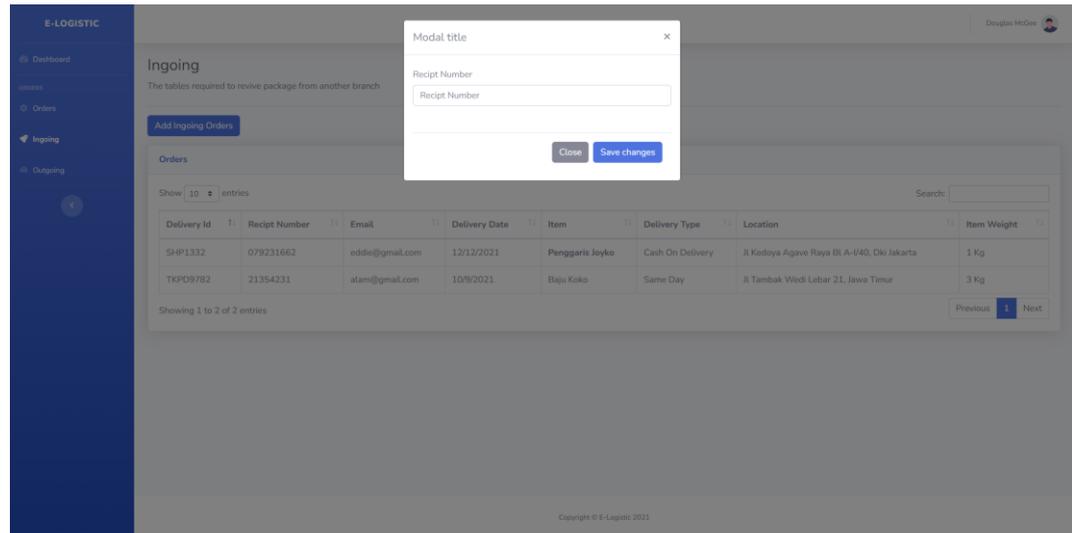
Gambar 4.21 Order Melalui Web

Selanjutnya merupakan rancangan masukan pada halaman *outgoing* dan *incoming*. *Incoming* dan *Outgoing orders* dapat ditambahkan hanya dengan memasukkan nomor resi saja, sehingga sistem akan *fetch* data pemesanan tersebut dan menampilkannya pada *page incoming* dan *outgoing*.



Gambar 4.22 Incoming Orders

*Outgoing* merupakan pengiriman barang dari cabang tersebut ke *warehouse* ataupun cabang terdekat dari rumah pemilik barang. *Input* data *outgoing* dapat dilakukan admin di cabang jasa pengiriman barang dengan memasukkan nomor resi pengiriman ke *webiste*. Setelah itu data akan otomatis ter-*fetch* dan masuk kepada *table* di halaman *outgoing*.

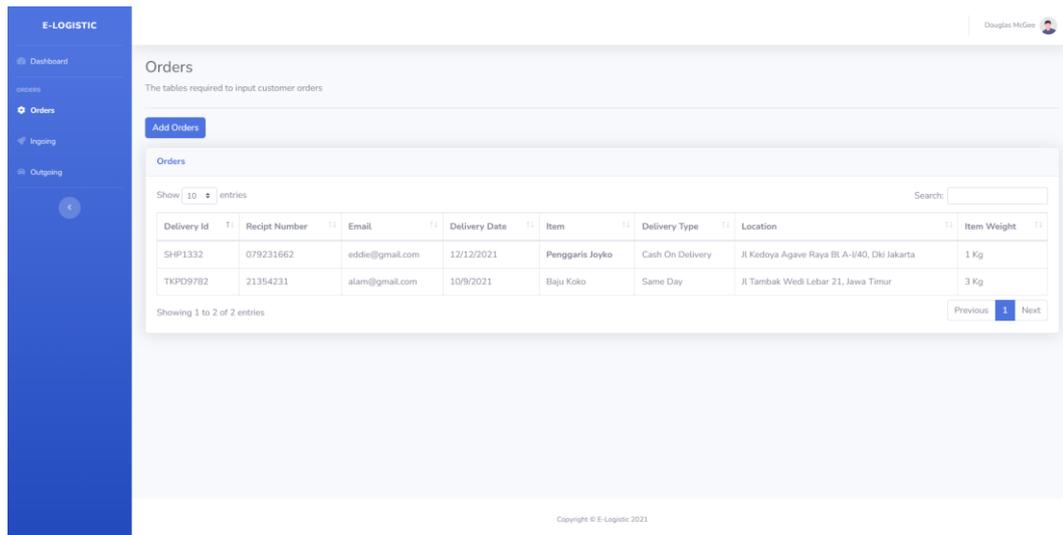


Gambar 4.23 *Outgoing Orders*

#### 4.3.4 Rancangan Keluaran (*Output*)

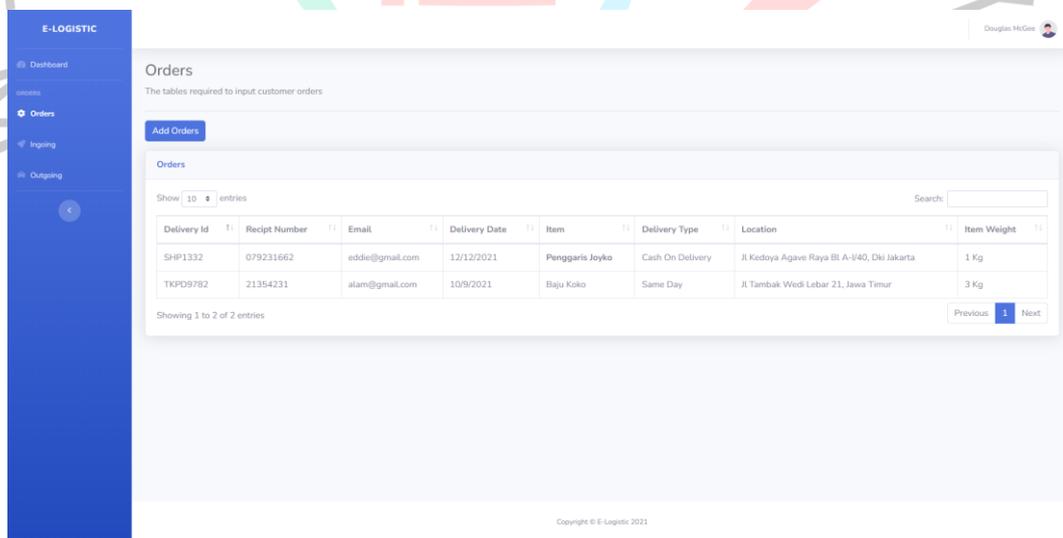
Perancangan keluaran merupakan sebuah informasi yang ditampilkan oleh sistem kepada pemakai aplikasi melalui sistem informasi. Perancangan keluaran penting karena merupakan bentuk aktivitas perancangan sistem fisik yang menghasilkan keluaran sistem untuk dihasilkan dan diusulkan.

Rancangan keluaran yang terdapat pada *e-logistic* terdapat pada 4 halaman yaitu pada halaman *orders*, *outgoing*, *inging* dan juga halaman user. Pada halaman *orders* merupakan *output* dari data pengiriman barang yang telah dilakukan oleh admin secara offline ataupun oleh *customer* melalui *website*. Selanjutnya *inging* dan *outgoing* merupakan barang yang masuk ke cabang dan barang yang keluar di cabang. Terakhir yaitu pada halaman user merupakan output dari pemesanan barang yang berupa informasi barang, dan juga konfirmasi serta *reschedule* penerimaan barang. Rancangan halaman *orders* pada Gambar 4.24.



**Gambar 4.24** Halaman *Orders*

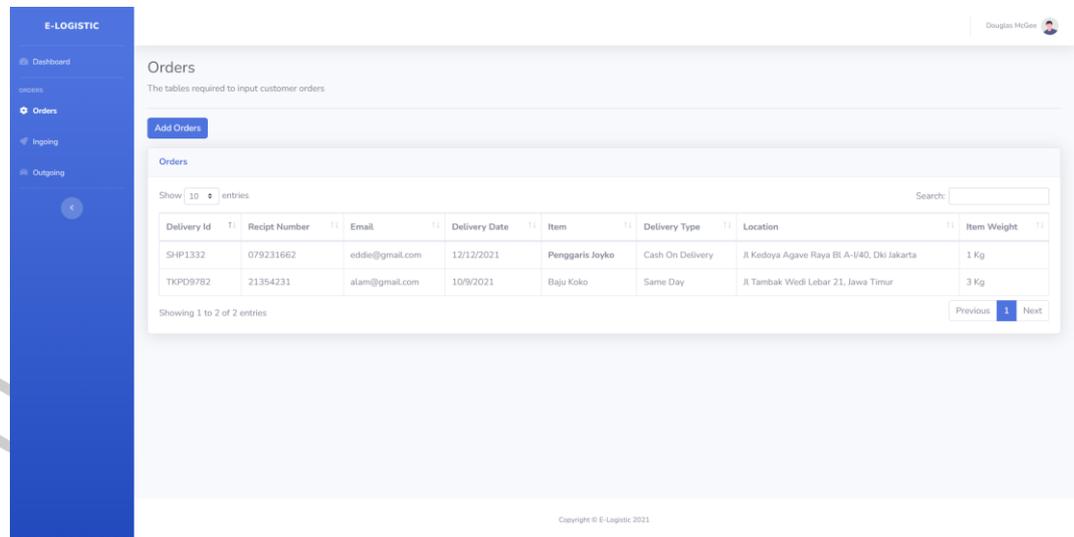
Rancangan keluaran pada halaman *ingoing* merupakan keluaran dari data-data *orders* yang di *input* ke dalam halaman *ingoing* dengan memasukkan nomor resi, data *ingoing* merupakan pengiriman masuk dari cabang lain ke cabang terkait. Halaman *ingoing* ditunjukkan pada **Gambar 4.25**.



**Gambar 4.25** Halaman *Ingoing*

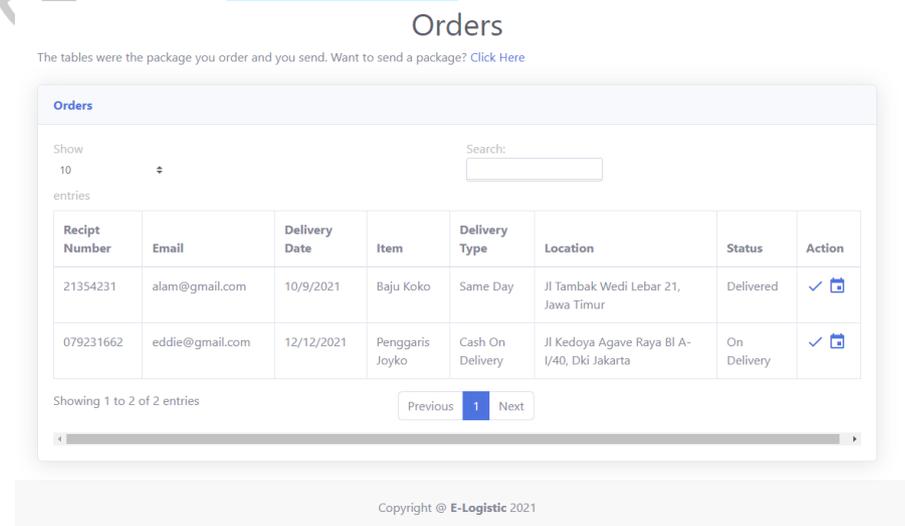
Pada Rancangan keluaran halaman *outgoing*, sistem akan menampilkan data barang yang akan keluar dari cabang ke cabang terpilih ataupun ke *warehouse*. Rancangan tersebut ditujukan untuk

melakukan *tracking* dimana barang berada yang nantinya akan masuk ke dalam informasi ke pemilik barang. Halaman *outgoing* ditampilkan pada **Gambar 4.26**.



**Gambar 4.26** Halaman *Outgoing*

Rancangan keluaran pada halaman user merupakan data dari pengiriman barang yang dimiliki oleh *costumer*. Data – data tersebut merupakan data dimana barang berada, informasi barang yang dimiliki user, dan juga terdapat konfirmasi pengiriman barang dan juga *reschedule* barang.



**Gambar 4.27** Halaman User

#### 4.4 Perancangan Implementasi

Setelah tahap analisis dan perancangan telah dijalankan, maka tahap selanjutnya adalah implementasi. Implementasi adalah tahap implementasi sekaligus pengujian untuk sistem baru dan merupakan tahap dimana aplikasi siap dijalankan dalam keadaan nyata, efek dari sistem baru akan diketahui dengan pasti, serta segala kelebihan dan kekurangan dari sistem dan program aplikasi. Testing yang dilakukan oleh penulis bertujuan untuk memeriksa apakah terdapat *bugs* yang dapat mengganggu jalannya sistem. Pengujian dilakukan dengan cara memeriksa apakah dari hasil *input* yang dimasukkan oleh penulis akan menghasilkan *output* yang sesuai dengan yang diharapkan. Setelah tidak ditemukan *bugs* atau *error*, tahap selanjutnya adalah mengimplementasikan aplikasi tersebut untuk dapat digunakan oleh user. Berikut merupakan rancangan testing yang akan dilakukan oleh penulis,

- rancangan testing dapat dilihat pada **Tabel 4.14**.

**Table 4.14** Rancangan Testing

No.	Test Name	Test Steps	Expected Results
1.	Login	<ol style="list-style-type: none"><li>1. Mengakses Aplikasi E-Logistic</li><li>2. Mengisi Username dan Password</li><li>3. Klik Button Login</li></ol>	Login Sukses
2.	Register	<ol style="list-style-type: none"><li>1. Mengakses Aplikasi E-Logistic</li><li>2. Mengisi email, username, password, no handphone.</li><li>3. Klik button register</li></ol>	Register Sukses
3.	Admin Add Orders	<ol style="list-style-type: none"><li>1. Mengakses Aplikasi E-Logistic</li><li>2. Klik menu orders</li><li>3. Klik button add orders</li><li>4. Klik submit button</li></ol>	Add Orders Sukses
4.	Admin Edit Orders	<ol style="list-style-type: none"><li>1. Mengakses Aplikasi E-Logistic</li><li>2. Klik menu orders</li><li>3. Klik button edit</li><li>4. Klik submit button</li></ol>	Edit Orders Sukses
5.	Admin Delete Orders	<ol style="list-style-type: none"><li>1. Mengakses Aplikasi E-Logistic</li><li>2. Klik menu orders</li><li>3. Klik button delete</li></ol>	Delete Orders Sukses

6.	Admin Add Ingoing	<ol style="list-style-type: none"> <li>1. Mengakses Aplikasi E-Logistic</li> <li>2. Klik menu Ingoing</li> <li>3. Klik button add ingoing orders</li> <li>4. Klik submit button</li> </ol>	Add Ingoing Orders Sukses
7.	Admin Edit Ingoing Orders	<ol style="list-style-type: none"> <li>1. Mengakses Aplikasi E-Logistic</li> <li>2. Klik menu ingoing orders</li> <li>3. Klik button edit</li> <li>4. Klik submit button</li> </ol>	Edit Ingoing Orders Sukses
8.	Admin Delete Ingoing Orders	<ol style="list-style-type: none"> <li>1. Mengakses Aplikasi E-Logistic</li> <li>2. Klik menu ingoing orders</li> <li>3. Klik button delete</li> </ol>	Delete Ingoing Orders Sukses
9.	Admin Add Outgoing Orders	<ol style="list-style-type: none"> <li>4. Mengakses Aplikasi E-Logistic</li> <li>5. Klik menu Outgoing</li> <li>6. Klik button add outgoing orders</li> <li>7. Klik submit button</li> </ol>	Add Outgoing Orders Sukses
10.	Admin Edit Outgoing Orders	<ol style="list-style-type: none"> <li>1. Mengakses Aplikasi E-Logistic</li> <li>2. Klik menu outgoing orders</li> <li>3. Klik button edit</li> <li>4. Klik submit button</li> </ol>	Edit Outgoing Orders Sukses
11.	Costumer Cek Resi	<ol style="list-style-type: none"> <li>1. Mengakses Aplikasi E-Logistic</li> <li>2. Klik input field cek resi</li> <li>3. Klik submit button</li> </ol>	Cek Resi Sukses
12.	Costumer Add Orders	<ol style="list-style-type: none"> <li>1. Mengakses Aplikasi E-Logistic</li> <li>2. Klik add orders</li> <li>3. Klik submit button</li> </ol>	Customer Add Orders Sukses
13.	Costumer Confirm Cash On Delivery Orders	<ol style="list-style-type: none"> <li>1. Mengakses Aplikasi E-logistic</li> <li>2. Klik button confirm</li> </ol>	Confirm Data Added
14.	Costumer Reschedule Cash On Delivery orders	<ol style="list-style-type: none"> <li>1. Mengakses Aplikasi E-Logistic</li> <li>2. Klik Button Reschedule</li> <li>3. Klik Submit Button</li> </ol>	Reschedule Data Added

