

## **BAB IV**

### **HASIL DAN ANALISIS PENELITIAN**

#### **4.1. Analisis Perancangan sistem**

Menurut kristanto (2018:6) Analisis perancangan sistem itu memiliki tiga perangkat yang di butuhkan yaitu perangkat manusia, perangkat lunak, dan perangkat keras. Perangkat keras biasanya berupa komputer, Perangkat lunak adalah sebuah program dan perangkat manusia biasanya berupa manajer, programmer, analisis sistem dan lain-lain. Kalo menurut Mulyani(2016:38) Analisis sistem merupakan suatu teknik penelitian terhadap sebuah sistem dengan menguraikan komponen-komponen pada suatu sistem dengan tujuan untuk mempelajari komponen itu dengan keterkaitannya dengan komponen lain yang membentuk suatu sistem sehingga dapat mengambil keputusan atau kesimpulan mengenai sistem tersebut untuk mengetahui baik itu kelebihan atau kelemahannya.

Analisis perancangan sistem juga adalah sebuah proses penjabaran yang berasal dari sistem dari bagian ke bagian untuk melakukan evaluasi permasalahan dengan tujuan agar setiap program dapat memenuhi suatu kebutuhan dari penggunaannya dan di harapkan dapat berfungsi dengan baik. Untuk melakukan analisis perancangan sistem dapat dilakukan untuk menganalisa sistem yang berjalan serta mendokumentasi kebutuhan-kebutuhan suatu pengguna menurut(Erliyani et al., 2020).

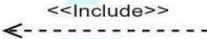
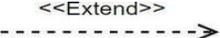
#### **4.2. Perancangan Diagram sistem usulan**

Pengunaan UML dalam mendesain suatu sistem bertujuan untuk memberikan gambaran secara luas dengan sistem yang akan digunakan atau dikembangkan dan dirancang. Perancangan diagram sistem usulan itu digunakan untuk pengembangan dan pembuatan suatu sistem informasi pembelian barang *second* dengan menggunakan pendekatan metode *Object Oriented Analysis Design(OOAD)*. Yaitu metode ini terdiri dari lima diagram yaitu *Use Case Diagram, Sequence Diagram, Activity Diagram, Class Diagram dan ERD*. Pada penelitian ini penulis memberikan suatu usulan dari beberapa permasalahan yang terjadi, yaitu dengan merancang sebuah sistem pembelian barang *second* pada toko authentic.secondary.

### 4.2.1 Use Case Diagram

Use case diagram mengilustrasikan fungsi-fungsi utama dari sistem dan pengguna yang akan berinteraksi dengan sistem tersebut dengan sederhana. Use case diagram memiliki elemen-elemen, yaitu *actors*, *use case*, *subject boundaries*, dan sekumpulan *relationship* antara *actors* dan *use case*. Berikut merupakan penjelasan dari elemen-elemen pada **Tabel 4.1** di bawah ini (Dennis, Tegarden, & Wixom, 2015, p. 121)

**Tabel 4.1 Deskripsi Use Case Diagram**

Nama	Notasi	Deskripsi
<i>Actor/Role</i>		<i>Actor</i> didefinisikan peran atau pengguna untuk berinteraksi dengan sistem. <i>Actor</i> bisa dijadikan saling bertukar meliputi baik manusia maupun organisasi.
<i>Use Case</i>		Use Case merupakan tindakan dan interaksi yang dilakukan <i>actor</i> . Use Case digambarkan bentuk elips yang horizontal ditampilkan oleh sistem.
<i>Association Relationship</i>		Asosiasi adalah hubungan antara dua pengklasifikasi, seperti kelas atau kasus penggunaan, yang menjelaskan alasan hubungan dan aturan yang mengatur hubungan tersebut.
<i>Include Relationship</i>		<i>Include</i> merupakan fungsionalitas dari use case lainnya dan menspesifikasikan secara eksplisit.
<i>Extend Relationship</i>		<i>Extend</i> menunjukkan bahwa target dapat memperluas perilaku dari <i>use case</i> sumber dan merupakan relasi tambahan.
<i>Generalization Relationship</i>		Generalisasi merupakan hubungan antara kedua entitas sedemikian rupa sehingga satu entitas mewarisi fungsionalitas entitas lain. Dimana fungsi diwakili oleh panah ke arah <i>use case</i> atau <i>actor</i> .
<i>System Boundary</i>		<i>System Boundary</i> adalah persegi panjang yang dapat digambar dalam diagram kasus penggunaan untuk memisahkan kasus penggunaan internal sistem dari aktor yang berada di luar sistem

Sumber : (Business Process and Functional Modeling, chapter 4. p.122)



**Gambar 4.1** Use case diagram Pembelian barang second

Berikut ini merupakan penjelasan dari *use case* diagram pada **Gambar 4.1** di atas

1. Calon pelanggan membuka halaman registrasi untuk melakukan pendaftaran di situs web atau aplikasi pembelian barang *second*.
2. Aktivasi email, setelah calon pelanggan berhasil mendaftar mereka menerima email aktivasi pada alamat email yang telah mereka daftarkan. Aktivasi email berisi tautan atau intruksi untuk mengaktifkan akun mereka.
3. Melihat daftar outfit, pelanggan dapat melihat daftar outfit atau pakaian yang tersedia di platform pembelian barang *second*, pelanggan bisa melihat daftar outfit setelah masuk menggunakan *email* dan *password*.

4. Melihat wishlist, dimana pelanggan melihat daftar barang yang di favoritkan.
5. Tambah keranjang, Pelanggan dapat menelusuri katalog produk dan memilih pakaian yang ingin di beli. Setelah menemukan produk yang diinginkan, pelanggan dapat menambahkannya ke keranjang belanja.
6. Kurangi keranjang, pelanggan mengurangi jumlah outfit yang ada di dalam keranjang belanja saat belanja.
7. *Checkout* adalah langkah terakhir dalam pembelian di website pembelian barang *second*. Setelah pelanggan menambahkan produk ke keranjang, mereka melanjutkan ke proses *checkout* untuk menyelesaikan pembelian.
8. Melihat riwayat transaksi, pelanggan masuk ke dalam akun mereka di pembelian barang *second* untuk melihat riwayat transaksi. Riwayat transaksi mencakup detail pesanan sebelumnya yang telah mereka lakukan.
9. Kelola *outfit*, *owner* kelola *outfit* yang dijual di pembelian barang *second* ini mencakup penambahan produk baru, pembaharuan informasi produk, dan penghapusan produk yang tidak lagi tersedia.
10. Kelola biodata toko, pemilik toko dapat mengelola akun mereka seperti mengubah informasi profil dan mengelola data akun di situs web aplikasi pembelian barang *second*.
11. Kelola pesanan pelanggan, *owner* dari pembelian barang *second* dapat mengelola pesanan seperti mengonfirmasi pesanan, memproses pengiriman dan menangani permintaan pelanggan.
12. List dashboard penjualan. Pemilik toko berhasil masuk kedalam sistem untuk melihat data penjualan produk dan untuk melihat grafik dan diagram tabel.

#### **4.2.2 Spesifikasi Use Case Diagram**

Spesifikasi use case diagram adalah dokumen yang merinci bagaimana suatu sistem atau aplikasi berinteraksi dengan aktor (pengguna atau sistem eksternal lainnya) dalam situasi tertentu. Use case diagram sendiri adalah representasi visual dari interaksi antara aktor dan sistem.

**Tabel 4. 1** Specification Registrasi

Use Case Name	Registrasi	
Actor	Calon Pelanggan	
Description	Calon pelanggan melakukan registrasi	
Pre-Condition	Calon pelanggan mengisi form registrasi terlebih dahulu	
Post-Condition	Calon pelanggan berhasil melakukan login	
Sukses Skenario	Aktor	Sistem
	<ol style="list-style-type: none"> <li>1. Klik halaman registrasi</li> <li>2. Input email dan Input password</li> <li>3. Klik button registrasi</li> <li>5. Klik link aktivasi di email</li> </ol>	<ol style="list-style-type: none"> <li>4. Sistem verifikasi email, jika email terpakai kembali ke hal registrasi dan jika email tersedia sistem mengirim link aktivasi di email</li> <li>6. Sistem, jika “yes” link expire di arahkan ke hal registrasi, jika “no” di arahkan ke hal login.</li> </ol>
Alternatives flows	Jika validasi gagal, sistem akan menampilkan halaman registrasi kembali dengan <i>alert</i> (peringatan) kesalahan penginputan	

**Tabel 4. 2** Specification aktivasi email

Use Case Name	Aktivasi email	
Actor	Calon pelanggan	
Description	Calon pelanggan melakukan aktivasi email	
Pre-Condition	Calon pelanggan telah berhasil menyelesaikan pendaftaran	
Post-Condition	Akun calon pelanggan sudah diaktifkan	
Sukses Skenario	Aktor	Sistem
	<ol style="list-style-type: none"> <li>1. Klik link aktivasi email</li> </ol>	<ol style="list-style-type: none"> <li>2. Melakukan aktivasi email</li> <li>3. Memberikan aktivasi email kepada user</li> </ol>
Alternatives flows	Jika pelanggan belum mendapatkan aktivasi email, pastikan kembali aktivasi email di cek kembali dan melihat di bagian spam email.	

**Tabel 4. 3** *Specification Melihat daftar outfit*

Use Case Name	Melihat daftar outfit	
Actor	Pelanggan	
Description	Pelanggan telat melihat daftar outfit	
Pre-Condition	Pelanggan telah masuk ke dalam sistem	
Post-Condition	Sudah login dan menampilkan halaman lihat daftar outfit	
Sukses Skenario	Aktor	Sistem
	1.Pelanggan menekan button daftar outfit	2. Menampilkan halaman Daftar outfit
Alternatives flows	Pelanggan memiliki opsi untuk kembali ke halaman daftar outfit atau kategori untuk melihat produk lainnya.	

**Tabel 4. 4** *Specification melihat wishlist*

Use Case Name	Melihat wishlist	
Actor	Pelanggan	
Description	Pelanggan untuk melihat produk yang tersedia di platform.	
Pre-Condition	Sudah masuk ke bagian wishlist	
Post-Condition	Sudah login menandai outfit yang di favoritkan	
Sukses Skenario	Aktor	Sistem
	1.Menekan button melihat wishlist	2.Sistem menandai wishlits yang di pilih
Alternatives flows	Pelanggan memiliki opsi untuk kembali ke halaman daftar produk atau kembali untuk melihat produk lainnya.	

**Tabel 4.5** *Specification tambah keranjang*

Use Case Name	Tambah keranjang	
Actor	Pelanggan	
Description	Pelanggan untuk menambahkan produk ke dalam keranjang belanja mereka	
Pre-Condition	Pelanggan masuk ke dalam sistem tambah outfit	
Post-Condition	Sudah login dan berhasil menambah jumlah outfit	
Sukses Skenario	Aktor	Sistem
	1. Melihat salah satu produk 2. Pelanggan klik button keranjang	3. Sistem menambahkan keranjang yang dipilih
Alternatives flows	Jika stok produk yang diminta tidak mencukupi, sistem memberikan pesan bahwa produk tidak tersedia dalam jumlah yang diminta.	

**Tabel 4.6** *Specification kurangi keranjang*

Use Case Name	Kurangi outfit ke keranjang	
Actor	Pelanggan	
Description	Pelanggan mengurangi jumlah outfit ke dalam keranjang belanja mereka	
Pre-Condition	Pelanggan sudah menambahkan outfit ke keranjang	
Post-Condition	Sudah login dan berhasil mengurangi outfit ke keranjang	
Sukses Skenario	Aktor	Sistem
	1. Menekan button kurangi outfit ke keranjang	2. Sistem berhasil mengurangi
Alternatives flows	Jika sistem tidak bisa mengurangi outfit ke keranjang silahkan di coba memuat ulang	

**Tabel 4. 7 Specification Checkout**

Use Case Name	Checkout	
Actor	Pelanggan	
Description	Pelanggan melakukan checkout pembelian produk.	
Pre-Condition	Sudah tambah outfit ke keranjang	
Post-Condition	Sudah login dan Data order sudah bertambah	
Sukses Skenario	Aktor	Sistem
	1.Masuk ke halaman profile, input nama lengkap, nomor telepon dan alamat lengkap. 2.User mencari barang 3.Masuk ke keranjang 4.Checkout ke keranjang 6.Masuk ke halaman pembayaran pilih kurir dan pilih pembayaran 7.Bayar ke kode rekening 8.User cek riwayat pembayaran	5.Sistem, “jika ada barang lain yang ingin ditambah” jika ya masuk ke halaman cari barang jika tidak masuk ke halaman pembayaran  8.Sistem.”kode bayar expire” jika ya diarahkan ke check out keranjang jika tidak di arahkan ke cek riwayat pembayaran
Alternatives Flows	Jika pada checkout outfit terdapat masalah lain selama proses pembayaran, harap memuat ulang.	

**Tabel 4. 8 Specification Melihat riwayat transaksi**

Use Case Name	Melihat riwayat transaksi	
Actor	Pelanggan	
Description	Pelanggan melihat riwayat transaksi	
Pre-Condition	Status riwayat transaksi masih <i>default</i>	
Post-Condition	Status riwayat transaksi sudah berhasil berubah	
Sukses Skenario	Aktor	Sistem
	1.Klik button riwayat transaksi  3.Pelanggan melihat riwayat transaksi	2.Menampilkan data riwayat transaksi
Alternatives flows	Pelanggan memiliki opsi untuk kembali ke halaman Home.	

**Tabel 4.9** *Specification kelola outfit*

Use Case Name	Kelola Outfit	
Actor	Pemilik Toko	
Description	Pemilik Toko mengelola outfit	
Pre-Condition	Pemilik toko masuk ke dalam sistem kelola outfit	
Post-Condition	Pemilik toko berhasil kelola outfit	
Sukses Skenario	Aktor	Sistem
	1. Select Dashboard Pakaian, user melakukan add, edit dan delete  3. Klik data tersimpan	2. Sistem "Jika error masuk ke select dashboard pakaian dan jika sukses data tersimpan.
Alternatives flows	-	

**Tabel 4. 10** *Specification kelola biodata toko*

Use Case Name	Kelola Biodata Toko	
Actor	Pemilik toko	
Description	Pemilik toko melakukan kelola biodata toko	
Pre-Condition	Pemilik toko masuk ke dalam sistem kelola biodata outfit	
Post-Condition	Pemilik toko berhasil kelola biodata toko	
Sukses Skenario	Aktor	Sistem
	1. Menekan button menu kelola biodata toko  3. Melakukan pemantauan kelola biodata toko	2. Menampilkan halaman kelola biodata toko
Alternatives flows	-	

**Tabel 4. 11** *Specification kelola pesanan pelanggan*

Use Case Name	Kelola pesanan pelanggan	
Actor	Pemilik toko	
Description	Pemilik toko melakukan kelola pesanan pelanggan	
Pre-Condition	Pemilik toko masuk ke dalam sistem kelola pesanan pelanggan	
Post-Condition	Pemilik toko berhasil kelola pesanan pelanggan	
Sukses Skenario	Aktor	Sistem
	1. Pelanggan kirim kode rekening ke pemilik toko 2. Pemilik toko cek dashboard pesanan 4. Button konfirmasi pesanan 6. Packing pesanan untuk ke Kurir 7. Button input nomor resi dari kurir 9. User menyapaikan ke pelanggan cek riwayat pesanan	3. Sistem, jika saldo masuk ke rekening jika belum cek row dashboard pesanan, jika sudah masuk button konfirmasi pesanan. 5. Ubah status “menunggu” ke “diproses” 8. Sistem, ubah status “diproses” ke “terkirim” dan input nomor resi ke field no_resi
Alternatives Flows	Jika kelola pesanan pelanggan terdapat masalah pesanan tidak dapat dikelola silahkan di coba lagi dan memuat ulang	

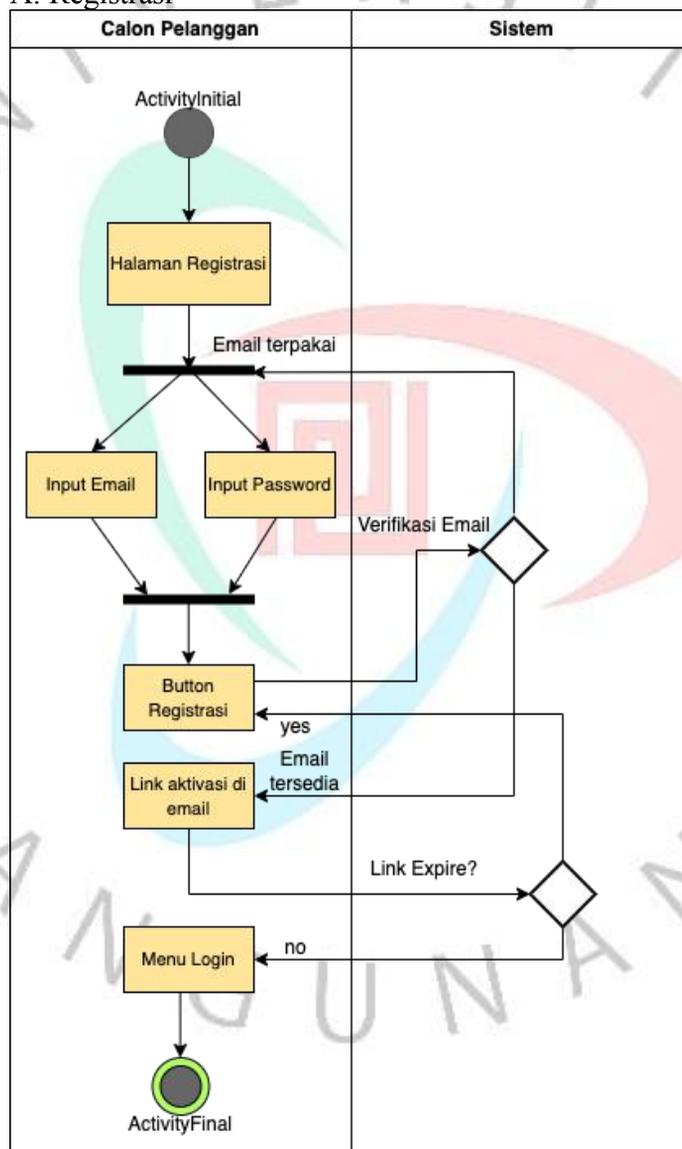
**Tabel 4. 12** *Specification liat dashboard penjualan*

Use Case Name	Lihat dashboard penjualan	
Actor	Pemilik toko	
Description	Pemilik toko dapat melihat dashboard penjualan, seperti berapa banyak barang yang dikeluarkan	
Pre-Condition	Pemilik toko telah masuk ke dalam sistemnya	
Post-Condition	Pemilik toko berhasil melihat dan menginterpretasikan dashboard penjualan	
Sukses Skenario	Aktor	Sistem
	1. User berhasil masuk kedalam sistem 2. User masuk ke dalam Dashboard penjualan	3. Menampilkan dashboard penjualan
Alternatives Flows	Jika tidak bisa melihat dashboard penjualan. Silahkan di muat ulang	

### 4.2.3 Activity diagram

*Activity diagram* menurut Handini(2016) *Activity diagram* adalah suatu aliran aktifitas atau kerja dari sebuah sistem atau suatu proses bisnis. *Activity diagram* itu dihubungkan dengan panah, panah tersebut mengarahkan suatu urutan aktivitas dalam organisasi. *Activity diagram* juga menggambarkan alur kontrol secara garis besar. Meskipun diagram aktivitas menunjukkan suatu aliran yang berbeda seperti bercabang ataupun paralel. *Activity diagram* menggambarkan proses bisnis sebuah proses dari urutan aktivitas agar lebih mudah dipahami.

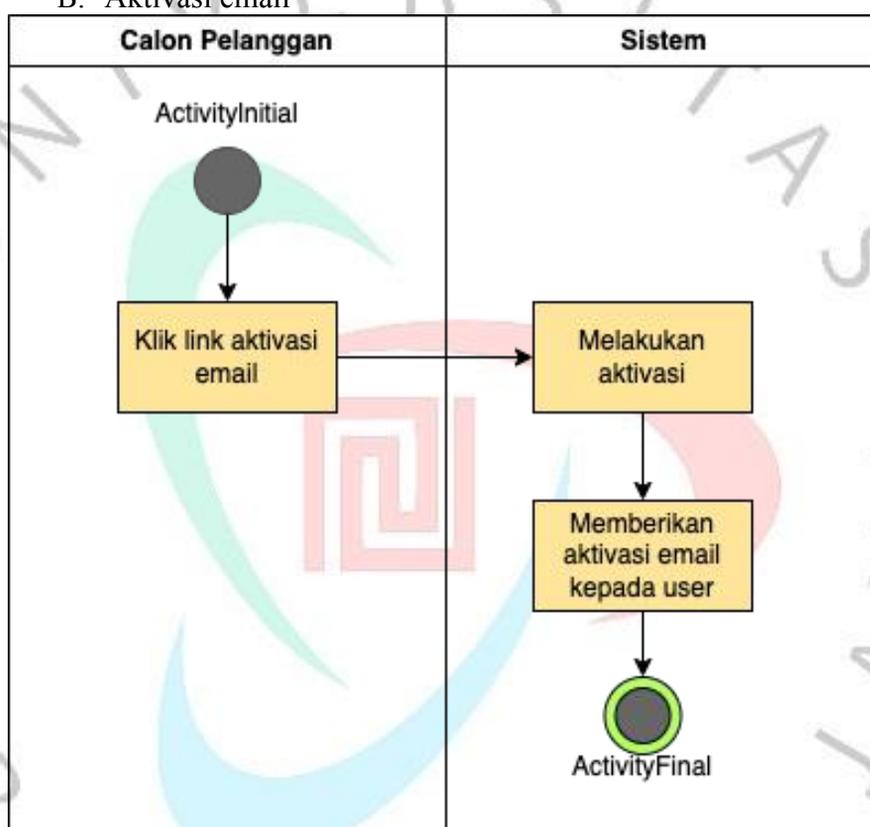
#### A. Registrasi



Gambar 4.2 Activity Diagram Registrasi

Pada **Gambar 4.2** di atas merupakan aktivitas diagram registrasi, yang mana di situ ada pelanggan dan sistem. Di mulai dari tanda status awal masuk ke bagian halaman registrasi setelah itu ada simbol *fork* yang di digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel setelah itu masuk ke bagian input email dan input password, setelah itu ada simbol *join* di gunakan untuk kegiatan yang di gabungkan , setelah itu masuk ke bagian klik button registrasi setelah itu verifikasi email masuk ke sistem jika email tersedia link aktifasi masuk ke email setelah itu masuk ke sistem tautan kedaluwarsa kalo ya klik button registrasi kalo tidak masuk ke menu login.

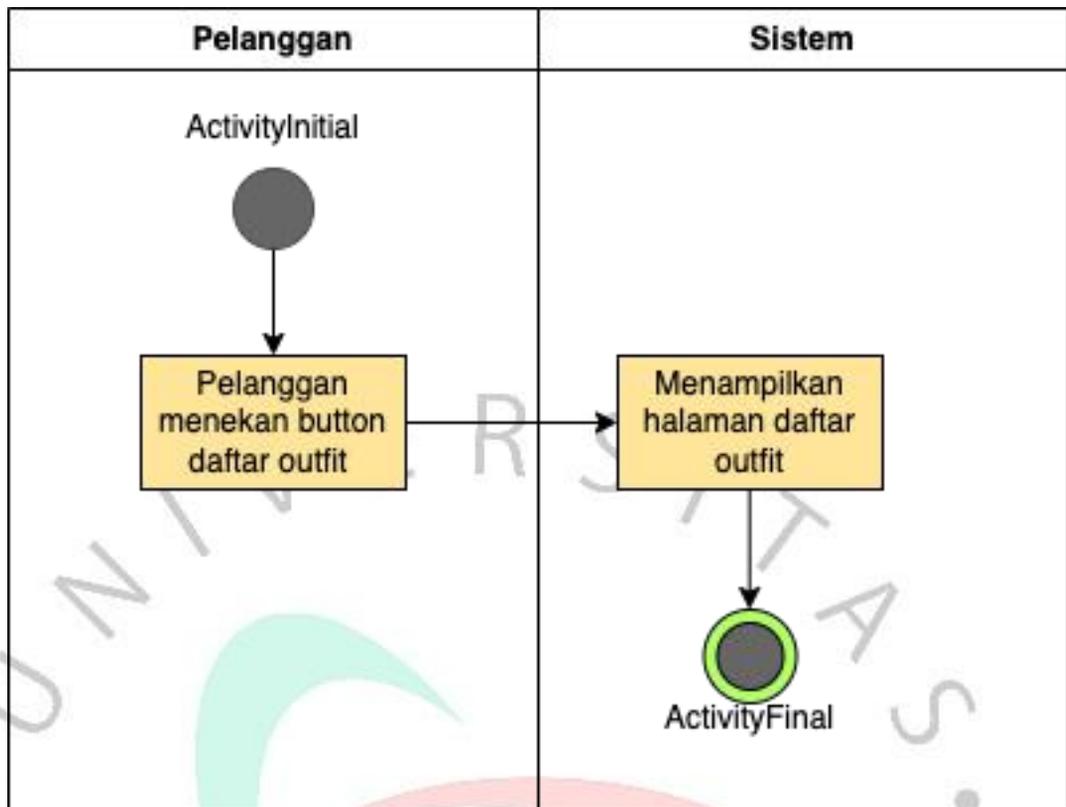
B. Aktivasi email



Gambar 4.3 Activity Diagram Aktivasi Email

Pada **Gambar 4.3** Aktivasi email calon pelanggan mengklik aktivasi email setelah itu sistem melakukan aktivasi untuk diberikan kepada user. Pada umumnya dilakukan dengan cara pengguna menerima email yang berisi tautan atau kode verifikasi.

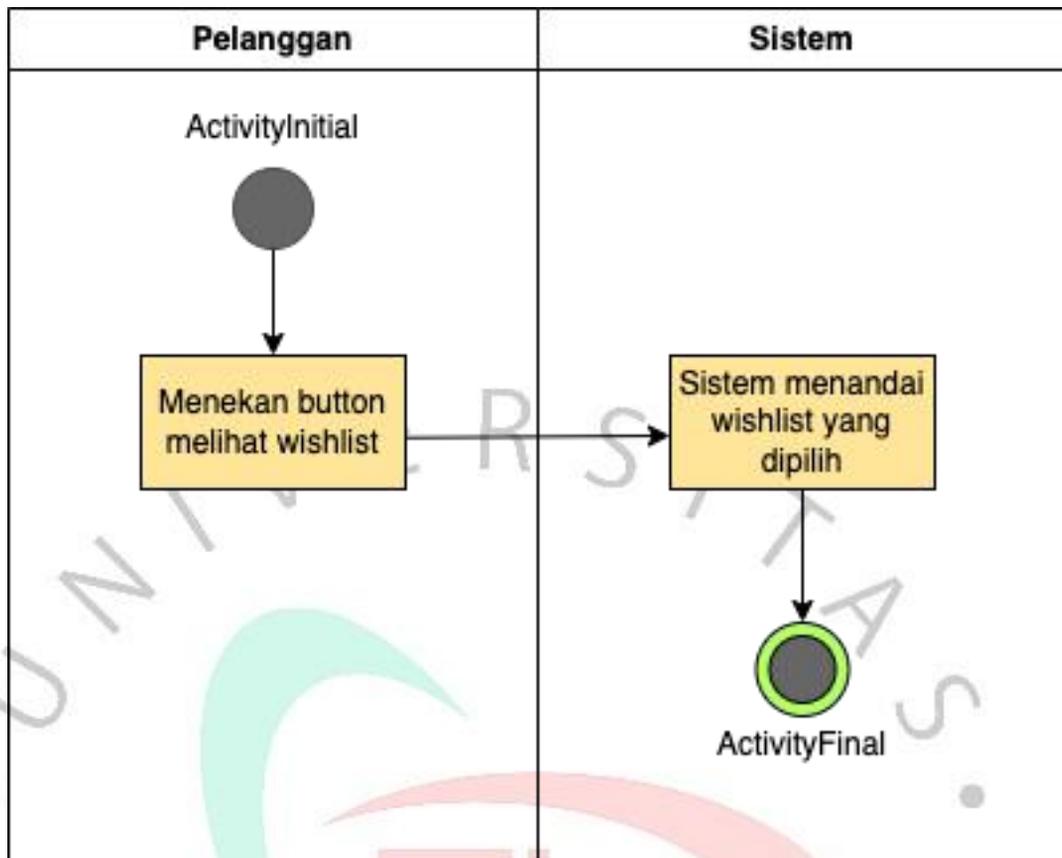
C. Melihat daftar outfit



Gambar 4.4 Activity Diagram Melihat daftar outfit

Pada **Gambar 4.4** Melihat Daftar Outfit, pelanggan menekan button daftar outfit lalu sistem menampilkan halaman daftar outfit. Dengan adanya daftar outfit membantu pelanggan mengeksplorasi gaya baru dan melihat berbagai macam outfit yang tersedia.

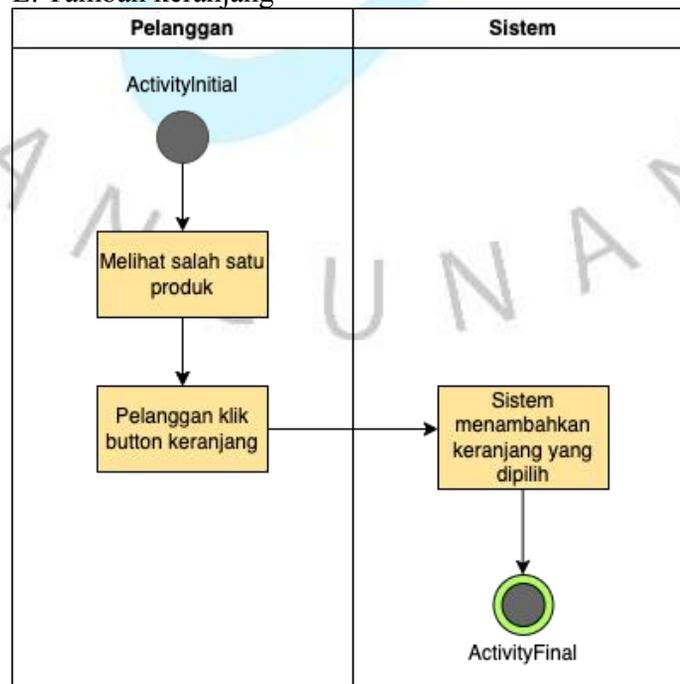
#### D. Melihat wishlist



Gambar 4.5 Activity Diagram melihat wishlist

Pada **Gambar 4.5** melihat wishlist pelanggan meneka buton melihat wishlist atau barang yang di favoritkan setelah itu sistem menandai wishlist yang dipilih. Biasanya terjadi pada daftar barang yang diinginkan seseorang untuk dibeli atau yang difavoritkan oleh user.

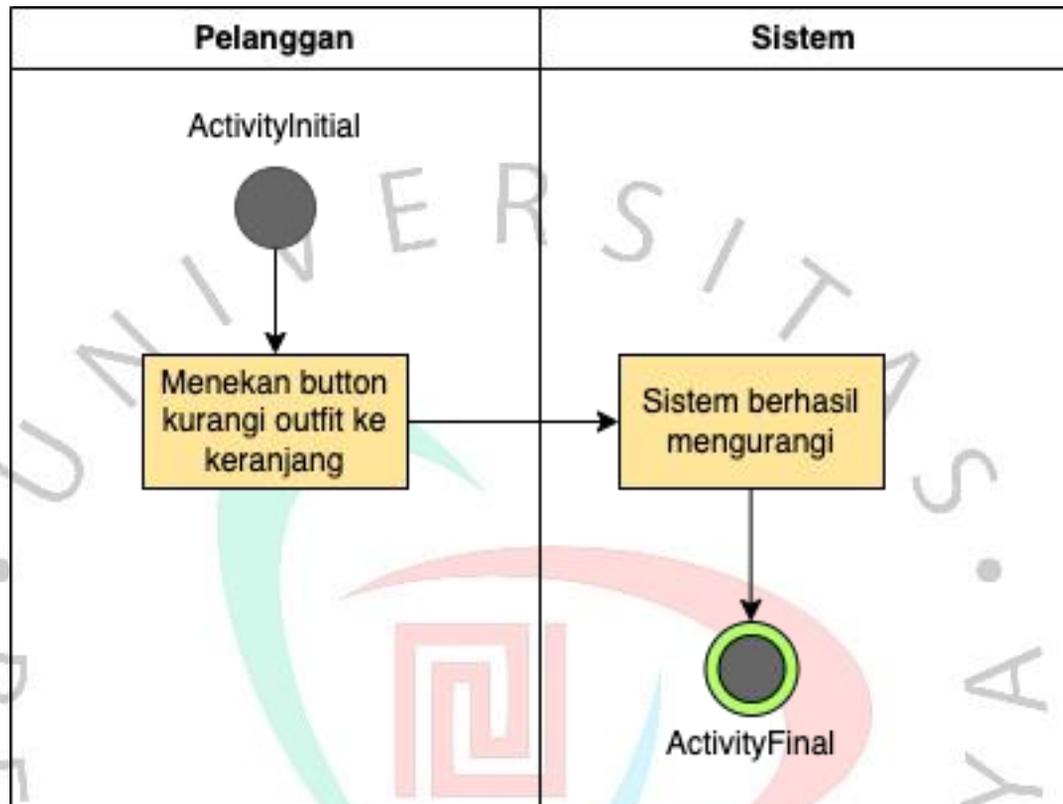
#### E. Tambah keranjang



Gambar 4.5 Activity Diagram tambah keranjang

Pada **Gambar 4.6** tambah keranjang, pelanggan melihat salah satu produk setelah itu pelanggan klik button keranjang lalu sistem menambahkan keranjang yang dipilih. Tambah keranjang adalah suatu tindakan menambahkan barang ke dalam keranjang belanja saat belanja online.

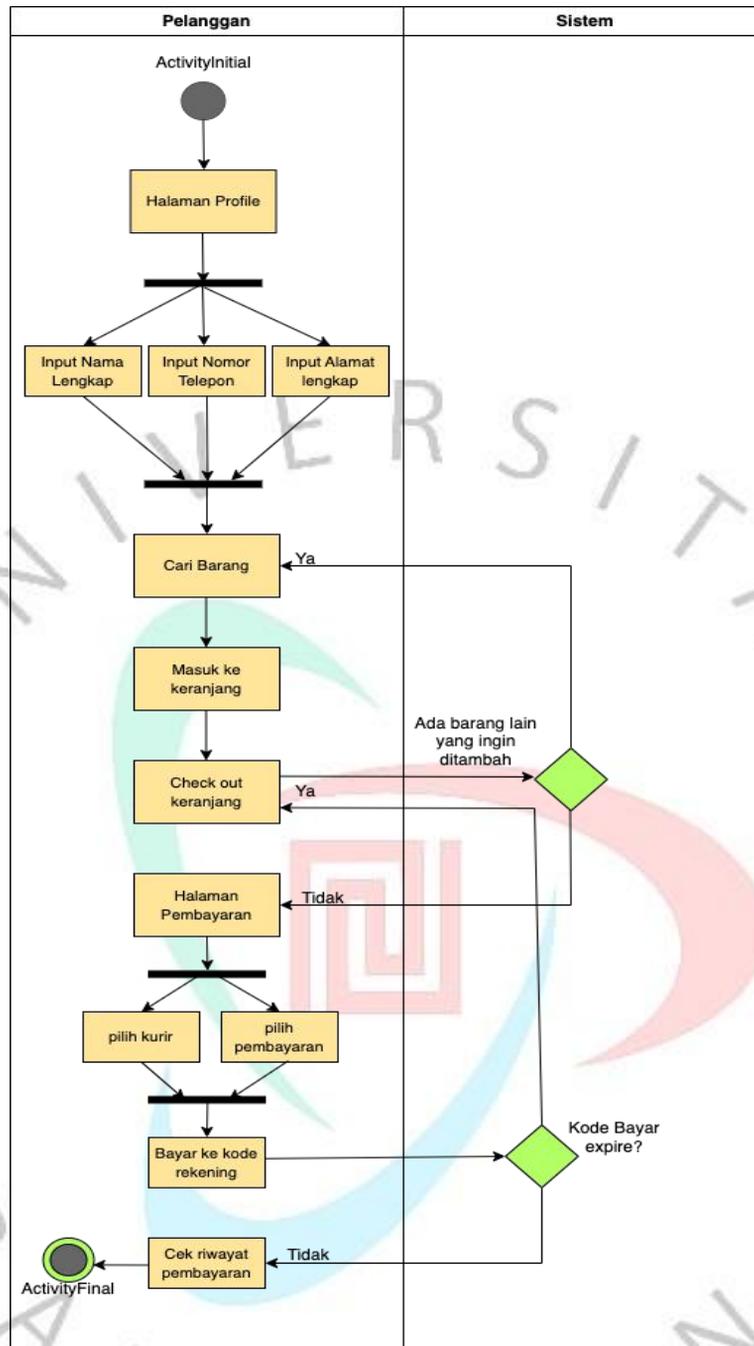
F. Kurangi keranjang



Gambar 4.6 Activity Diagram tambah keranjang

Pada **gambar 4.6** kurangi outfit pelanggan menekan button kurangi outfit ke keranjang setelah sistem berhasil menguranginya. Yang dimana dilakukan sebuah pengurangan jumlah barang yang dibeli oleh *user* dalam satu transaksi sebelum terjadi checkout.

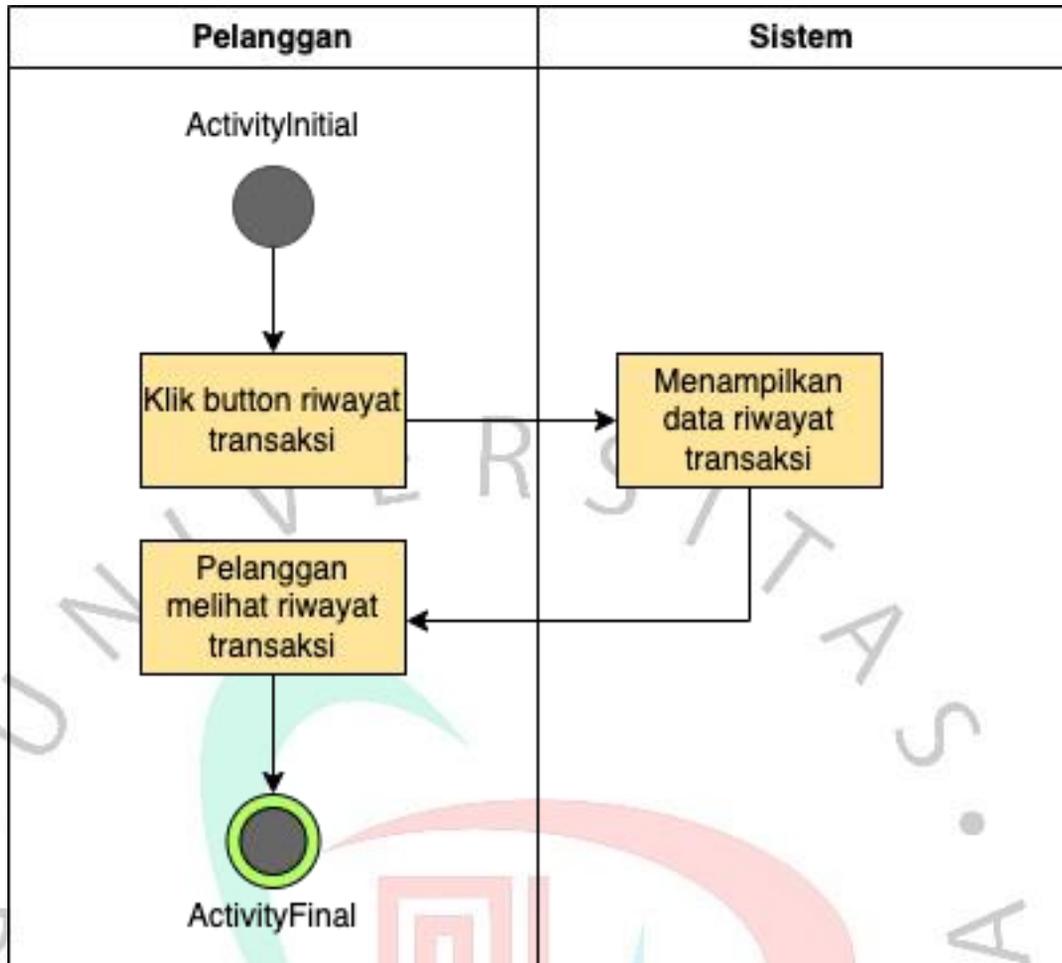
## F. Checkout



Gambar 4.7 Activity Diagram Checkout

Pada gambar 4.7 di atas pelanggan masuk ke halaman profile setelah itu menginput nama lengkap, nomor telepon dan alamat lengkap masuk ke cari barang masuk ke keranjang, jika ada barang lain ingin ditambah masuk ke cari barang jika tidak lanjut ke hal pembayaran setelah itu pilih kurir, dan pilih pembayaran lalu bayar ke kode rekening jika kode *expire* akan di arahkan ke chekout keranjang dan jika tidak bisa cek riwayat pembayaran.

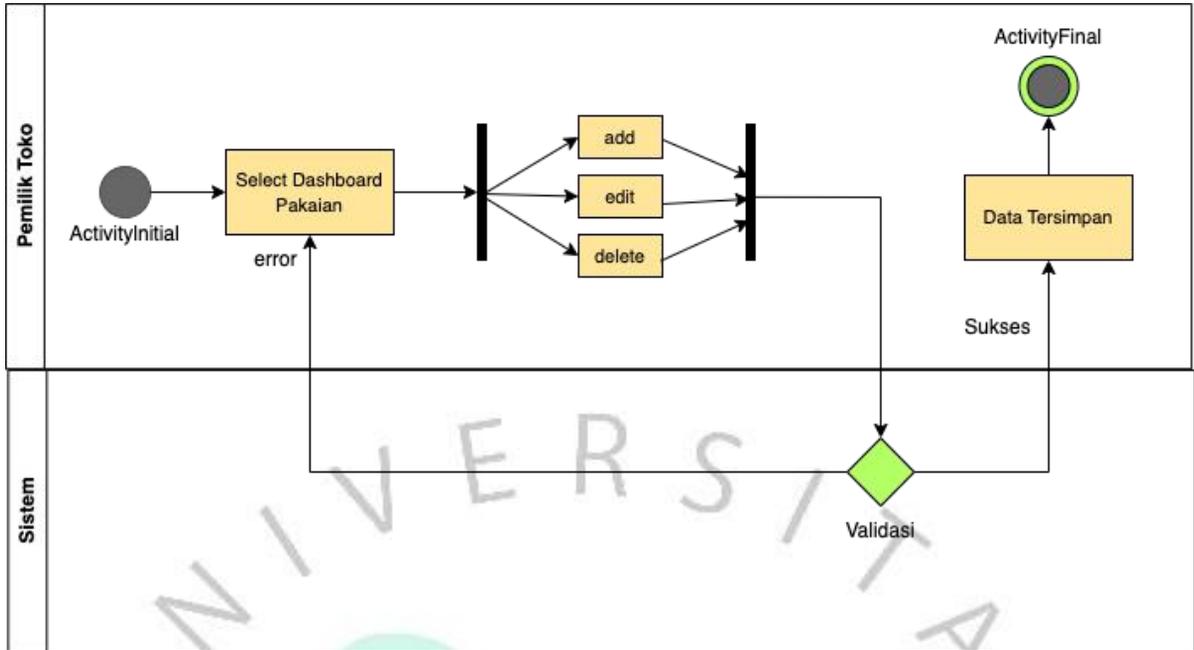
F. Melihat riwayat transaksi



Gambar 4.8 Activity Diagram Melihat riwayat transaksi

Pada gambar 4.8 melihat riwayat transaksi, pelanggan klik button riwayat transaksi setelah itu user melihat riwayat transaksi. User melihat riwayat transaksi. User melihat riwayat transaksi untuk membantu user dalam melihat deskripsi, harga total, jumlah dan tanggal atau waktu transaksi.

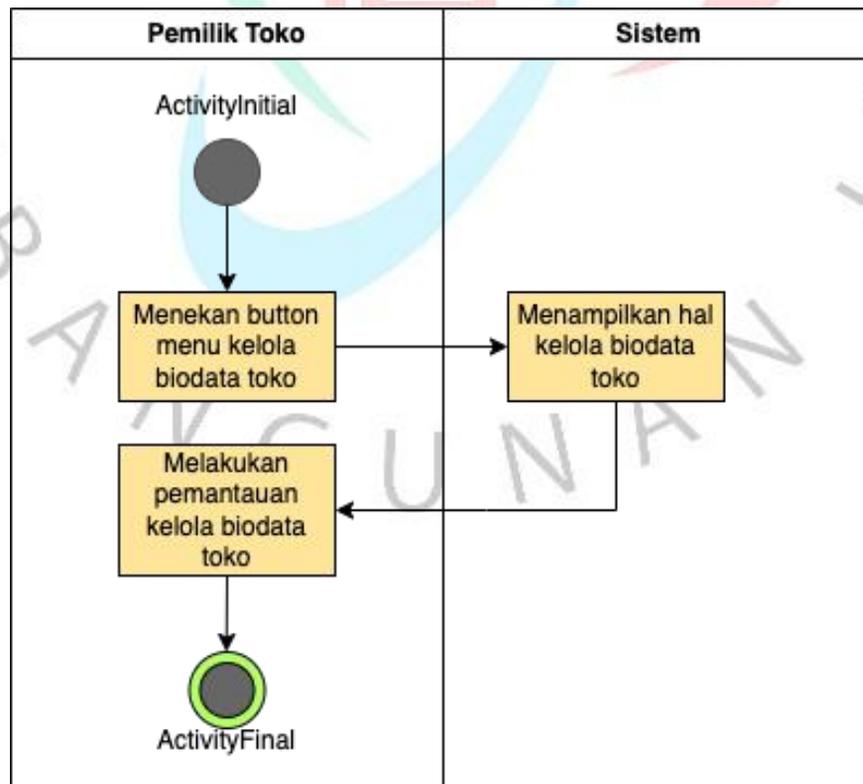
## F. Kelola Outfit



Gambar 4.9 Activity Diagram kelola outfit

Pada gambar 4.9 Kelola outfit, Pemilik toko select dashboard pakaian setelah itu pemilik toko dapat melakukan *add, edit* dan *delete*, sistem memvalidasi jika terjadi error akan di arahkan ke halaman select dashboard pakaian dan jika sukses data tersimpan.

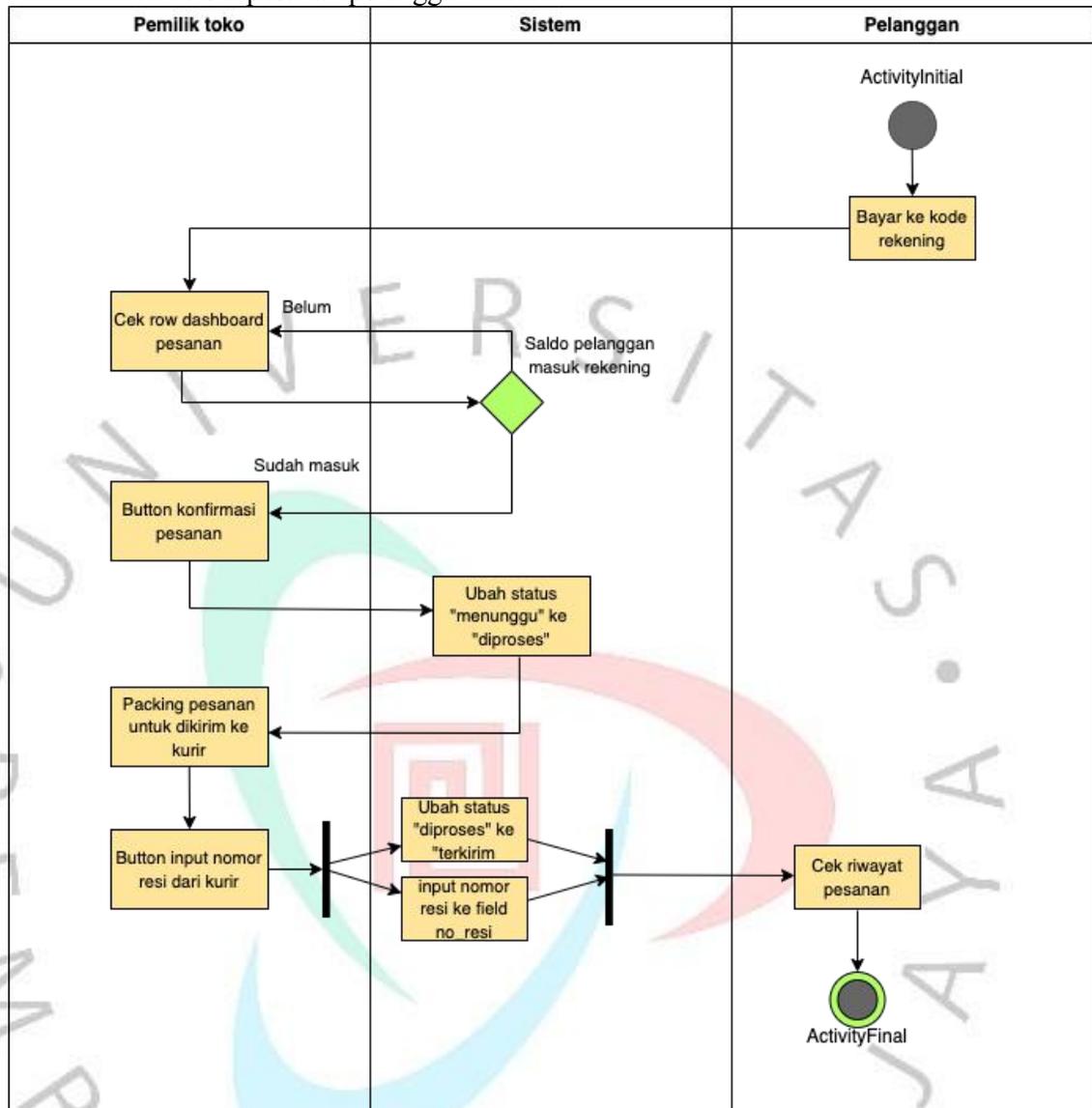
## F. Kelola biodata toko



Gambar 4.10 Activity Diagram Kelola biodata toko

Pada **gambar 4.10** Kelola biodata toko, Pemilik toko menekan button menu kelola biodata toko setelah itu sistem menampilkan halaman kelola biodata toko untuk bisa pemilik toko melakukan pemantauan kelola biodata toko.

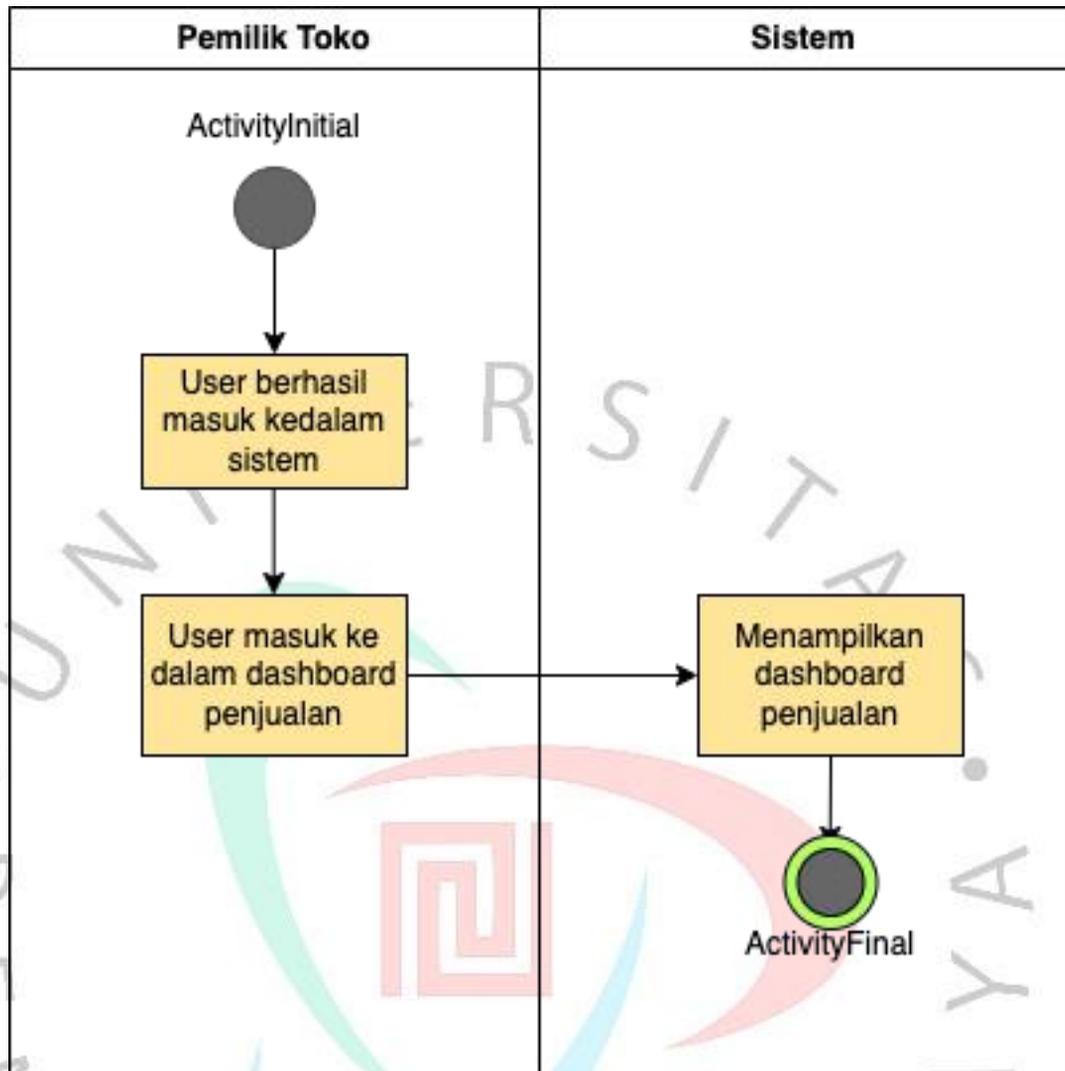
F. Kelola pesanan pelanggan



Gambar 4.11 Activity Diagram kelola pesanan pelanggan

Pada **gambar 4.11** kelola pesanan pelanggan, pelanggan bayar ke kode rekening lalu pemilik toko cek row dashboard pesanan. Sistem jika saldo pelanggan belum masuk rekening akan diarahkan halaman cek row dashboard pesanan jika sudah masuk user klik button konfirmasi pesanan setelah itu sistem ubah status “menunggu” ke “diproses” masuk ke pemilik toko akan packing pesanan untuk dikirim ke kurir lalu klik button input nomor resi dari kurir. Sistem, ubah status “diproses” ke “terkirim” dan input nomor resi ke field no\_resi setelah itu pelanggan bisa cek riwayat pesanan.

F. List dashboard penjualan



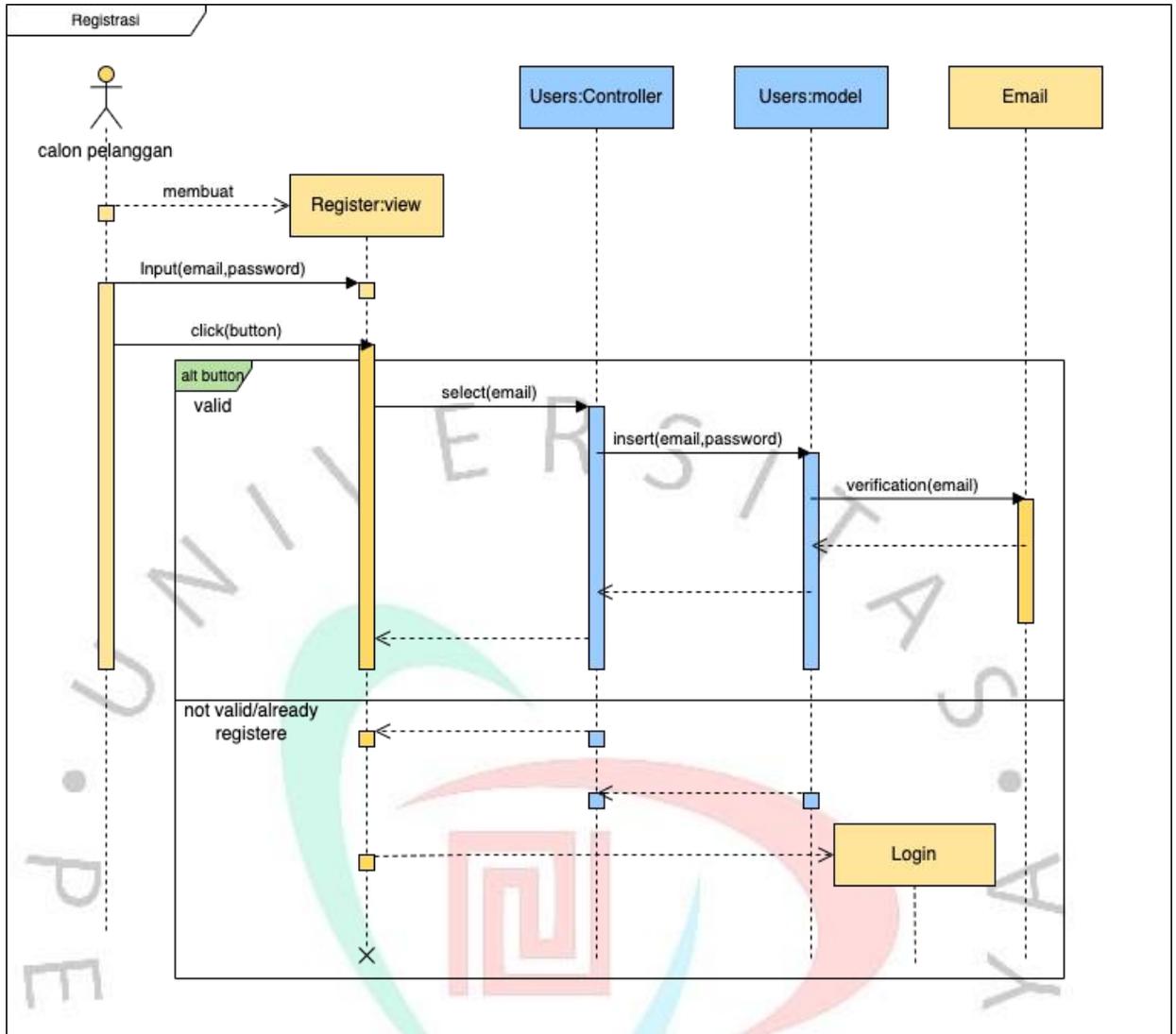
Gambar 4.12 Activity Diagram list dashboard penjualan

Pada gambar 4.12 list dashboard penjualan, Pemilik toko berhasil masuk ke dalam sistem, setelah itu pemilik toko masuk ke dalam dashboard penjualan. Sistem akan menampilkan dashboard penjualan

#### 4.2.4 Sequence Diagram

*Sequence diagram* menjelaskan suatu gambaran bagaimana aktor bisa berhubungan dengan objek pada use case diagram serta terdapat pesan pada setiap relasi tersebut. *Sequence diagram* juga menampilkan hubungan interaksi antar objek dalam suatu sistem. Perancangan *sequence diagram* menggambarkan alur *generic* yang memungkinkan disetiap skenario terdapat *use case*. *Sequence diagram* juga adalah alat model yang digunakan oleh para ahli di bidang pengembangan perangkat lunak untuk menggambarkan dan memahami interaksi antar objek dalam sistem untuk memberikan gambaran visual tentang bagaimana objek-objek berkomunikasi satu sama lain dalam suatu skenario atau alur kerja tertentu. Para ahli menggunakan *sequence diagram* untuk menyajikan secara jelas dan terstruktur bagaimana objek-objek berkomunikasi dalam suatu sistem untuk mengidentifikasi dan merancang interaksi antara objek-objek untuk membantu dalam memahami urutan kejadian dan *sequence diagram* adalah bagian dari *Unified Modeling language(uml)*. Berikut adalah rancangan *sequence diagram* pada pembelian barang *second branded*.

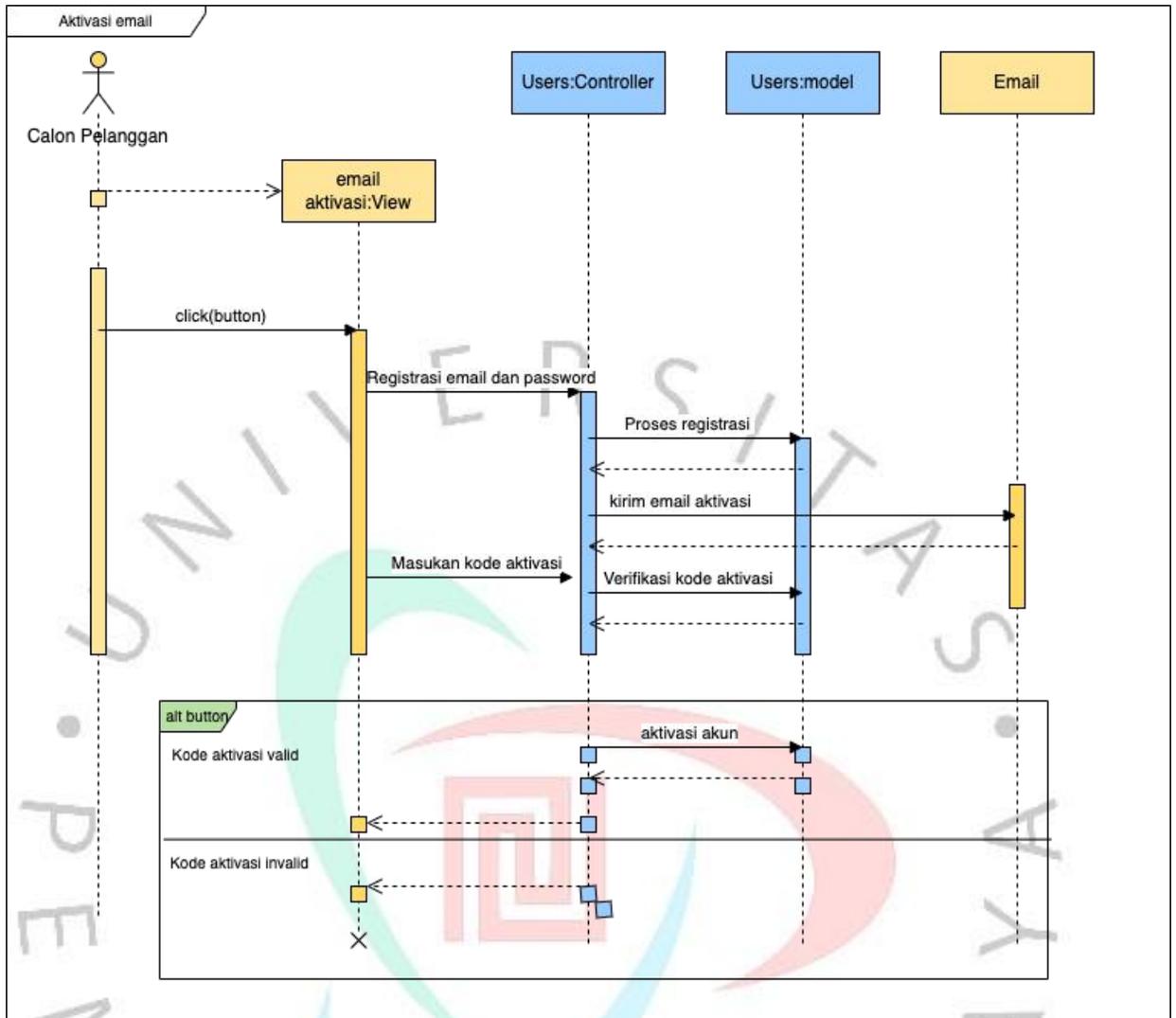
## A. Registrasi



Gambar 4.13 Sequence Diagram Register

Pada **Gambar 4.13** di atas adalah *sequence diagram* Registrasi pengguna membuka tampilan registrasi dan mengisi formulir dengan email dan password, tampilan registrasi mengirimkan data *email* dan *password* ke *user controller*. *User controller* menerima data dari tampilan registrasi dan memutuskan operasi yang harus dieksekusi. *User controller* memilih operasi registrasi dan meneruskan data ke *user model*, *User model* menerima data registrasi dari *user controller* dan memeriksa kevalidan *email* dan *password*, Jika data valid, *user model* mengembalikan hasil ke *user controller* dan pengguna diarahkan ke tampilan beranda( *menu view home*). Jika data tidak valid, *User Model* mengembalikan hasil ke *user controller* dan pengguna diarahkan kembali ke tampilan *login(login view)*.

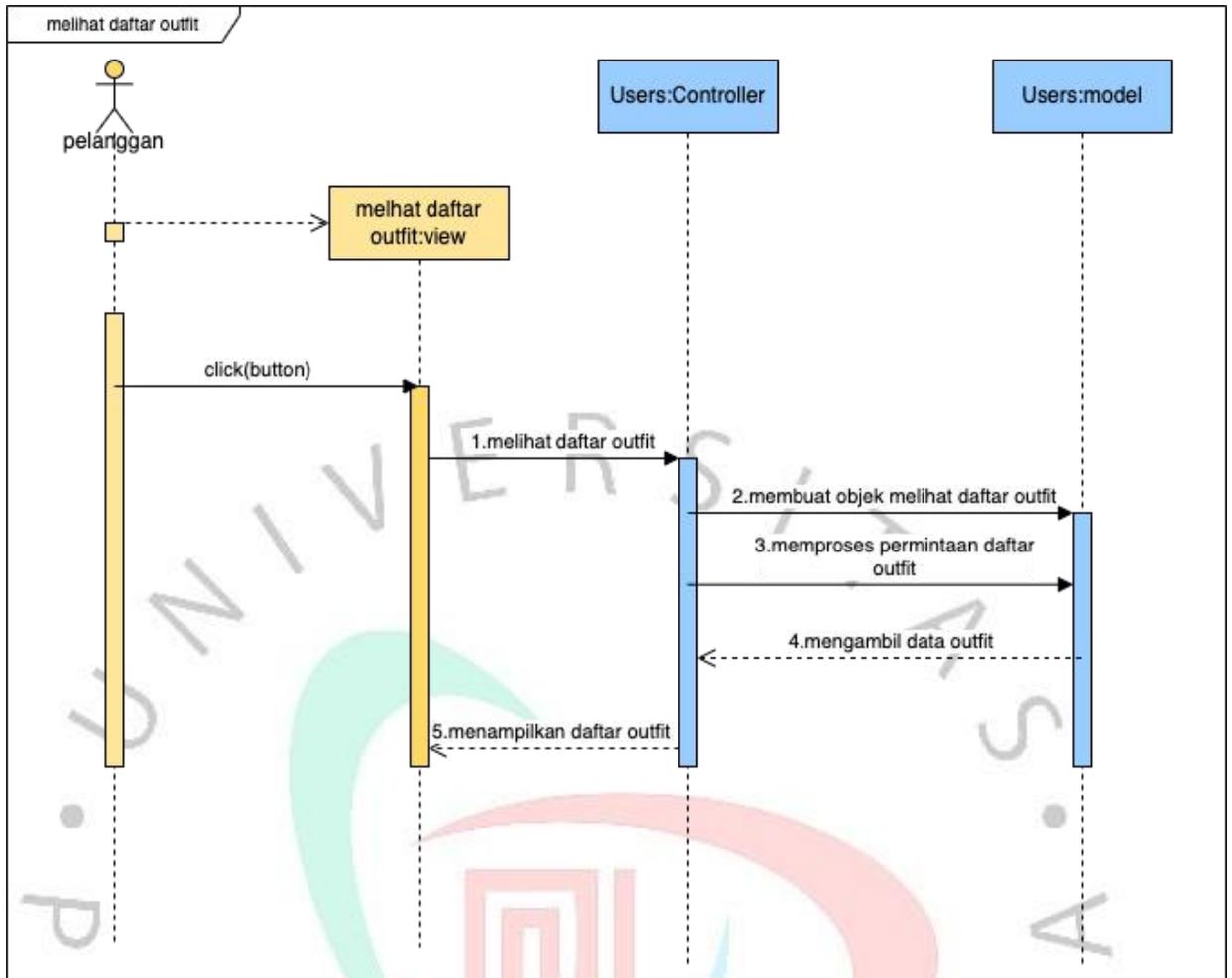
## B. Aktivasi email



Gambar 4.14 Sequence Diagram aktivasi email

Pada **Gambar 4.14** calon pelanggan registrasi dengan email dan password setelah itu masuk ke proses registrasi(*create user*) dan user berhasil dibuat kirim email aktivasi(*user,email*) dan email aktivasi terkirim setelah itu masukan kode aktivasi, verifikasi kode aktivasi dan di pastikan kode valid/invalid. Dan terakhir jika valid aktivasi akun dan akun diaktifkan berhasil login dan jika tidak valid aktivasi gagal.

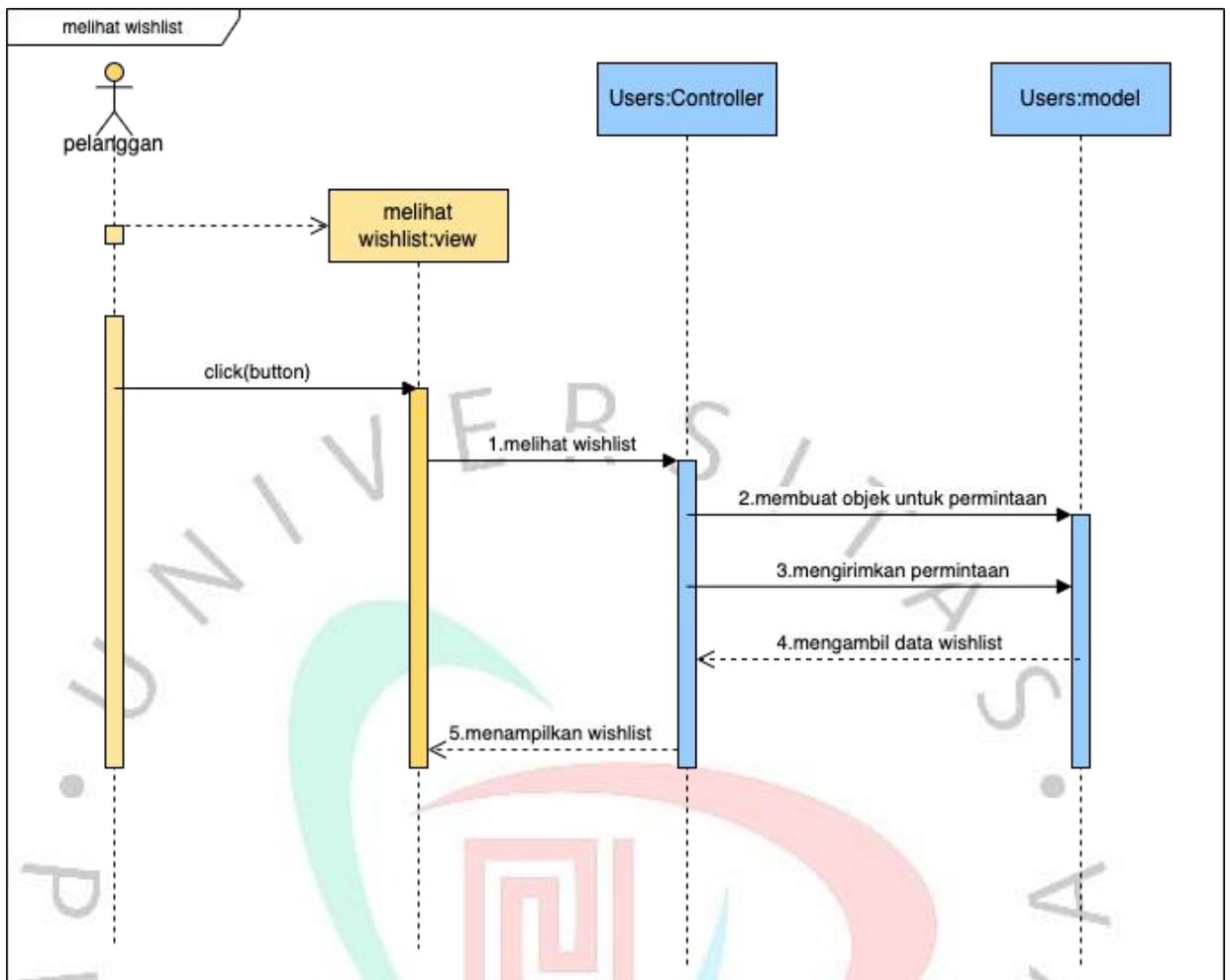
### C. Melihat daftar outfit



Gambar 4.15 *Sequence Diagram* melihat daftar outfit

Pada **gambar 4.15** pelanggan klik button melihat daftar outfit setelah itu membuat objek aksi untuk permintaan dan mengirimkan permintaan untuk daftar outfit setelah itu dilakukan *return message* mengambil dan memproses data outfit setelah berhasil mengembalikan daftar outfit.

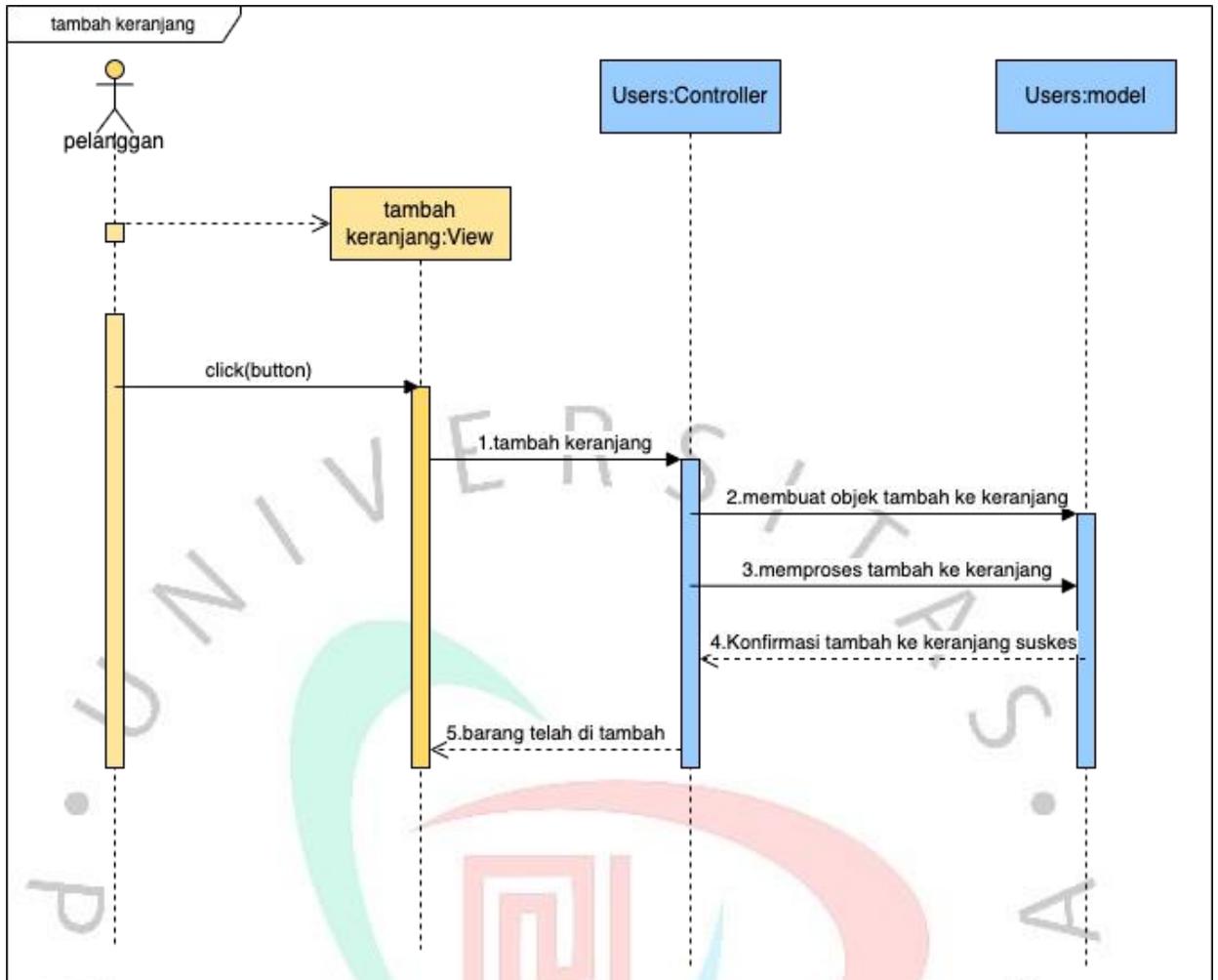
#### D. Melihat wishlist



Gambar 4.16 Sequence Diagram melihat wishlist

Pada gambar 4.16 pelanggan klik button melihat wishlist setelah itu membuat objek aksi untuk permintaan dan mengirimkan permintaan untuk melihat wishlist setelah itu dilakukan *return message* mengambil dan memproses data wishlist setelah berhasil mengembalikan lihat wislist.

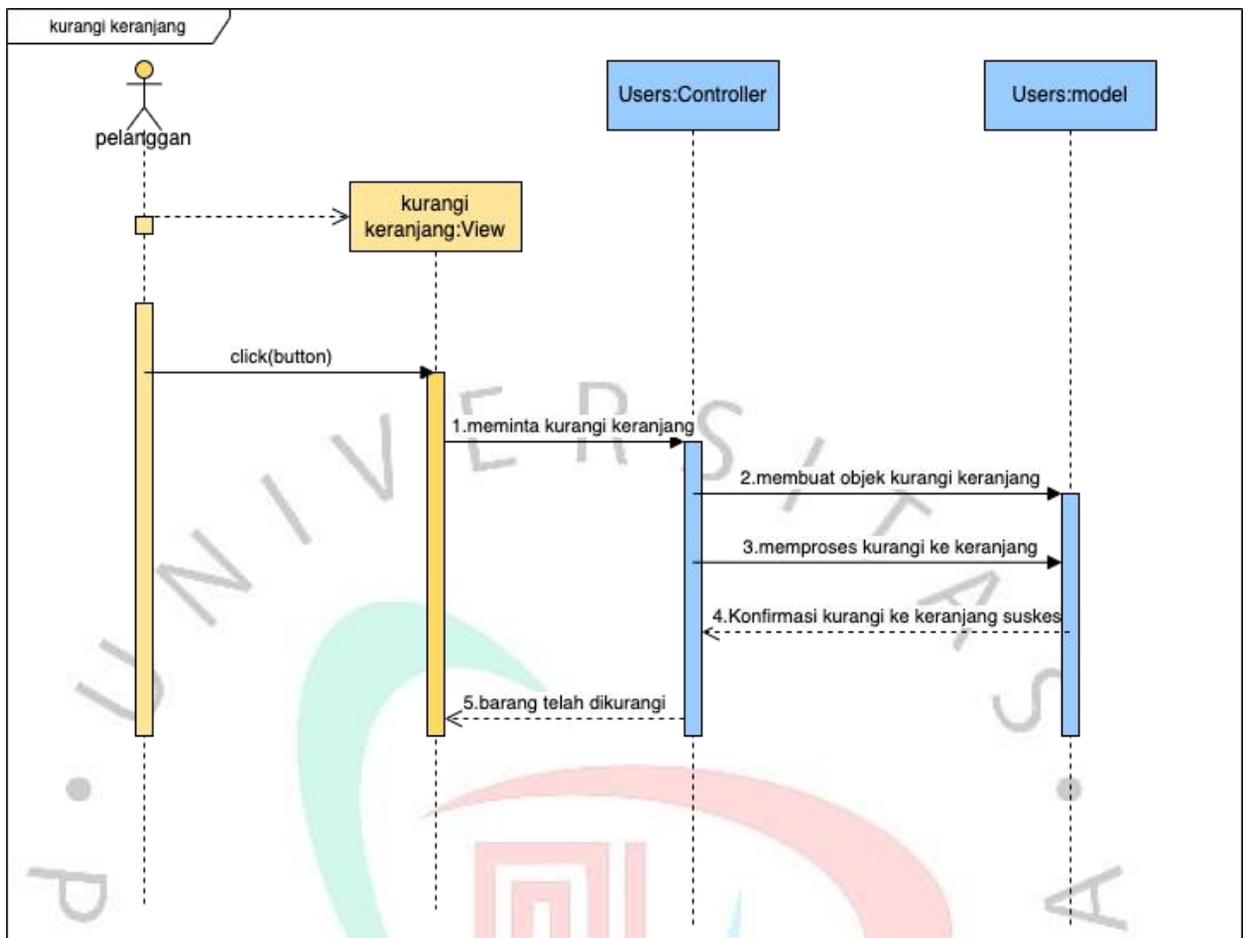
## E. Tambah keranjang



Gambar 4.17 Sequence Diagram tambah keranjang

Pada gambar 4.17 pelanggan meminta tambah keranjang setelah itu masuk membuat objek tambah ke keranjang setelah itu memproses tambah ke keranjang dan berhasil konfirmasi tambah ke keranjang dan terakhir berhasil memberi tahu pelanggan bahwa barang telah di tambahkan ke keranjang

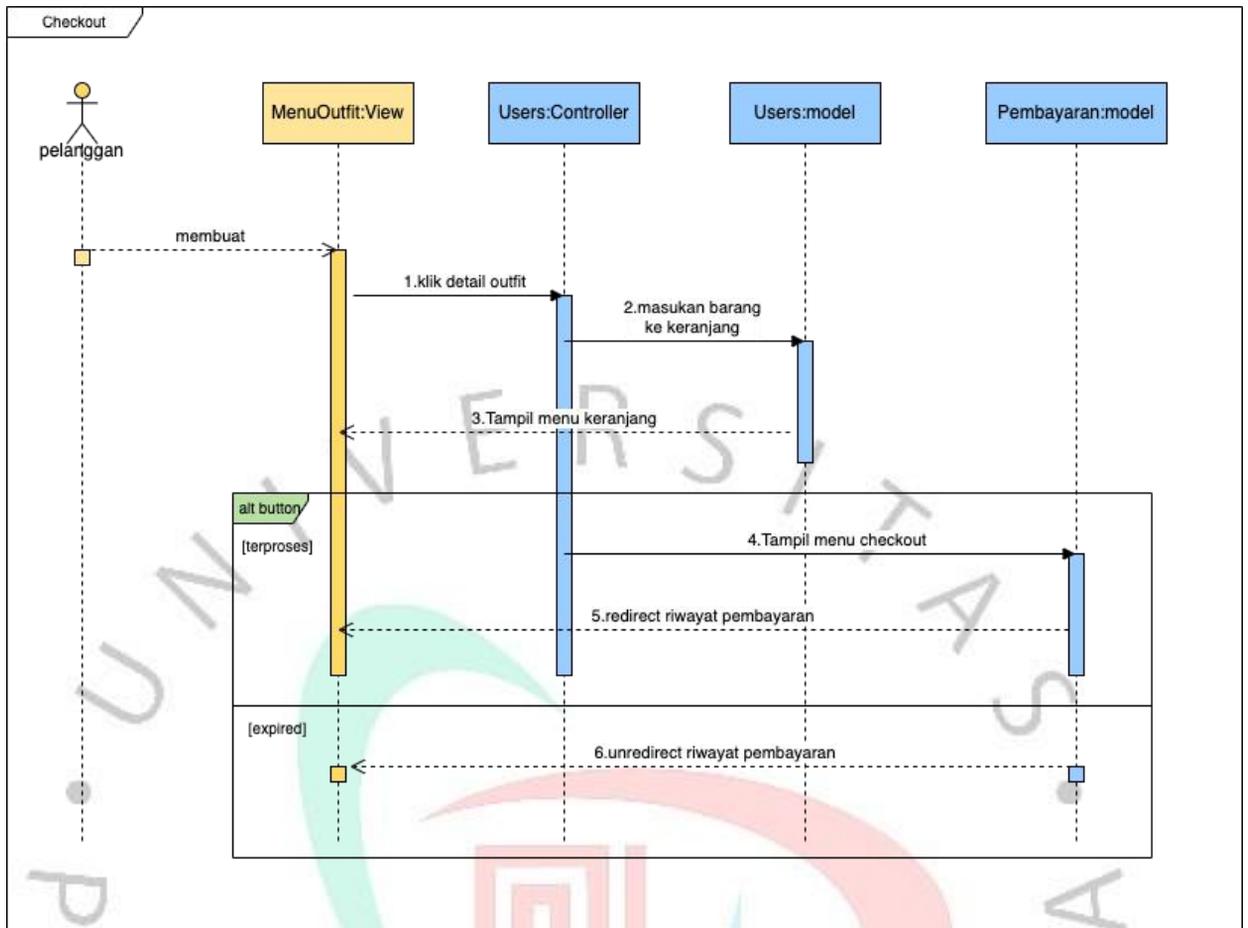
## E. Kurangi keranjang



Gambar 4.18 *Sequence Diagram* kurangi keranjang

Pada **gambar 4.18** pelanggan meminta kurangi dari keranjang setelah itu membuat objek kurangi keranjang setelah itu memproses kurangi dari keranjang dan konfirmasi kurangi dari keranjang dan pelanggan berhasil mengurangi barang dari keranjang.

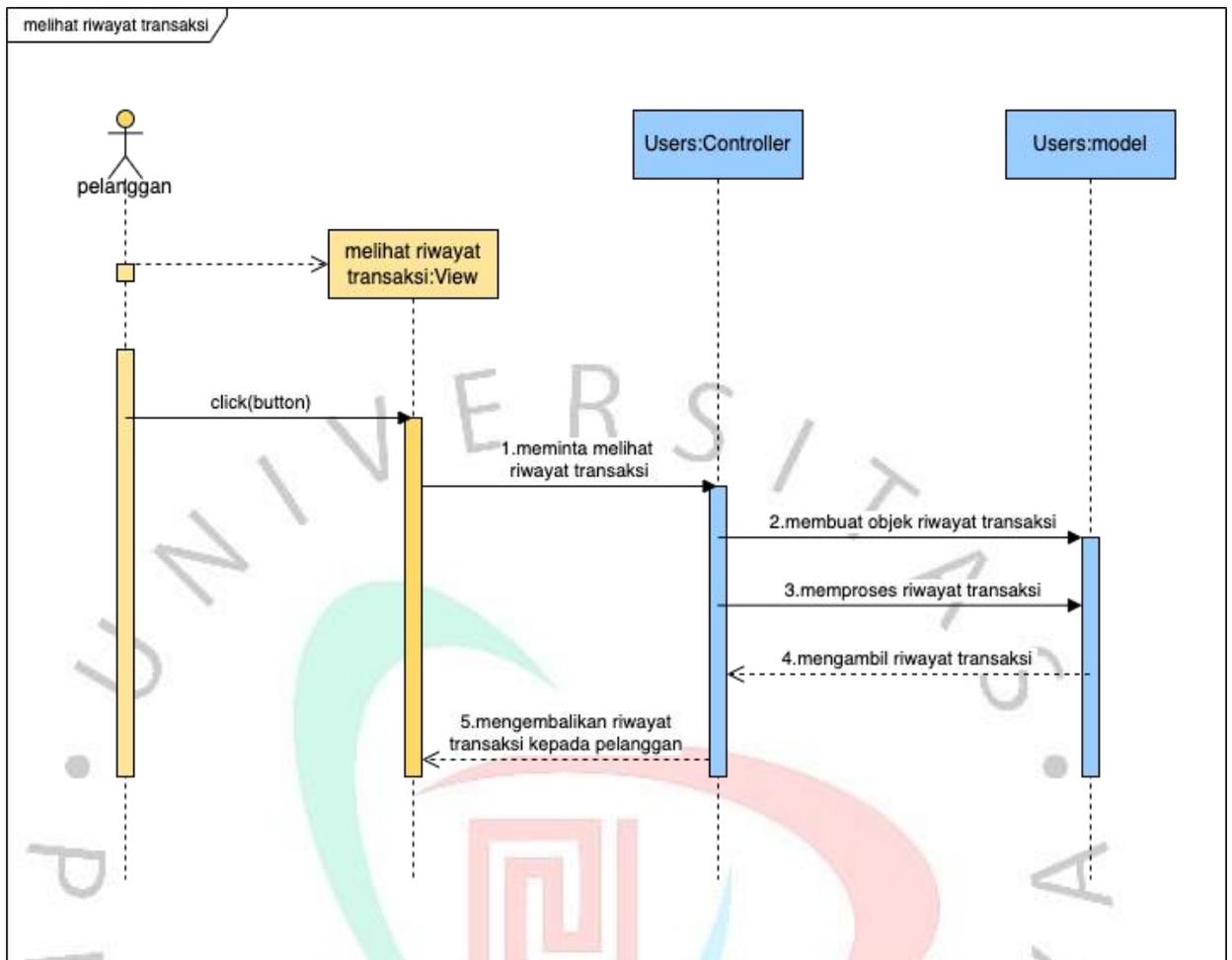
## F. Checkout



Gambar 4.19 Sequence Diagram Checkout

Pada gambar 4.19 di atas sequence diagram checkout, pelanggan klik detail outfit setelah itu masukan barang ke keranjang setelah itu tampil menu keranjang jika sistem nya terproses masuk ke tampilan menu checkout dan akan ke *direct* ke riwayat pembayaran jika sistemnya kadaluarsa aka ke *direct* ke riwayat pembayaran.

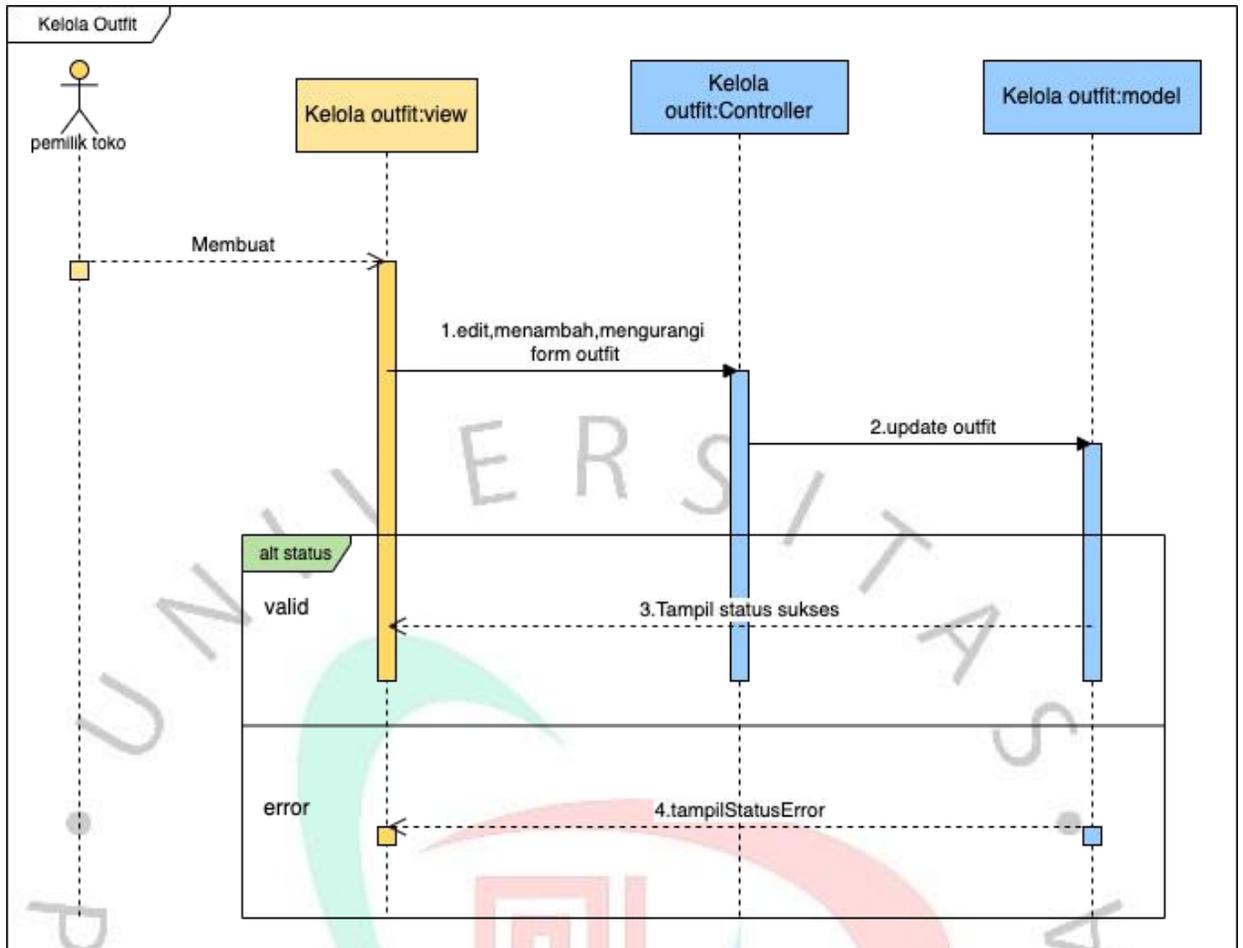
## G. Melihat riwayat transaksi



Gambar 4.20 Sequence Diagram melihat riwayat transaksi

Pada **gambar 4.20** gambar sequence diagram melihat riwayat transaksi, pelanggan klik button riwayat transaksi dari situ bagian *controller* meminta melihat riwayat transaksi setelah itu kirim pesan ke bagian model untuk membuat objek riwayat transaksi dan memproses riwayat transaksi setelah itu pesan kembali mengambil riwayat transaksi dan mengembalikan riwayat transaksi kepada pelanggan.

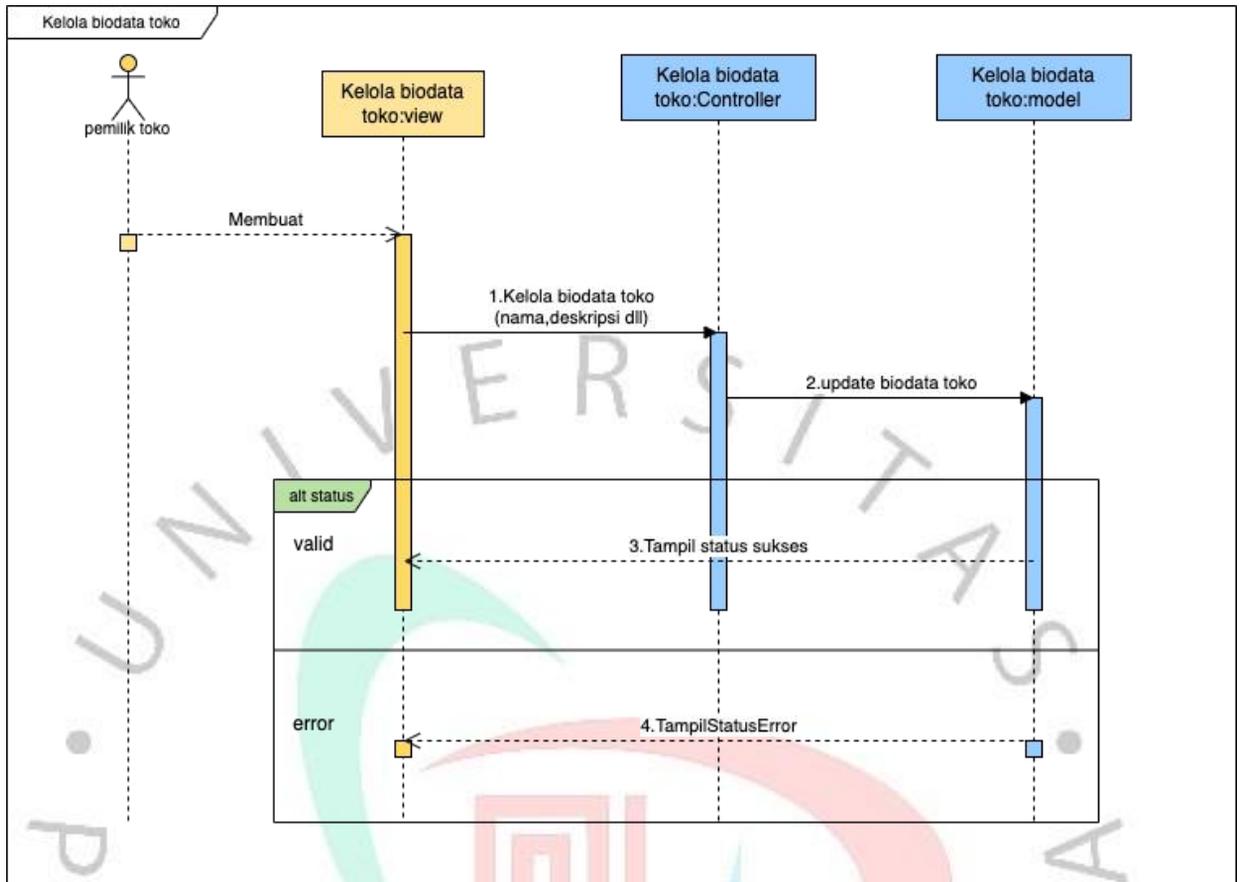
## H. Kelola outfit



Gambar 4.21 Sequence Diagram kelola outfit

Pada gambar 4.21 sequence diagram kelola outfit, pemilik toko kelola outfit mengirim pesan ke bagian *controller* untuk edit,menambah,mengurangi form outfit setelah itu controller mengirim pesan ke model untuk update outfit jika valid tampil status sukses jika tidak tampil status error.

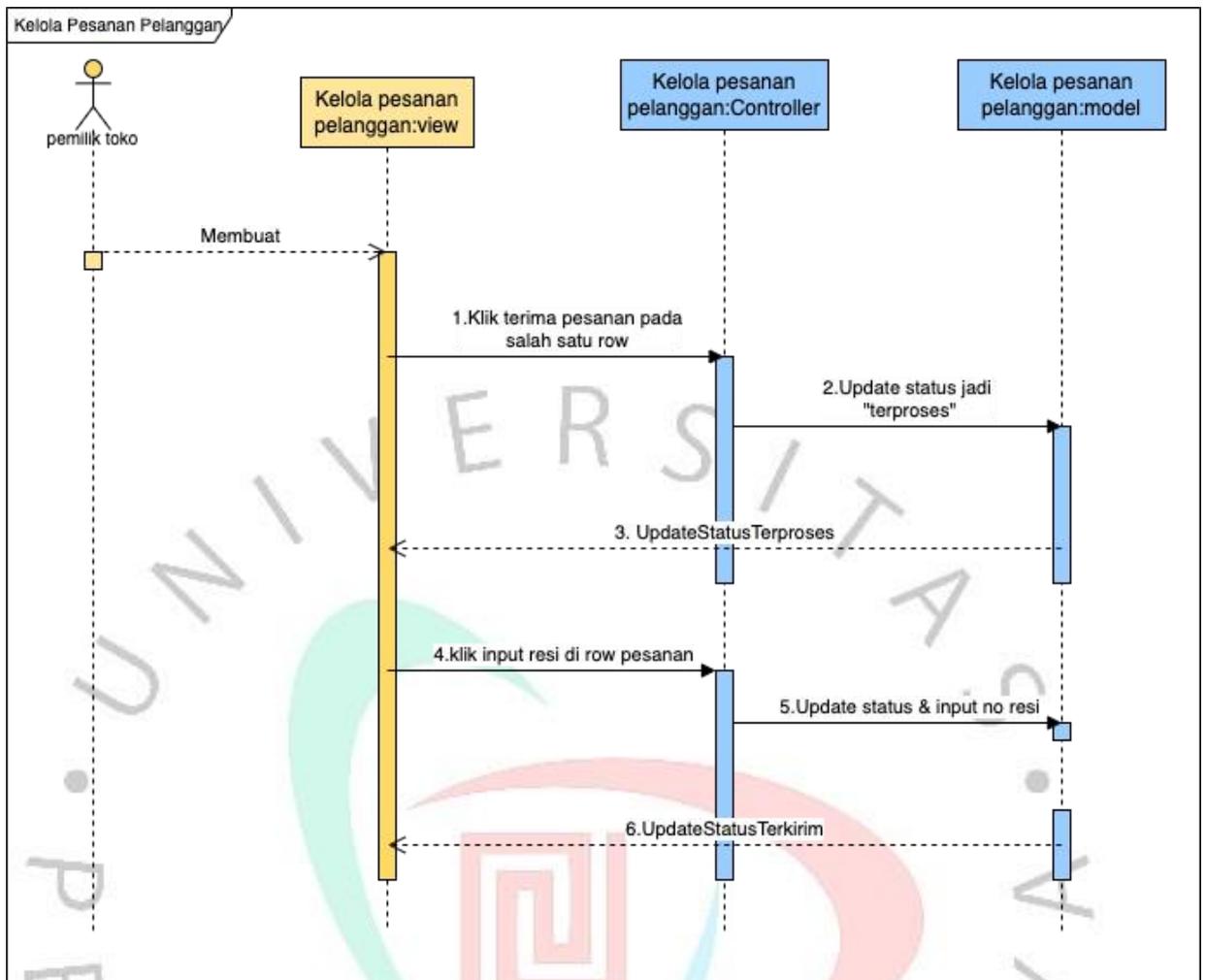
## I. Kelola biodata toko



Gambar 4.22 Sequence Diagram kelola biodata toko

Pada **gambar 4.22** sequence diagram kelola biodata toko, pemilik toko mengirim pesan ke bagian *controller* untuk kelola biodata toko (nama,deskripsi dan lain-lain) setelah itu dari bagian *controller* mengirim pesan ke model untuk *update* biodata toko jika valid tampil status sukses dan jika tidak tampil status error

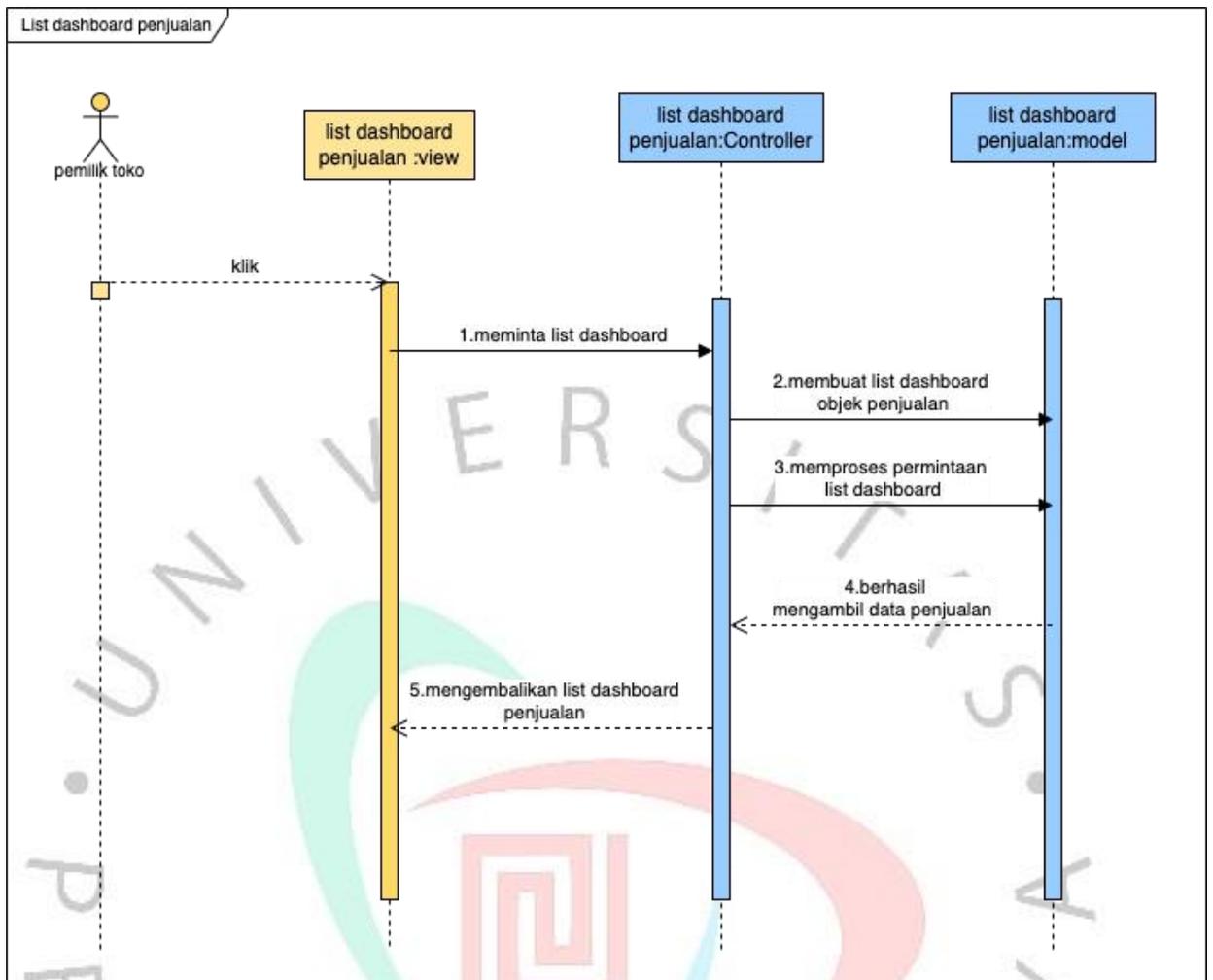
## J. Kelola pesanan pelanggan



Gambar 4.23 Sequence Diagram kelola pesanan pelanggan

Pada **gambar 4.23** sequence diagram kelola pesanan pelanggan, pemilik toko membuat kelola pesanan pelanggan mengirim pesan klik terima pesanan pada salah satu row ke objek *controller*, setelah itu update status jadi terproses ke objek model dan update status terproses. Klik input resi di row pesanan lalu update status dan input no resi dan update status terkirim.

## K. List dashboard penjualan



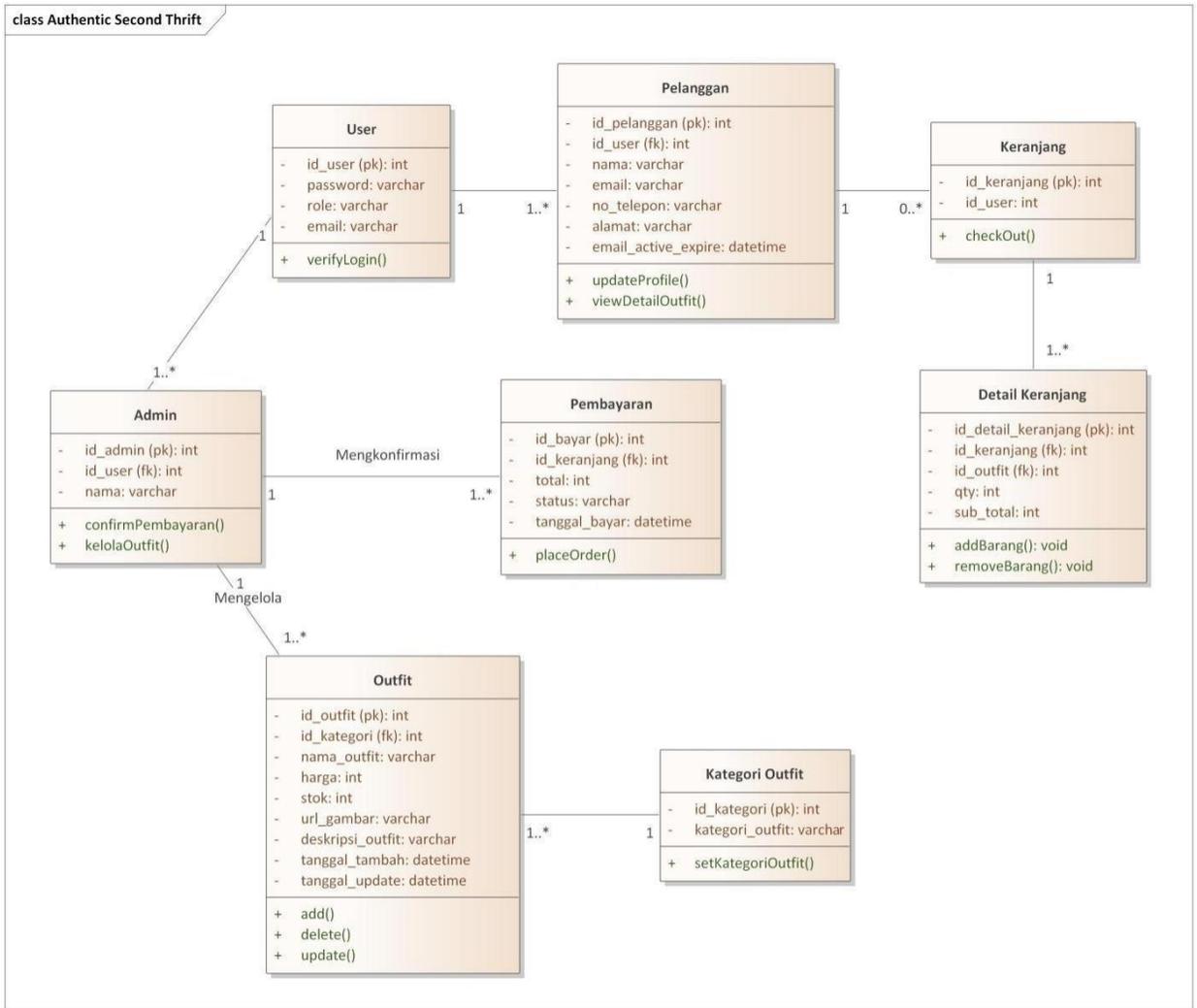
Gambar 4.24 Sequence Diagram list dashboard penjualan

Pada **gambar 4.24** sequence diagram list dashboard penjualan, pemilik toko klik list dashboard setelah itu penjualan meminta list dashboard ke objek controller setelah itu mengirim pesan untuk membuat list dashboard objek penjualan setelah itu memproses permintaan list dashboard dan berhasil mengambil data penjualan dan mengembalika list dashboard penjualan.

#### 4.2.5 Class Diagram

Class diagram adalah salah satu dari pemodelan UML (*Unified Modeling Language*) yang digunakan untuk menggambarkan struktur statis dari suatu sistem. Class diagram ini menunjukkan kelas-kelas dalam sistem, yang menghubungkan antara kelas-kelas tersebut serta atribut dan metode yang dimiliki oleh setiap kelas. Class diagram membantu memvisualisasikan struktur objek dan hubungan di antara mereka dalam suatu sistem.

Berikut adalah struktur kelas-kelas atau suatu objek-objek yang terdapat dalam suatu sistem dan hubungan antar kelas tersebut. Diagram ini memberikan gambaran tentang struktur data dalam suatu sistem dan bagaimana data tersebut diolah dan disimpan. Berikut ini adalah hasil *class diagram* sistem transaksi pemesanan di toko *authentic.secondthrift* yang bisa dilihat pada gambar 4.25 dibawah ini :



Gambar 4.25 Class diagram sistem informasi Authentic.secondthrift

#### 4.2.6 Specification table

Berikut adalah spesifikasi table pada toko *authentic.secondthrift*, diantaranya :

##### 4.2.6.1 User

Tabel : User

Primary Key : id\_user

Foreign Key : -

Tabel 4.13 *Specification user*

Nomor	Field	Length	Tipe Data	Keterangan
1	id_user	11	int	Primary Key
3	email	50	varchar	Email pengguna
4	password	255	varchar	Password pengguna
5	role	11	varchar	Role pengguna: P = Pelanggan, A = Admin

#### 4.2.6.2 Pelanggan

Tabel : Pelanggan

Primary Key : id\_pelanggan

Foreign Key : id\_user

Tabel 4.14 *Specification pelanggan*

Nomor	Field	Length	Tipe Data	Keterangan
1	id_pelanggan	11	int	Primary Key
2	id_user	11	int	Foreign Key
3	Nama	50	varchar	Nama pengguna
4	no_telepon	50	varchar	Email pengguna
5	Alamat	255	varchar	Alamat pengguna
6	email_active_expire	datetime	datetime	Mengecek apakah email sudah aktif / expire

#### 4.2.6.3 Admin

Tabel : Admin

Primary Key : id\_admin

Foreign Key : id\_user

Tabel 4.15 *Admin*

Nomor	Field	Length	Tipe Data	Keterangan
1	id_admin	11	int	Primary Key
2	id_user	11	int	Foreign Key
3	nama	50	varchar	Nama pengguna

#### 4.2.5.4 Pembayaran

Tabel : Pembayaran

Primary Key : id\_bayar

Foreign Key : id\_keranjang

Tabel 4.16 Pembayaran

Nomor	Field	Length	Tipe Data	Keterangan
1	id_bayar	11	int	Primary Key
2	id_keranjang	11	int	Foreign Key
3	total	11	int	Total pembayaran
4	status	11	varchar	Status pembayaran: P = Pesanan Pending, S = Sedang Diproses, K = Sedang Dikirim, G = Gagal/Expire
5	tanggal_bayar	datetime	datetime	Mencetak waktu tanggal pesanan yang masuk

#### 4.2.6.5 Outfit

Tabel : Outfit

Primary Key : id\_outfit

Foreign Key : id\_kategori

Tabel 4.17 outfit

Nomor	Field	Length	Tipe Data	Keterangan
1	id_outfit	11	int	Primary Key
2	id_kategori	11	int	Foreign Key
3	nama_outfit	50	varchar	Nama outfit
4	harga	11	int	Harga outfit
5	stok	11	int	Stok outfit
6	url_gambar	255	varchar	URL gambar outfit
7	deskripsi_outfit	255	varchar	Deskripsi outfit
8	tanggal_add	datetime	datetime	Kapan data pakaian ditambahkan
9	tanggal_update	datetime	datetime	Kapan data pakaian diperbarui

#### 4.2.6.6 Kategori outfit

Tabel : Kategori Outfit

Primary Key : id\_kategori

Foreign Key : -

Tabel 4.18 Kategori outfit

Nomor	Field	Length	Tipe Data	Keterangan
1	id_kategori	11	int	Primary Key
2	kategori_outfit	50	varchar	Kategori outfit

#### 4.2.6.7 Keranjang

Tabel : Keranjang

Primary Key : id\_keranjang

Foreign Key : id\_user

Tabel 4.19 keranjang

Nomor	Field	Length	Tipe Data	Keterangan
1	id_keranjang	11	int	Primary Key
2	id_user	11	int	Foreign Key

#### 4.2.6.8 Detail keranjang

Tabel : Detail Keranjang

Primary Key : id\_detail\_keranjang

Foreign Key : id\_keranjang, id\_outfit

Tabel 4.20 Detail keranjang

Nomor	Field	Length	Tipe Data	Keterangan
1	id_detail_keranjang	11	int	Primary Key
2	id_keranjang	11	int	Foreign Key
3	id_outfit	11	int	Foreign Key
4	qty	11	int	Jumlah outfit
5	sub_total	11	int	Subtotal harga outfit

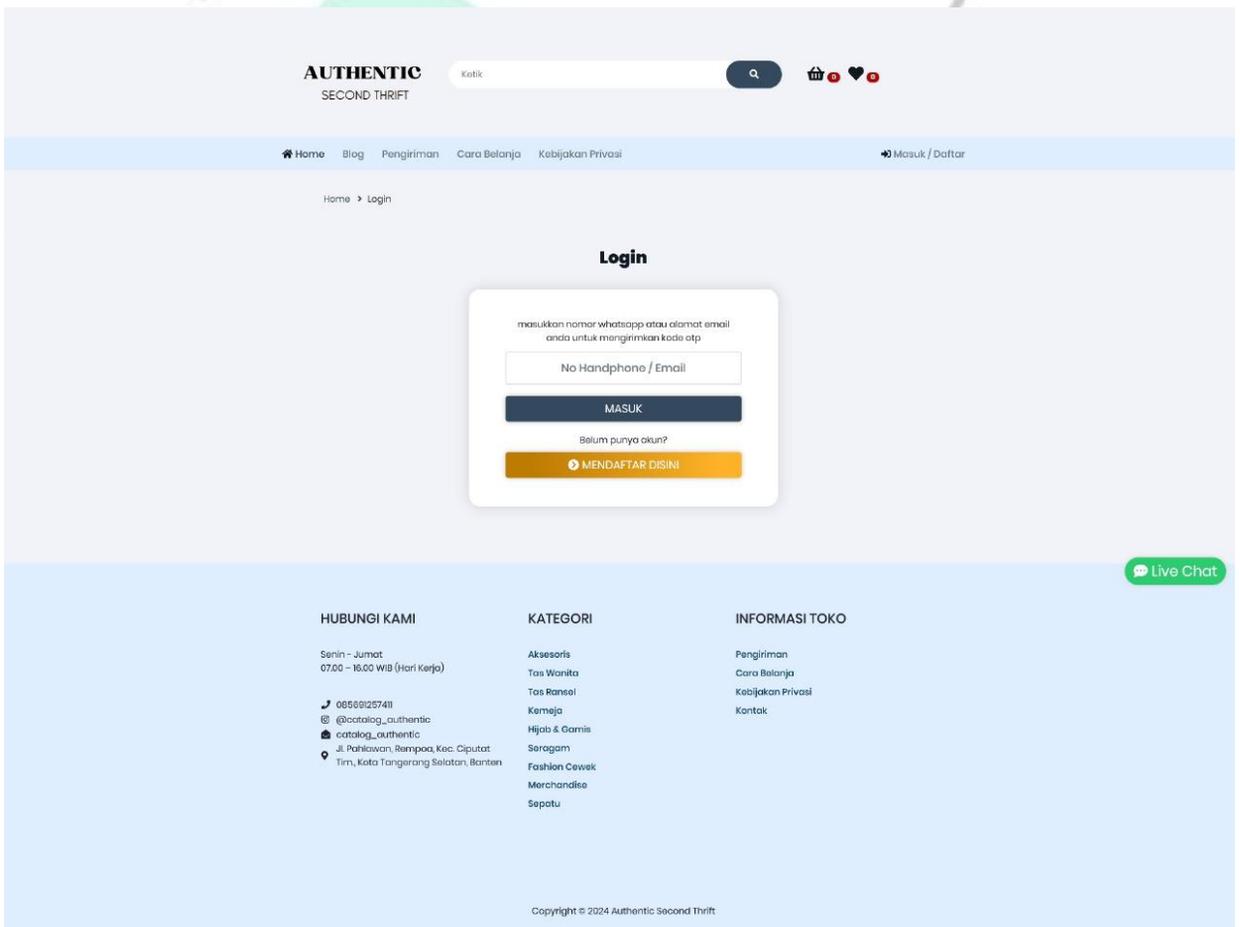
Keterangan

- I. **Primary Key** adalah kolom yang unik dan tidak boleh ada nilai duplikat
- II. **Foreign Key** adalah kolom yang mengacu ke nilai kolom lain di tabel lain

## 4.3. Perancangan Antar Muka Pengguna

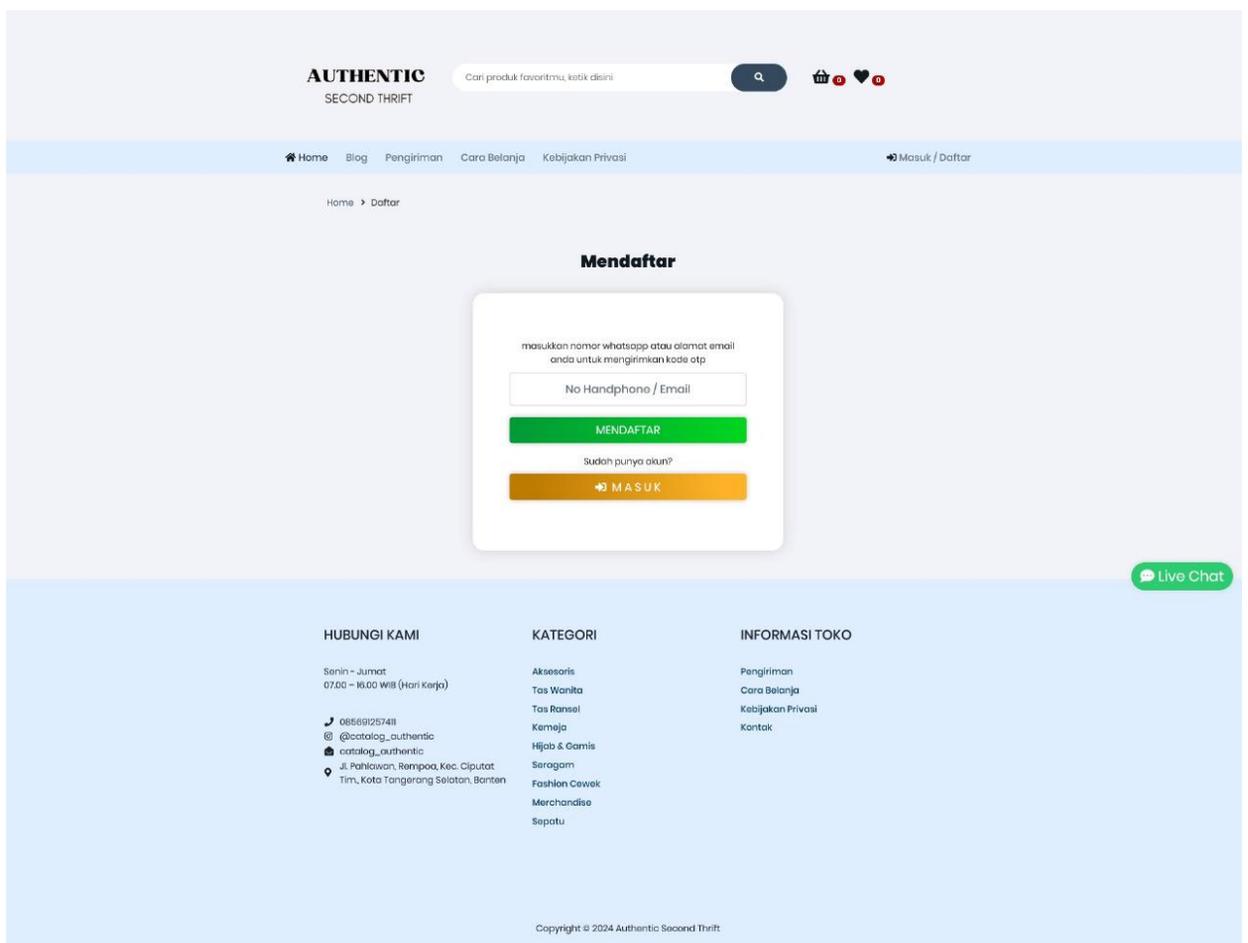
### 4.3.1 Rancangan Layout Layar Atau Tampilan

Pembuatan rancangan layout atau tampilan berfungsi sebagai penggambaran dari tampilan sistem yang dirancang. Rancangan layout layar atau tampilan merujuk pada perencanaan dan penataan elemen-elemen visual di dalam antarmuka pengguna suatu aplikasi atau sistem. Rancangan pembuatan layout ini penting karena untuk perangkat lunak yang di rancang. Pada sistem ini yang saya buat yaitu pembelian barang *second branded* terdapat beberapa layout yaitu *login*, *regist*, cetak pembayaran, *Checkout* pesanan, *delete outfit*, detail *outfit*, dan lain lain. Berikut beberapa tampilan yang dapat dilihat mulai dari halaman register sampai ke halaman checkout. Berikut **Gambar 4.23**



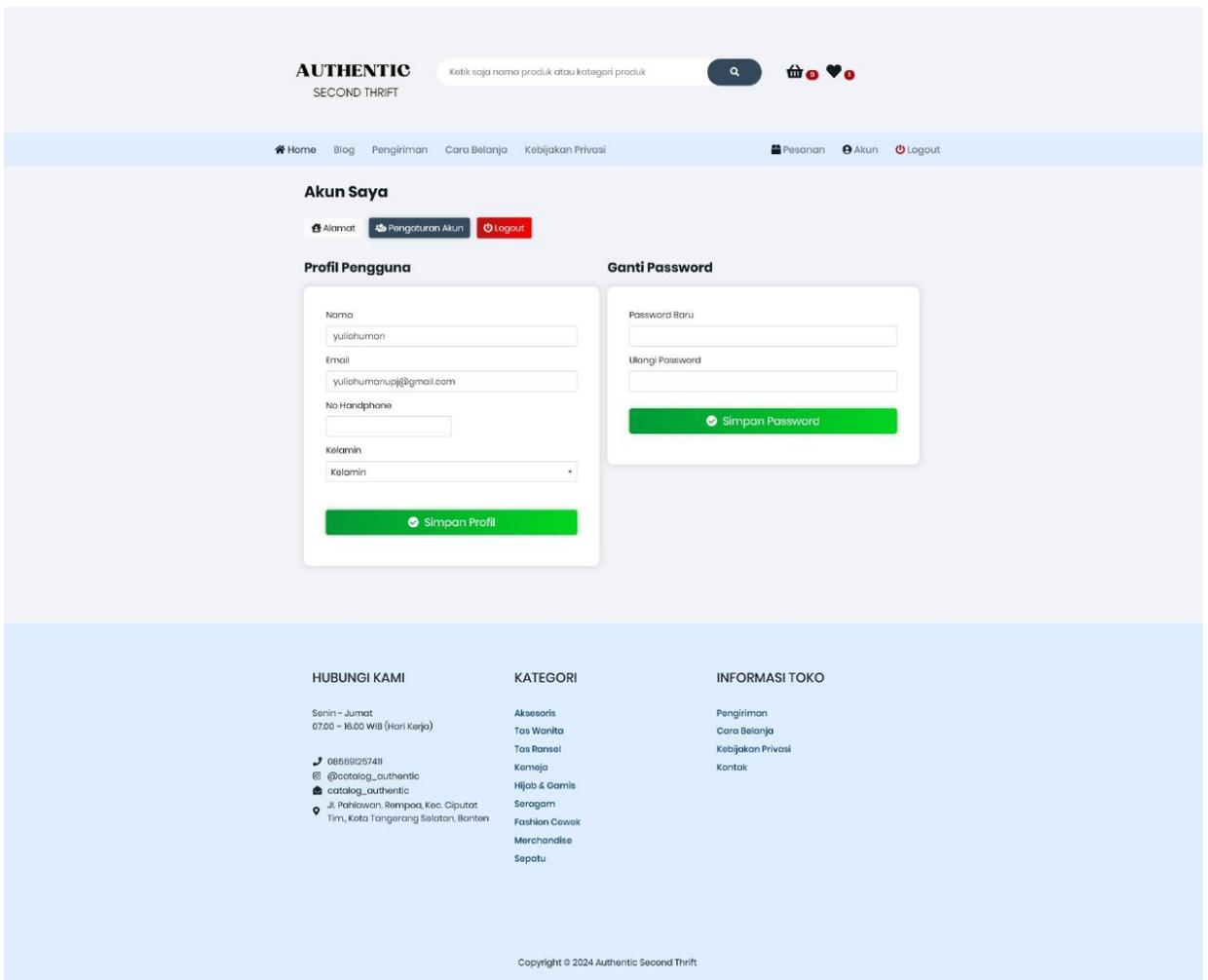
**Gambar 4.26** Halaman Login

Pada **Gambar 4.23** di atas adalah gambar halaman login yaitu masih berbentuk seperti wireframe yang dimana halaman login ini gunanya untuk memvalidasi identitas pengguna dan memverifikasi identitas pengguna. Hanya diminta untuk memasukkan kredensial seperti nama pengguna dan kata sandi. Dan Halaman login ini juga untuk membantu dalam pelacakan aktivitas pengguna dengan mengetahui siapa yang login dan kapan mereka melakukannya.



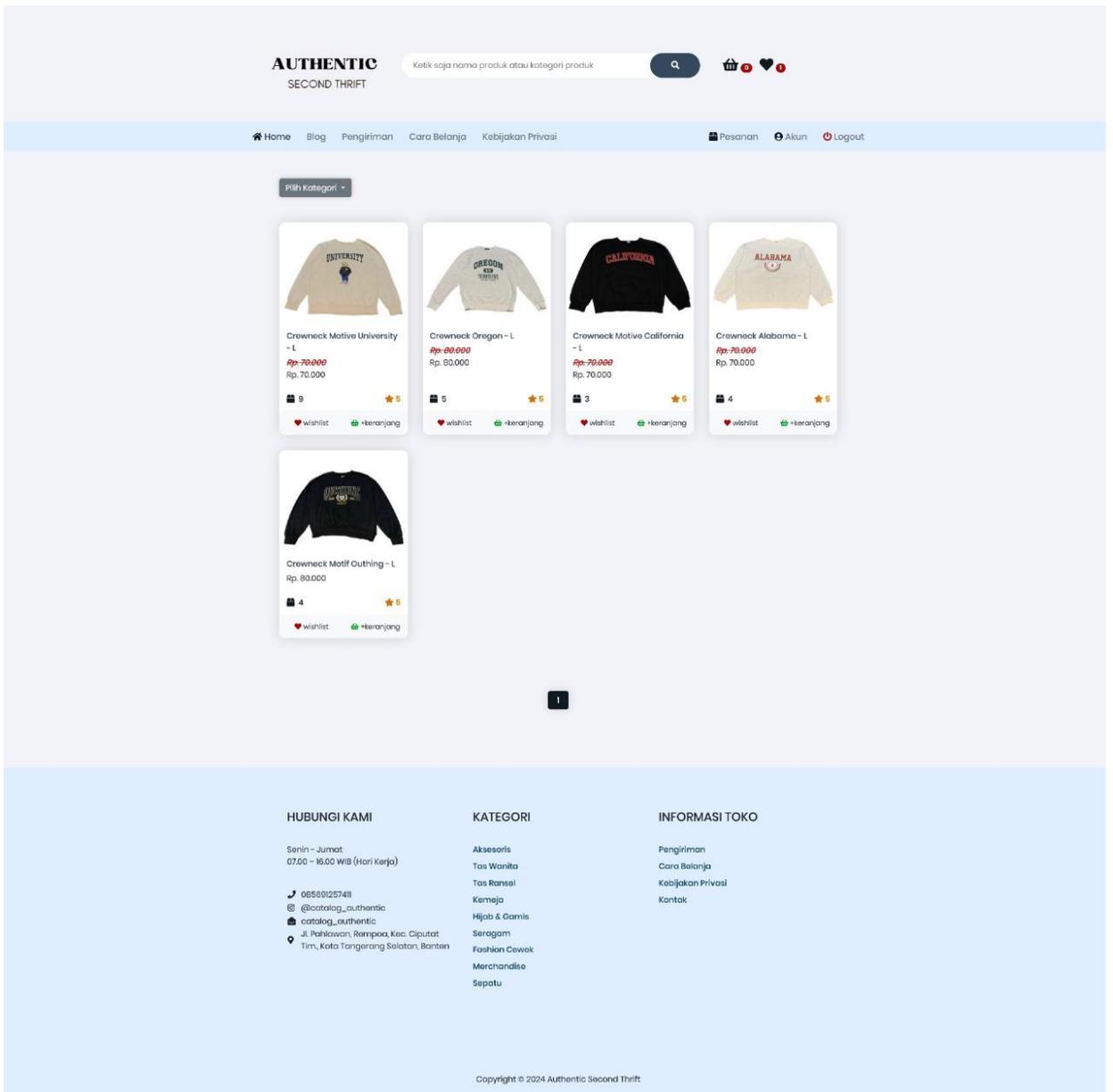
**Gambar 4.27** halaman registrasi

Pada **Gambar 4.27** adalah gambar registrasi yaitu halaman sebelum login, halaman registrasi adalah untuk pengguna membuat akun baru dan menyediakan informasi yang diperlukan.



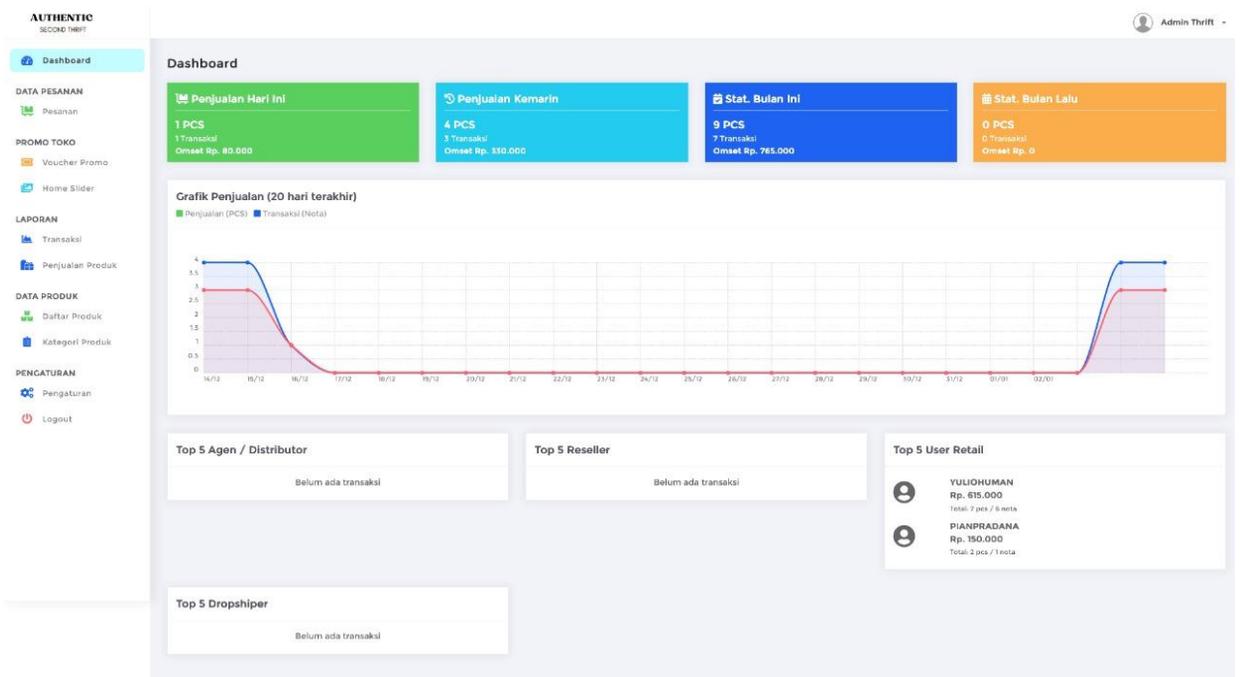
*Gambar 4.28 Profile pelanggan*

Pada **Gambar 4.28** di atas adalah gambar profile pelanggan yang dimana fungsinya untuk memberikan gambaran umum tentang informasi pribadi pelanggan seperti nama, alamat, nomor telepon dan informasi kontak lainnya. Tujuan dibuatnya halaman profile adalah untuk memberikan pengalaman yang personal dan disesuaikan. Serta memungkinkan pelanggan untuk mengelola akun mereka dengan mudah. Dan berguna untuk mengelola pengaturan akun, termasuk pengaturan keamanan seperti perubahan kata sandi dan preferensi akun dan berfungsi sebagai titik sentral bagi pelanggan untuk berinteraksi dengan layanan atau produk yang digunakan.



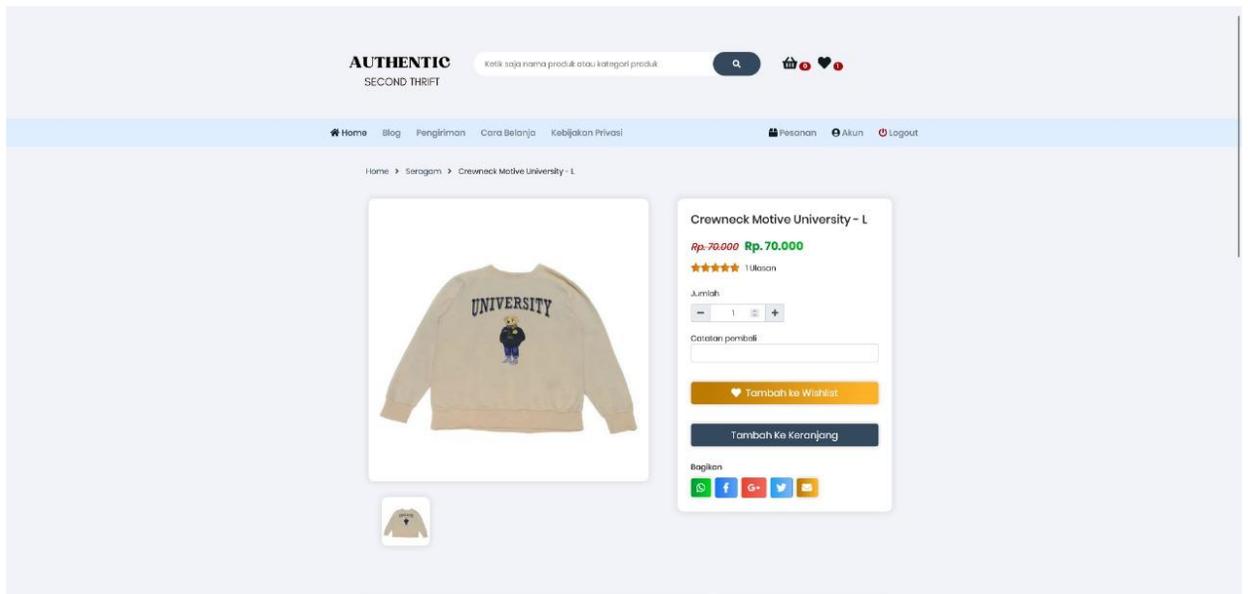
Gambar 4.29 Menu outfit

Pada **Gambar 4.29** adalah gambar dari halaman menu *outfit* tampilan *user interface* di atas adalah pemilihan opsi dari kaos yang mempunyai *brand* dan di rancang untuk meningkatkan pengalaman pengguna dan memfasilitasi proses penjelahan serta pembelian, memudahkan juga untuk pelanggan menjelajahi berbagai kategori dan jenis pakaian, setiap *item* yang di tampilkan dilengkapi oleh harga dan nama barangnya. Tampilan gambar di atas menyajikan produk pakaian terbaru dan melakukan pembaruan koleksi melalui menu membantu menjaga situs web agar tetap relevan dan menarik bagi pengguna.



**Gambar 4.30** Admin dashboard

Pada **Gambar 4.30** *admin dashboard* dalam pembelian barang *second* memiliki berbagai kegunaan yang penting untuk membantu *owner* atau pemilik toko dalam mengelola, memantau dan mengoptimalkan operasional toko *authentic.secondthrift*. Kegunaan *admin dashboard* memungkinkan pemantauan *real-time* terhadap tingkat persediaan barang, membantu *owner* atau pemilik toko untuk mengidentifikasi kapan stok sedang menipis atau ketika perlu dilakukan pemesanan ulang. Dan *Admin dashboard* juga dapat menyajikan analisis penjualan harian, mingguan atau bulanan.

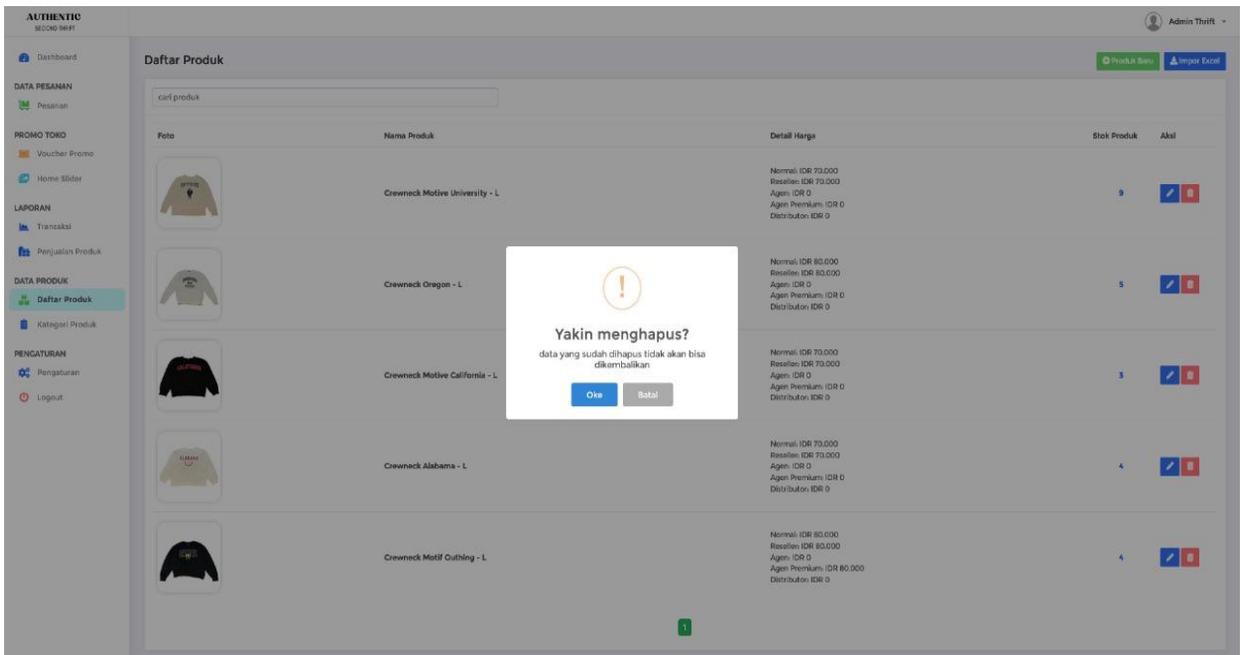


Gambar 4.31 detail outfit

Pada **Gambar 4.31** di atas adalah gambar dari detail outfit yaitu kegunaan dari *detail outfit* yaitu memiliki peran yang sangat penting dalam memberikan informasi lengkap kepada pelanggan dan mendukung proses pembelian. *Detail outfit* menyajikan id pakaian, nama pakaian, stok pakaian dan deskripsi tentang pakaian tersebut. Dengan adanya *detail outfit* atau detail pakaian itu untuk memudahkan pelanggan melihat gambar dari pakaian tersebut untuk mendapatkan gambaran yang jelas tentang produk sebelum membeli. *Detail outfit* atau detail pakaian juga untuk memberikan pandangan yang lebih baik tentang pengalaman orang lain dengan produk tersebut dan membantu pelanggan membuat keputusan yang lebih informasional.

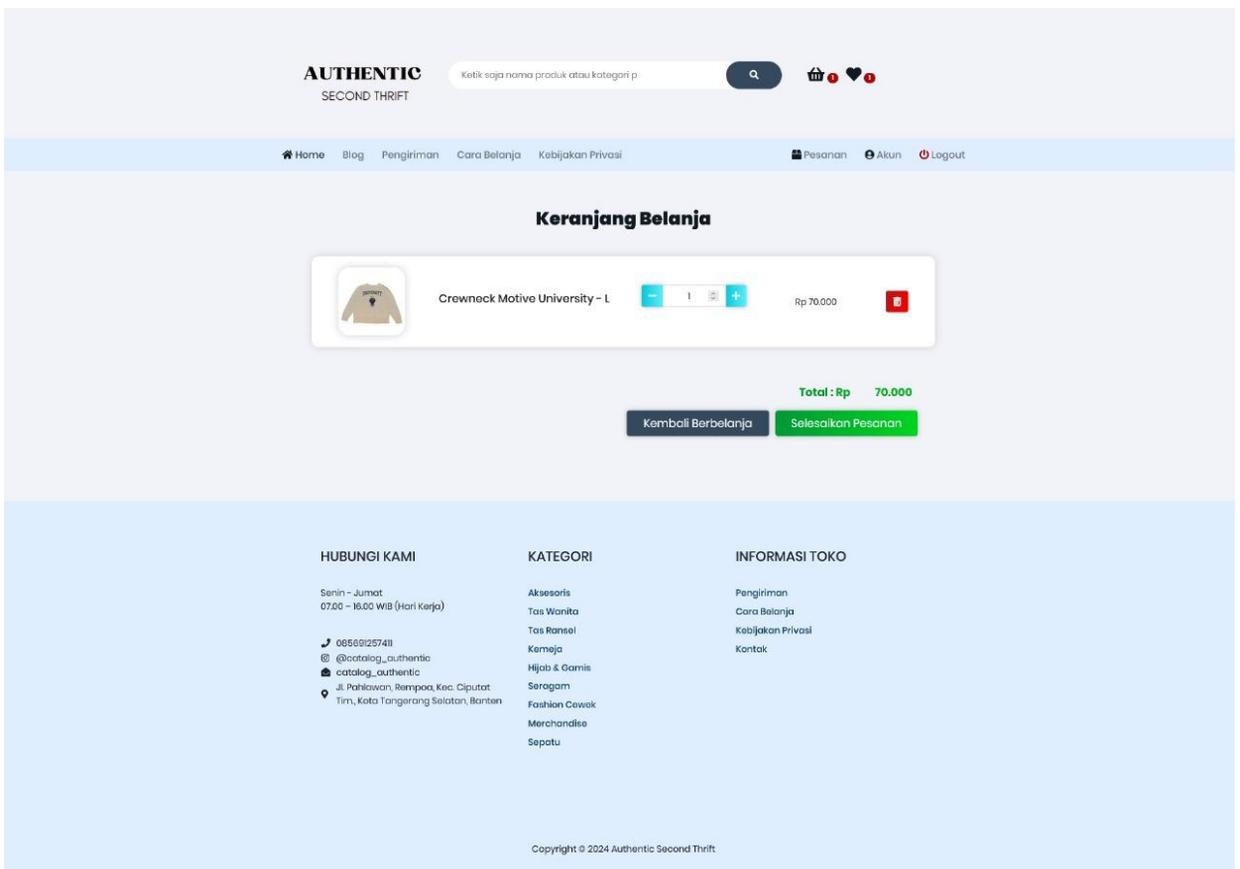
**Gambar 4.32** add outfit

Pada **Gambar 4.32** di atas adalah *add outfit* atau tambah pakaian yaitu adalah untuk memudahkan pelanggan pada saat ingin menambah barang sebagai sarana untuk memberikan kemudahan bagi pengguna untuk memasukkan produk ke dalam suatu keranjang. Menu tambah outfit memungkinkan pengguna untuk dengan cepat menambahkan produk untuk mempercepat proses pembelian. Pengguna dengan sangat mudah mengklik tambah pakaian hal ini meningkatkan kenyamanan dan efisiensi berbelanja online. Setelah menambah produk menu ini biasanya pelanggan atau pembeli menindaklanjuti pembelian barangnya.



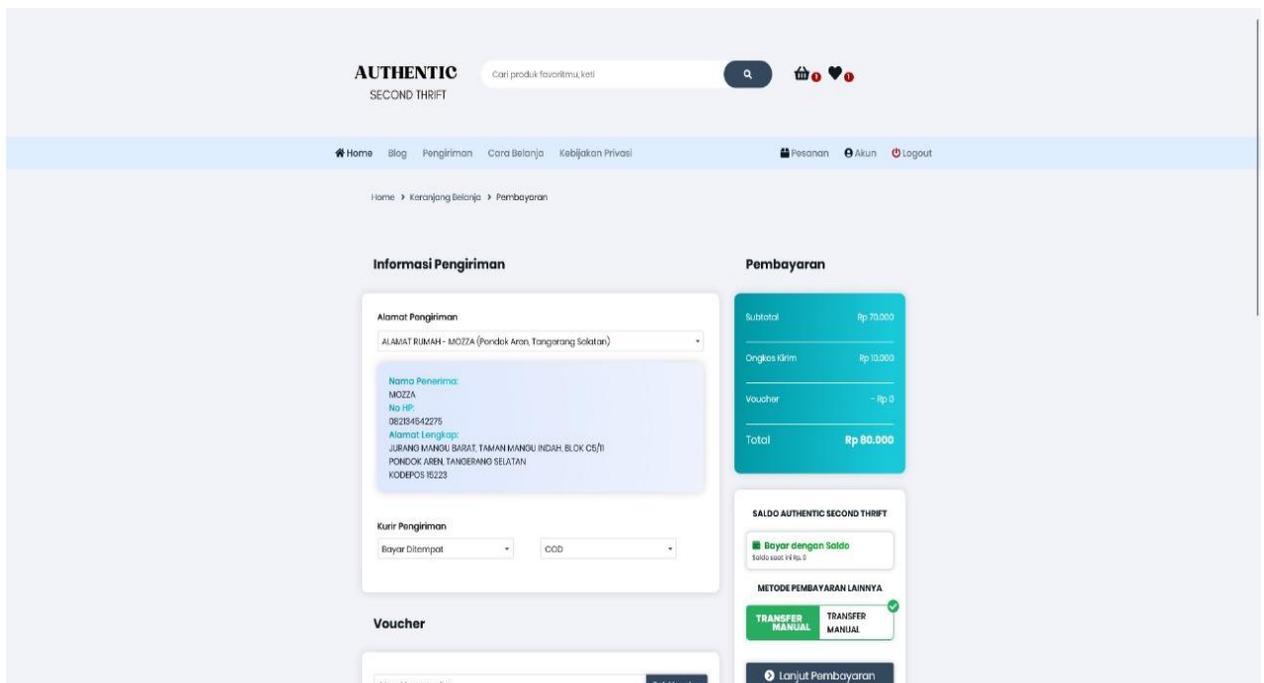
*Gambar 4.33 delete outfit*

Pada **Gambar 4.33** adalah gambar dari *delete outfit* atau hapus pakaian, gunanya ada hapus pakaian adalah memungkinkan pengguna menghapus produk tertentu dari keranjang belanja mereka. Ini memberikan fleksibilitas kepada pelanggan untuk memilih dan mengelola item yang ingin mereka beli dan juga bila mana pelanggan secara tidak sengaja menambahkan produk yang salah atau dengan jumlah yang tidak sesuai menu hapus pakaian memungkinkan mereka untuk melakukan koreksi dengan mudah tanpa harus keluar dari keranjang belanja. Dan pelanggan dapat menggunakan menu hapus pakaian untuk menghapus produk tertentu dan kemudian menelusuri katalog produk untuk menemukan atau memilih produk yang lebih sesuai dengan keinginan atau kebutuhan pelanggan.



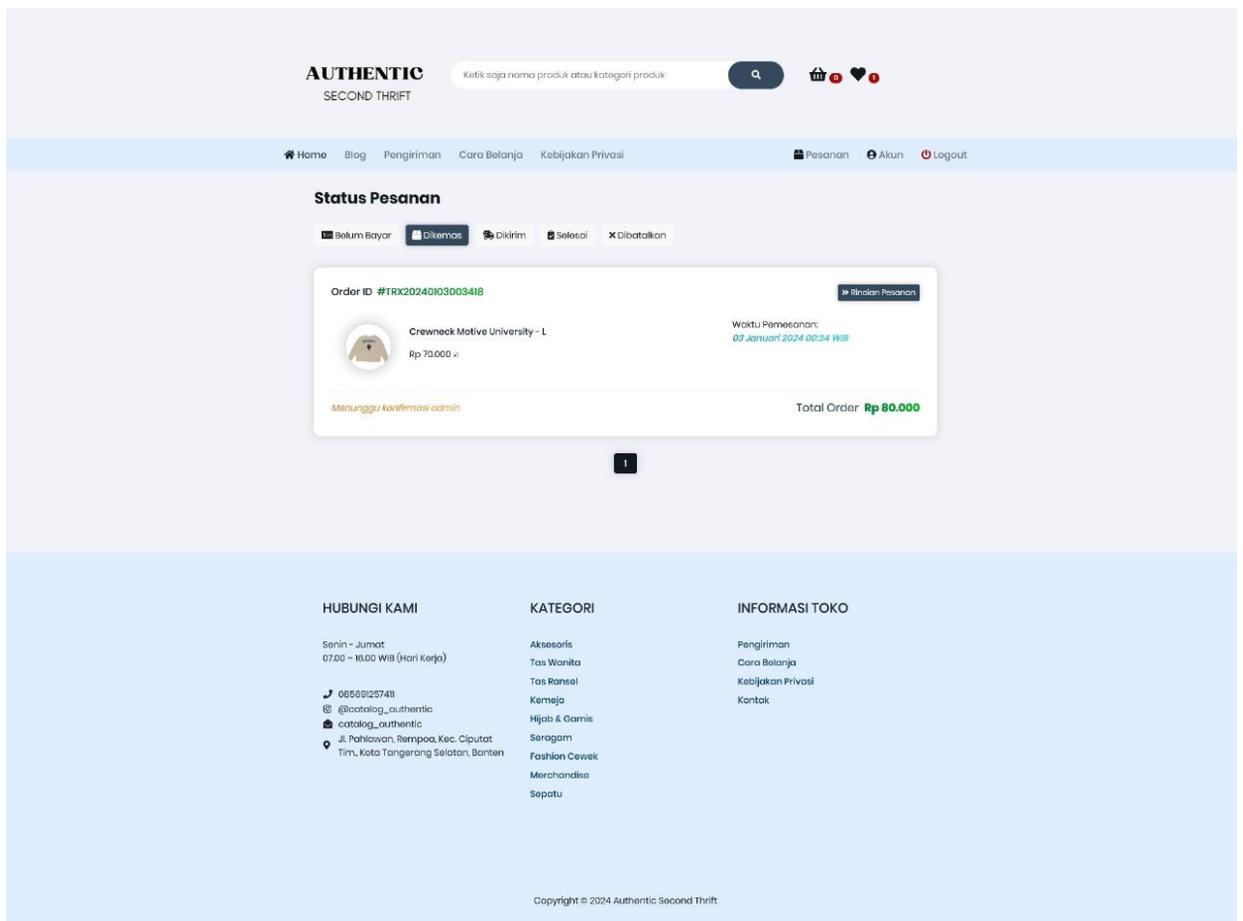
*Gambar 4.34 keranjang belanja*

Pada **Gambar 4.34** yaitu adalah gambar halaman keranjang belanja gunanya keranjang belanja memungkinkan pelanggan untuk melihat daftar produk yang telah mereka pilih untuk dibeli, jadinya untuk memberikan gambaran keseluruhan tentang produk yang akan dibeli sebelum melanjutkan ke pembayaran. Pelanggan bisa meninjau kembali produk yang telah dipilih dan melakukan koreksi jika ada suatu kesalahan atau perubahan yang diperlukan seperti mengubah jumlah produk atau menghapus produk tertentu. Kesimpulannya adalah keranja belanja membantu dan menyusun pesanan pembelian, memberikan gambaran yang jelas tentang biaya dan memberikan fleksibilitas kepada pelanggan untuk mengaturnya.



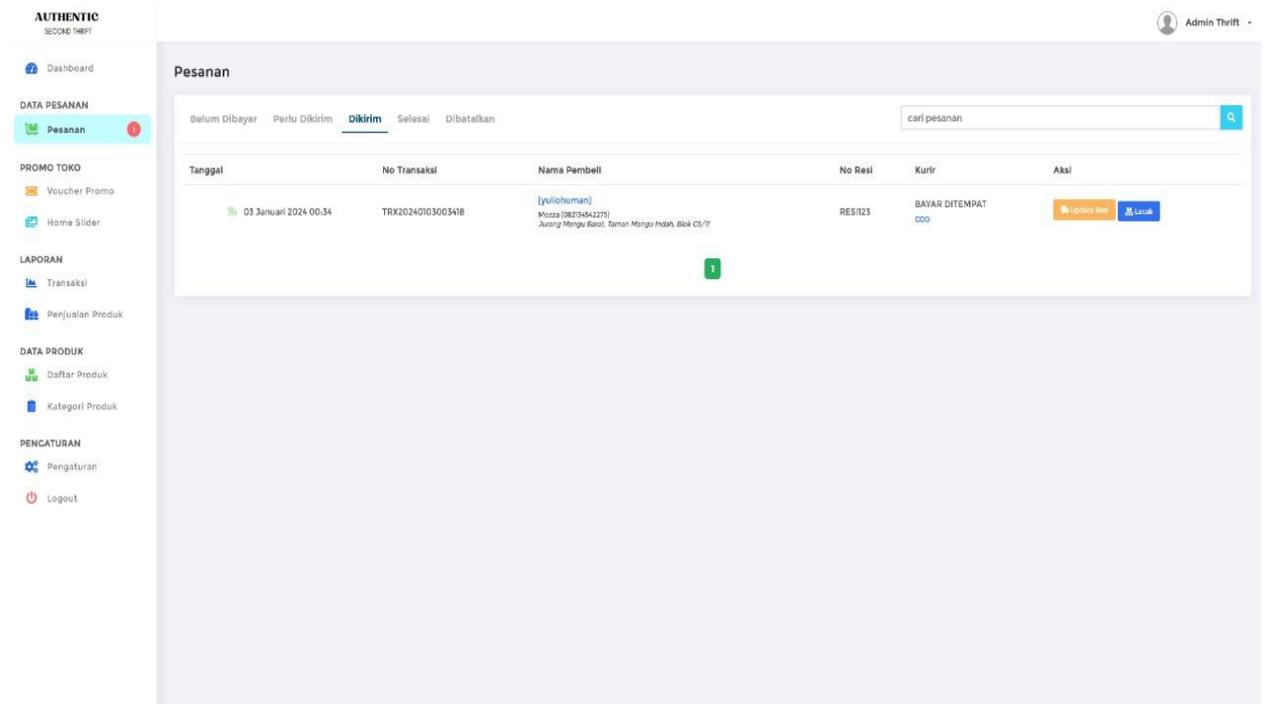
**Gambar 4.35** checkout pesanan

Pada **Gambar 4.35** adalah gambar dari halaman *checkout* pesanan gunanya untuk proses pembelian barang tahap akhir dari proses pembelian di mana pelanggan menyelesaikan transaksi mereka dengan membayar dan mengonfirmasi pesanan. *Checkout* memungkinkan pembeli atau pelanggan untuk bisa memasukkan informasi pembayaran seperti rincian informasi pembayaran lainnya agar transaksi dapat diproses. Pelanggan dapat mengonfirmasi atau memasukkan alamat pengiriman tempat barang yang akan dikirim. Kesimpulan adanya *checkout* pembelian adalah untuk menyelesaikan pembelian mereka dengan lancar.



**Gambar 4.36** cetak pembayaran

Pada **Gambar 4.36** adalah halaman cetak pembayaran berfungsi untuk sebagai bukti transaksi bagi pelanggan dengan menyimpan atau mencetak konfirmasi pembayaran, pelanggan memiliki rekaman berupa fisik yang dapat mereka gunakan untuk referensi. Cetak pembayaran dapat membantu dalam proses rekonsiliasi keuangan pemilik toko dapat menggunakan konfirmasi pembayaran sebagai referensi saat menyusun laporan keuangan atau menyelesaikan pencocokan transaksi. Dan cetak pembayaran dapat berfungsi sebagai pengingat rincian transaksi termasuk barang atau layanan yang dibeli, jumlah yang dibayarkan dan informasi lainnya.



*Gambar 4.37 kelola pesanan*

Pada **Gambar 4.37** diatas adalah gambar dari halaman kelola pesanan, kelola pesanan memungkinkan *owner*/pemilik toko untuk melacak status pesanan mereka mulai dari proses pembayaran hingga pengiriman dan memberikan informasi *valid* tentang di mana pesanan tersebut berada dalam rantai pasokan dan untuk memungkinkan pelanggan untuk mengonfirmasi bahwa pembayaran telah diterima dan diverifikasi. Memberikan akses ke informasi rinci tentang pesanan seperti item yang dibeli, jumlah dan harga.

**AUTHENTIC**  
SECOND HPPF

Admin Thrift

Dashboard

**DATA PESANAN**

Pesanan

PROMO TOKO

Voucher Promo

Home Slider

LAPORAN

Transaksi

Penjualan Produk

DATA PRODUK

Daftar Produk

Kategori Produk

PENGATURAN

Pengaturan

Logout

**Pesanan**

Belum Dibayar Perlu Dikirim Dikirim **Selesai** Dibatalkan

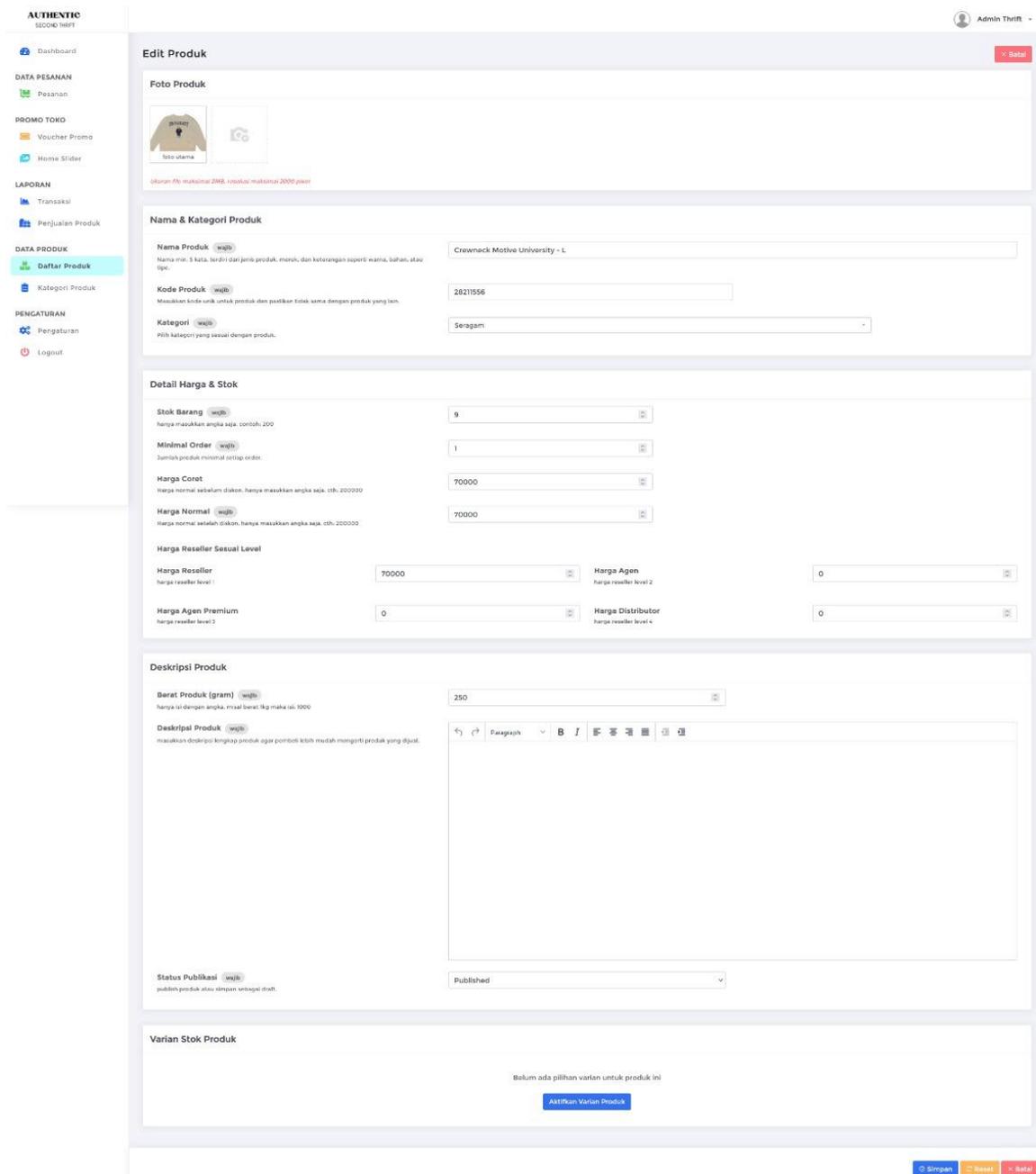
cari pesanan

Tanggal	No Transaksi	Nama Pembeli	Total	Total Ongkir	Kurir	Aksi
03 Januari 2024 00:34	TRX20240103003418	[yuliohuman] Moza (082134542275) Jurang Mangrove Barat, Taman Mangrove Indah, Blok C5/7i	80.000	10.000	BAYAR DITEMPAT COD	<a href="#">Detail</a> <a href="#">Lacak</a>
02 Januari 2024 23:41	TRX20240102234156 Bayar Ditempat (COD)	[yuliohuman] Moza (082134542275) Jurang Mangrove Barat, Taman Mangrove Indah, Blok C5/7i	90.000	10.000	BAYAR DITEMPAT COD	<a href="#">Detail</a> <a href="#">Lacak</a>
02 Januari 2024 00:02	TRX20240102000235	[planpradana] Allan Rifqi Perdana (082134542275) Jl. Cendana No.5	150.000	10.000	BAYAR DITEMPAT COD	<a href="#">Detail</a> <a href="#">Lacak</a>
01 Januari 2024 23:56	TRX20240101235613	[yuliohuman] Moza (082134542275) Jurang Mangrove Barat, Taman Mangrove Indah, Blok C5/7i	90.000	10.000	BAYAR DITEMPAT COD	<a href="#">Detail</a> <a href="#">Lacak</a>
01 Januari 2024 23:46	TRX20240101234658 Bayar Ditempat (COD)	[yuliohuman] Moza (082134542275) Jurang Mangrove Barat, Taman Mangrove Indah, Blok C5/7i	90.000	10.000	BAYAR DITEMPAT COD	<a href="#">Detail</a> <a href="#">Lacak</a>
01 Januari 2024 22:26	TRX20240101222617 Bayar Ditempat (COD)	[yuliohuman] Moza (082134542275) Jurang Mangrove Barat, Taman Mangrove Indah, Blok C5/7i	180.000	10.000	BAYAR DITEMPAT COD	<a href="#">Detail</a> <a href="#">Lacak</a>

1

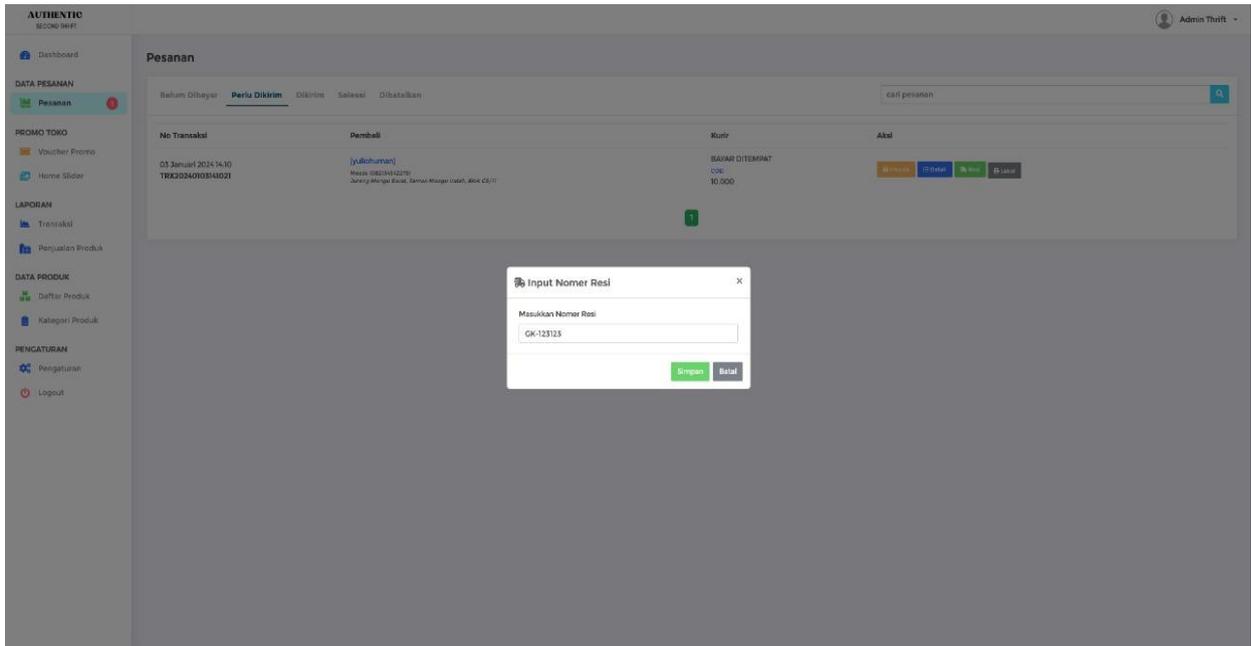
Gambar 4.38 kelola outfit

Pada Gambar 4.38 di atas adalah gambar dari halaman kelola outfit yang berfungsi sebagai fitur yang memungkinkan owner/pemilik toko untuk membuat, menyimpan, dan mengelola kombinasi pakaian dan memungkinkan pemilik toko dengan cepat memilih pakaian yang sudah dipilih sebelumnya dan untuk menghemat waktu dalam mempersiapkan penampilannya sehari-hari. Dan untuk memungkinkan pemilik toko merencanakan pembelian pakaian baru dengan lebih baik berdasarkan kebutuhan mereka untuk melengkapi outfit yang di buat oleh pemilik toko.



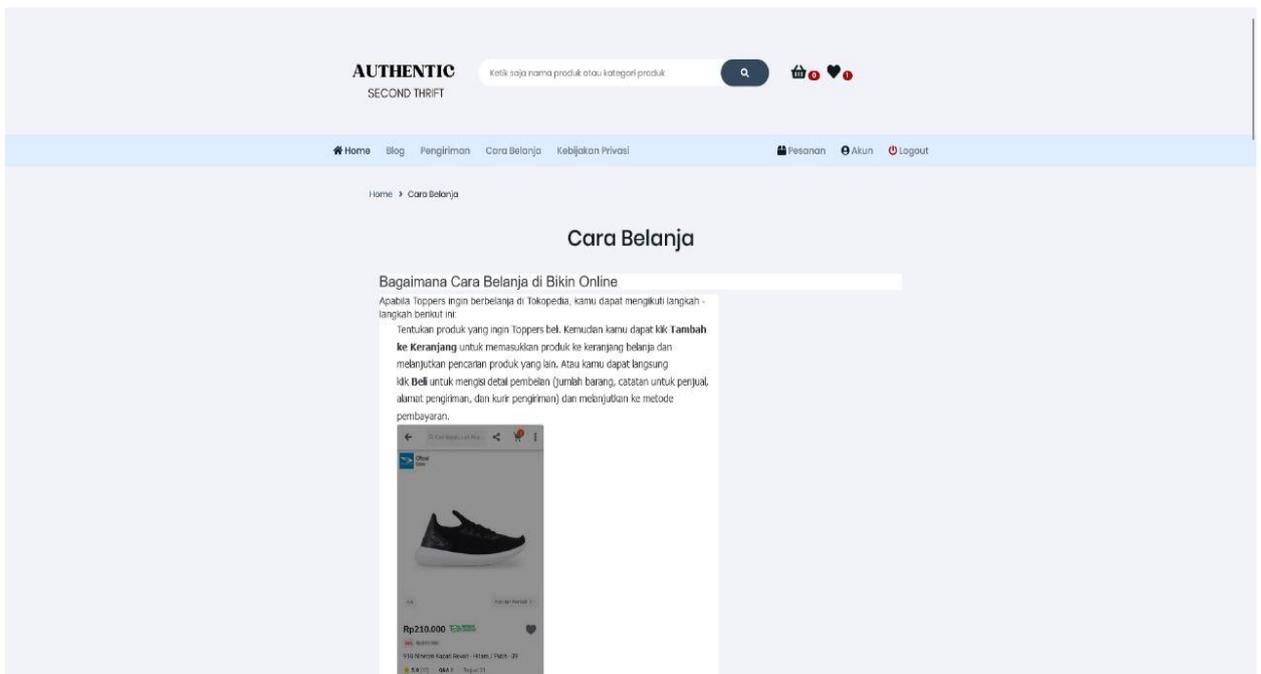
Gambar 4.39 Edit outfit

Pada **Gambar 4.39** adalah gambar dari halaman edit outfit gunanya untuk memberikan pemilik toko berkesempatan untuk menyusun ulang, mengubah atau menyesuaikan kombinasi dari pakaian yang di taro di *authentic.secondthrift*. Memberikan pemilik toko kemampuan untuk edit pakaian dan juga untuk meningkatkan peluang penjualan karena pembeli dapat dengan mudah melihat bagaimana beberapa item pakaian dapat dipadukan untuk menciptakan gaya yang diinginkan. Kesimpulannya edit pakaian penting untuk kebutuhan pelanggan.



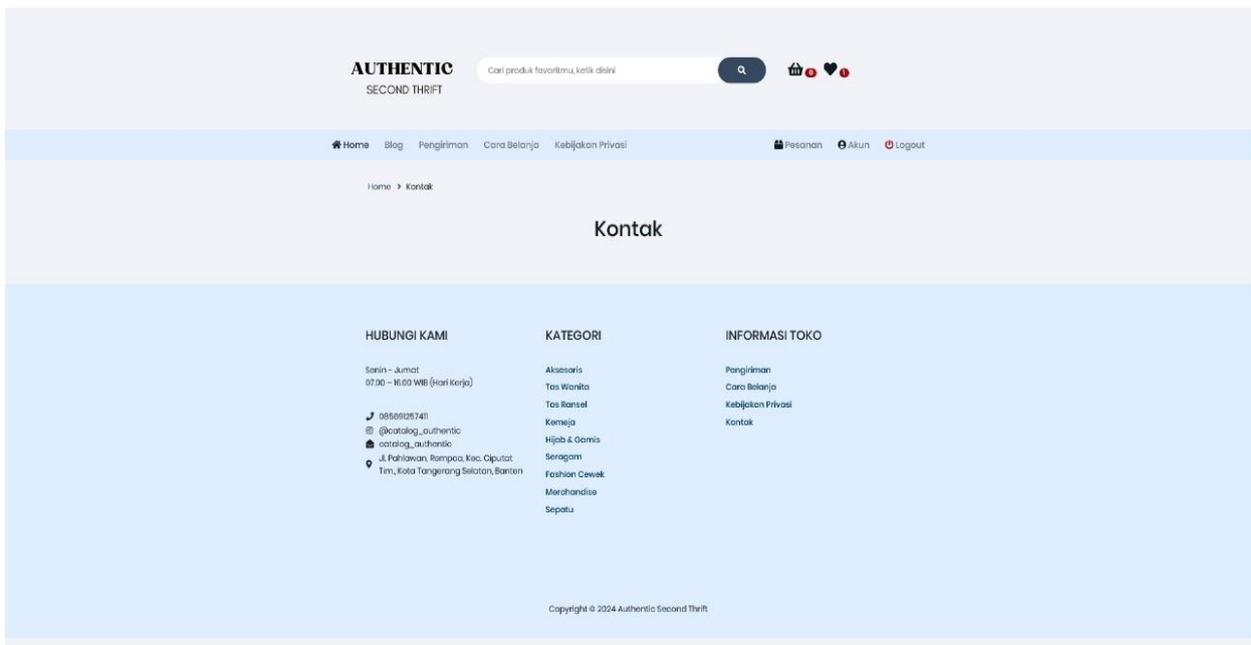
**Gambar 4.40** Isi Resi

Pada **Gambar 4.40** yaitu adalah halaman isi resi yang memungkinkan pelanggan dan pemilik toko untuk bisa melacak status pengiriman pesanan. Pelanggan dapat memantau perjalanan paket secara *real-time* dan pemilik toko bisa dapat memberikan informasi yang lebih akurat mengenai lokasi dan perkiraan waktu kedatangan. Nomor resi juga memberikan tingkat transparansi yang tinggi kepada pelanggan, mereka dapat memeriksa sendiri status pengiriman dan memastikan bahwa pesanan mereka sedang dalam proses.



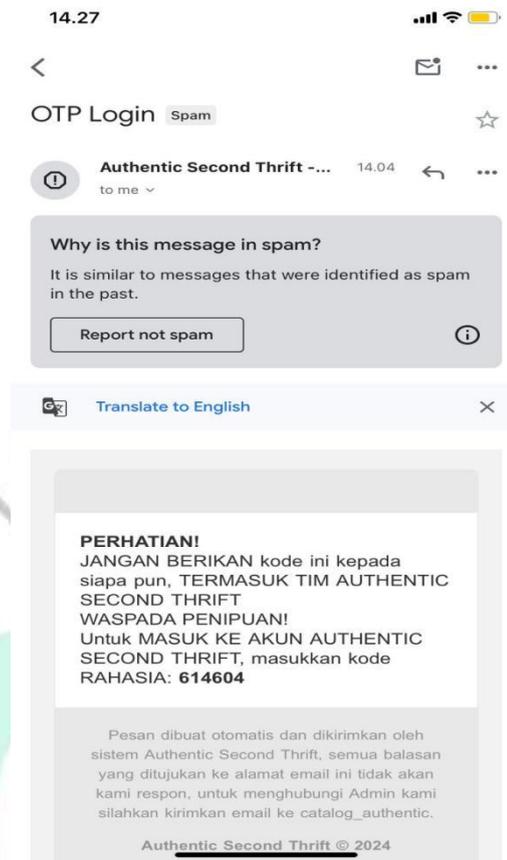
Gambar 4.41 About us

Pada **Gambar 4.41** adalah gambar dari halaman *about us* adalah halaman yang menampilkan suatu identitas untuk toko *online* atau bisnis untuk memperkenalkan diri. Informasi tentang latar belakang dari toko *authentic.secondthrift* untuk mengetahui visi dan misi dan dapat meningkatkan kepercayaan pengguna terhadap aplikasi dan bisnis.



**Gambar 4.42** *contact person*

Pada **Gambar 4.42** adalah halaman *contact person* aplikasi transaksi pembelian berperan penting dalam menyediakan saluran komunikasi yang efektif antara pelanggan dan pemilik toko. *Contact person* memungkinkan pelanggan untuk menghubungi toko dengan pertanyaan, keluhan atau permintaan pendukung. Hal ini dapat memberikan pelanggan rasa kenyamanan karena mereka tahu bahwa ada saluran komunikasi langsung yang dapat digunakan.



*Gambar 4.43 email aktivasi*

Pada **Gambar 4.43** adalah gambar email aktivasi yaitu digunakan untuk menverifikasi identitas pengguna yang baru mendaftar di aplikasi pembelian untuk membantu mencegah pendaftaran palsu atau penggunaan data palsu. Proses aktivasi email membantu meningkatkan keamanan aplikasi dengan memastikan bahwa hanya pengguna yang memiliki akses ke alamat email yang terdaftar yang dapat mengonfirmasi dan mengaktifkan akun.

#### 4.4. Perancangan Implementasi

Perancangan Implementasi membantu dalam pembuatan aplikasi untuk mendefinisikan tujuan yang ingin di capai. Melalui perancangan implementasi dapat disusun rencana yang sangat terstruktur dan terukur seperti mencakup sumber daya, jadwal waktu, dan pengelolaan risiko. Untuk perencanaan yang baik dapat membantu menghindari masalah dan menjamin kelancaran proses implementasi. Dan dalam pembuatan aplikasi dilakukan tahap pengujian untuk memastikan fitur-fitur yang berada didalam sistem telah berjalan dengan baik dan sesuai dengan apa yang di rancang dan bisa menemukan *bug* atau *error*.

*Tabel 4.21 Hasil Tahapan pengujian*

NO	Test Name	Test Step	Result
1.	Register	1.Mengakses halaman registrasi 2.Mengisi email & password 3.Klik buton registrasi	Success
2.	Aktivasi email	1.Mengakses email 2.Klik bagian spam aktivasi email 3.Mengecek aktivasi email	Success
3.	Melihat daftar outfit	1.Menampilkan halaman daftar outfit 2.Menekan button daftar outfit	Success
4.	Melihat wishlist	1.Menampilkan halaman wishlist 2.Menekan button melihat wishlist	Success
5.	Tambah keranjang	1.Melihat salah satu produk 2.Klik button keranjang 3.Menambahkan salah satu produk	Success
6.	Kurangi keranjang	1.Menampilkan halaman kurangi keranjang 2.Klik button kurangi keranjang	Success
7.	Checkout	1.Mengakses checkout pembelian 2.Memilih kurir 3.Bayar ke kode rekening 4.Cek riwayat pesanan	Success
8.	Melihat riwayat transaksi	1.Klik button riwayat transaksi 2.Mengakses data riwayat transaksi 3.Melihat riwayat transaksi	Success

9.	Kelola outfit	1.Mengakses halaman dashboard penjualan 2.Klik button <i>add,edit</i> dan <i>delete</i> 3.Klik data tersimpan	Success
10.	Kelola biodata toko	1.Mengakses halaman kelola biodata toko 2.Klik button kelola biodata toko 3.Melakukan pemantauan kelola biodata toko	Success
11.	Kelola pesanan pelanggan	1.Mengakses halaman kelola pelanggan 2.Klik konfirmasi pesanan 3.Mengisi no resi dari kurir	Success
12.	List dashboard penjualan	1.Mengakses halaman dashboard penjualan 2.Menampilkan halaman list dashboard penjualan	Success

