

DPPL-02

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Klip

untuk:

Lukman Nurdiansyah

Dipersiapkan oleh:

Kelompok 6

MUHAMMAD FACHRI HAIKAL - 1301202398

RIZKY FERDIAN PRASETYO - 1301204229

BAGUS PANULUH - 1301204312


M. NAUFAL RIFQI RAMDHANI - 1301201572

MUHAMMAD RAFI ANDEO PRAJA - 1301200278

Program Studi S1 Informatika

Fakultas Informatika, Universitas Telkom

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

	Prodi S1- Informatika Universitas Telkom	Nomor Dokumen		Halaman
		<i>DPPL-02 <01:6></i>		67
		Revisi	3	<i>Tgl: 15 Januari 2023</i>

Daftar Perubahan

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

Daftar Perubahan	2
Daftar Halaman Perubahan	3
Daftar Isi	4
Daftar Gambar	7
1. Pendahuluan	9
1.1. Tujuan Penulisan Dokumen	9
1.2. Lingkup Masalah	9
1.3. Definisi dan Istilah	9
1.4. Referensi	10
2. Perancangan Global	11
2.1. Rencana Lingkungan Implementasi	11
2.2. Deskripsi Arsitektur Perangkat Lunak	12
3. Perancangan Rinci	13
3.1. Realisasi Use Case	13
3.1.1. Use Case #1 Registrasi	14
3.1.1.1. Use Case Scenario #1 Registrasi	14
3.1.1.2. UI Design dan Deskripsi Objek UI #1 Registrasi	15
3.1.1.3. Identifikasi Object Baru dan Tipe Kelas #1 Registrasi	16
3.1.1.4. Sequence Diagram #1 Registrasi	17
3.1.1.5. Class Diagram #1 Registrasi	17
3.1.2. Use Case #2 Login	17
3.1.2.1. Use Case Scenario #2 Login	17
3.1.2.2. UI Design dan Deskripsi Objek UI #2 Login	18
3.1.2.3. Identifikasi Object Baru & Tipe Kelas #2 Login	20
3.1.2.4. Sequence Diagram #2 Login	20
3.1.2.5. Class Diagram #2 Login	21
3.1.3. Use Case #3 Logout	21
3.1.3.1. Use Case Scenario #3 Logout	21
3.1.3.2. UI Design dan Deskripsi Objek UI #3 Logout	22
3.1.3.3. Identifikasi Object Baru & Tipe Kelas #3 Logout	24
3.1.3.4. Sequence Diagram #3 Logout	24
3.1.3.5. Class Diagram #3 Logout	24
3.1.4. Use Case #4 Mengelola Data Produk	25
3.1.4.1. Use Case Scenario #4 Mengelola Data Produk	25
3.1.4.2. UI Design dan Deskripsi Objek UI #4 Mengelola Data Produk	26

3.1.4.3. Identifikasi Object Baru & Tipe Kelas #4 Mengelola Data Produk	28
3.1.4.4. Sequence Diagram #4 Mengelola Data Produk	29
3.1.4.5. Class Diagram #4 Mengelola Data Produk	29
3.1.5. Use Case #5 Mengelola Data Transaksi	30
3.1.5.1. Use Case Scenario #5 Mengelola Data Transaksi	30
3.1.5.2. UI Design dan Deskripsi Objek UI #5 Mengelola Data Transaksi	31
3.1.5.3. Identifikasi Object Baru & Tipe Kelas #5 Mengelola Data Transaksi	33
3.1.5.4. Sequence Diagram #5 Mengelola Data Transaksi	33
3.1.5.5. Class Diagram #5 Mengelola Data Transaksi	34
3.1.6. Use Case #6 Membeli Barang	34
3.1.6.1. Use Case Scenario #6 Membeli Barang	34
3.1.6.2. UI Design dan Deskripsi Objek UI #6 Membeli Barang	35
3.1.6.3. Identifikasi Object Baru & Tipe Kelas #6 Membeli Barang	36
3.1.6.4. Sequence Diagram #6 Membeli Barang	37
3.1.6.5. Class Diagram #6 Membeli Barang	37
3.1.7. Use Case #7 Halaman Utama Penjualan/Pembelian	37
3.1.7.1. Use Case Scenario #7 Halaman Utama Pembelian	37
3.1.7.2. UI Design dan Deskripsi Objek UI #7 Halaman Utama Penjualan/Pembelian	38
3.1.7.3. Identifikasi Object Baru & Tipe Kelas #7 Halaman Utama Penjualan/Pembelian	40
3.1.7.4. Sequence Diagram #7 Halaman Utama Penjualan/Pembelian	40
3.1.7.5. Class Diagram #7 Halaman Utama Penjualan/Pembelian	41
3.1.8. Use Case #8 Melihat Detail Produk	41
3.1.8.1. Use Case Scenario #8 Melihat Detail Produk	41
3.1.8.2. UI Design dan Deskripsi Objek UI #8 Melihat Detail Produk	41
3.1.8.3. Identifikasi Object Baru & Tipe Kelas #8 Melihat Detail Produk Penjualan/Pembelian	43
3.1.8.4. Sequence Diagram #8 Melihat Detail Produk	44
3.1.8.5. Class Diagram #8 Melihat Detail Produk	44
3.1.9. Use Case #9 Mengelola Data User	45
3.1.9.1. Use Case Scenario #9 Mengelola Data User	45
3.1.9.2. UI Design dan Deskripsi Objek UI #9 Mengelola Data User	45
3.1.9.3. Identifikasi Object Baru & Tipe Kelas #9 Mengelola Data User	47
3.1.9.4. Sequence Diagram #9 Mengelola Data User	48
3.1.9.5. Class Diagram #9 Mengelola Data User	48
3.1.10. Use Case #10 Add to Cart	48
3.1.10.1. Use Case Scenario #10 Add to Cart	48
3.1.10.2. UI Design dan Deskripsi Objek UI #10 Add to Cart	49
3.1.10.3. Identifikasi Object Baru & Tipe Kelas #10 Add to Cart	51

3.1.10.4. Sequence Diagram #10 Add to Cart	52
3.1.11.5. Class Diagram #10 Add to Cart	52
3.1.12. Use Case #11 Melihat History	52
3.1.12.1. Use Case Scenario #11 Melihat History	52
3.1.12.2. UI Design dan Deskripsi Objek UI #11 Melihat History	53
3.1.12.3. Identifikasi Object Baru & Tipe Kelas #11 Melihat History	55
3.1.12.4. Sequence Diagram #11 Melihat History	55
3.1.12.5. Class Diagram #11 Melihat History	56
3.2. Diagram Kelas Keseluruhan	57
3.3. Perancangan Data / Basis Data	60
3.4. Perancangan Algoritma dan/atau Query	60
4. Matriks Keruntutan (Requirement Traceability Matrix)	67

Daftar Gambar

Gambar 1	Arsitektur Perangkat Lunak	12
Gambar 2	Use Case Diagram	13
Gambar 3	Page Registrasi	15
Gambar 4	Sequence Diagram Registrasi	17
Gambar 5	Class Diagram Registrasi	17
Gambar 6	Page Login	18
Gambar 7	Page Forgot Password	19
Gambar 8	Sequence Diagram Login	20
Gambar 9	Class Diagram Registrasi	21
Gambar 10	Page Dashboard User	22
Gambar 11	Page Dashboard Admin	22
Gambar 12	Class Diagram Logout	24
Gambar 13	Class Diagram Logout	24
Gambar 14	Page Product	26
Gambar 15	Page Create Product	26
Gambar 16	Page Edit Product	27
Gambar 17	Sequence Diagram Mengelola Data Produk	29
Gambar 18	Class Diagram Mengelola Data Produk	29
Gambar 19	Page Transaction	31
Gambar 20	Page Edit Transaction	31
Gambar 21	Page Show Transaction	32
Gambar 22	Sequence Diagram Mengelola Data Transaksi	33
Gambar 23	Class Diagram Mengelola Data Transaksi	34
Gambar 24	Page Transaction	35
Gambar 25	Sequence Diagram Membeli Barang	37
Gambar 26	Class Diagram Membeli Barang	37

Gambar 27 Page Home	38
Gambar 28 Page Home	39
Gambar 29 Sequence Diagram Halaman Utama Penjualan/Pembelian	40
Gambar 30 Class Diagram Halaman Utama Penjualan/Pembelian	41
Gambar 31 Page Detail Product 1	42
Gambar 32 Page Detail Product 2	42
Gambar 33 Sequence Diagram Melihat Detail Produk	44
Gambar 34 Class Diagram Melihat Detail Produk	44
Gambar 35 Page User	46
Gambar 36 Page Edit User	46
Gambar 37 Sequence Diagram Mengelola Data User	48
Gambar 38 Class Diagram Mengelola User	48
Gambar 39 Page Shopping Cart	49
Gambar 40 Page Success	49
Gambar 41 Sequence Diagram Add to Cart	52
Gambar 42 Class Diagram Add to Cart	52
Gambar 43 Page My Transaction	53
Gambar 44 Page Show My Transaction	54
Gambar 45 Sequence Diagram Melihat History	55
Gambar 46 Class Diagram Melihat History	56
Gambar 47 Class Diagram Klip	57
Gambar 48 Entity Relationship Diagram Klip	60

1. Pendahuluan

1.1. Tujuan Penulisan Dokumen

Deskripsi Perancangan Perangkat Lunak (DPPL) adalah deskripsi perangkat lunak yang dibuat untuk memfasilitasi analisis, perencanaan, implementasi, dan pengambilan keputusan. Penulisan dokumen ini bertujuan untuk memperjelas bagaimana website klip terbentuk dari model perancangan sampai model desain. Dalam dokumen ini akan dijelaskan bagaimana inti dasar dari pengembangan website klip berupa use case diagram yang diperinci dan dipecah menjadi beberapa use case, diagram kelas secara keseluruhan, dan sequence diagram. Dokumen ini juga akan digunakan sebagai dokumen acuan dalam pengimplementasian dan perancangan website klip.

Dokumen ini digunakan oleh seluruh anggota kelompok yang merupakan kelompok pengembangan website klip yang nantinya akan dibangun atau direalisasikan. Dokumen ini juga digunakan oleh stakeholder sebagai dokumen acuan dalam bagaimana website klip akan bekerja dan terbentuk. Kami berharap bahwa dokumen ini dapat membantu kami dalam melakukan pengimplementasian website klip dengan hasil akhir yang cukup memuaskan tidak hanya bagi seluruh anggota kelompok tetapi untuk stakeholder juga.

1.2. Lingkup Masalah

Aplikasi berikut akan dibentuk berbasis website yang dibangun berdasarkan kebutuhan penjual yang belum atau ingin memiliki baju berupa seragam dan beberapa merchandise seperti gantungan kunci, kalung id card, jaket, gelang, stiker, tote bag, dan lainnya. Penjual baru akan merasa kesulitan dalam mendapatkan pelanggan dan tentu membutuhkan tambahan pelanggan. Tujuan dari pembuatan aplikasi berikut supaya penjual mendapatkan keuntungan yang tidak hanya stabil tetapi juga dapat balik modal dengan mendapatkan keuntungan yang meningkat. Akhirnya kami memberikan sebuah solusi dalam mengupayakan peningkatan penjualan tersebut berupa pembuatan Website bernama Klip, website ini diharapkan dapat membantu penjual agar mendapatkan keuntungan yang lebih dan dapat terbangunnya lahan penjualan baru yang lebih efektif dan efisien.

1.3. Definisi dan Istilah

Dalam dokumen ini terdapat beberapa istilah yang tidak digunakan secara umum oleh user diluar kelompok pengembangan. Definisi dari istilah tersebut tercantum dalam tabel berikut:

Backend	Sistem di balik layar yang mengolah database dan server
Frontend	Apa yang dapat pengguna lihat dalam tampilan suatu website atau aplikasi
Login	Proses untuk dapat mengakses fitur dengan

	memasukkan identitas dan password pengguna
Register	Proses pendaftaran suatu program untuk menghubungkan data pribadi dengan program tersebut
User	Aktor yang mengoperasikan atau menggunakan sistem
RAM	Tempat penyimpanan data sementara pada suatu mesin yang sedang berjalan dan dapat diakses secara acak
Sistem Operasi	Sebuah perangkat lunak yang dikembangkan untuk mengoperasikan perangkat keras suatu device
Device	Sebuah perangkat keras yang digunakan untuk mengoperasikan berbagai macam perangkat lunak sesuai spesifikasinya
Website	Sebuah perangkat lunak yang bekerja di dalam platform aplikasi browser

1.4. *Referensi*

- Spesifikasi Kebutuhan Perangkat Lunak (SKPL) Klip
- Software Design Description (SDD) IEEE Standard 1016-2009

2. Perancangan Global

2.1. *Rencana Lingkungan Implementasi*

Perangkat lunak yang kami bangun berupa sebuah website yang dapat diakses menggunakan beberapa software browser seperti Mozilla Firefox, Google Chrome, Safari, Internet Explorer, dan software browser lainnya. Website klip juga dapat digunakan dalam berbagai macam sistem operasi seperti Windows, Linux, Ios, dan Android. Akses menuju website klip juga diperlukan sebuah jaringan koneksi internet yang stabil supaya dalam pengaksesan website akan mendapatkan pengalaman terbaik dalam penggunaan beberapa fitur yang tersedia dari website klip.

Adapun spesifikasi minimal dari perangkat keras user yang perlu diperhatikan dalam pengaksesan website klip:

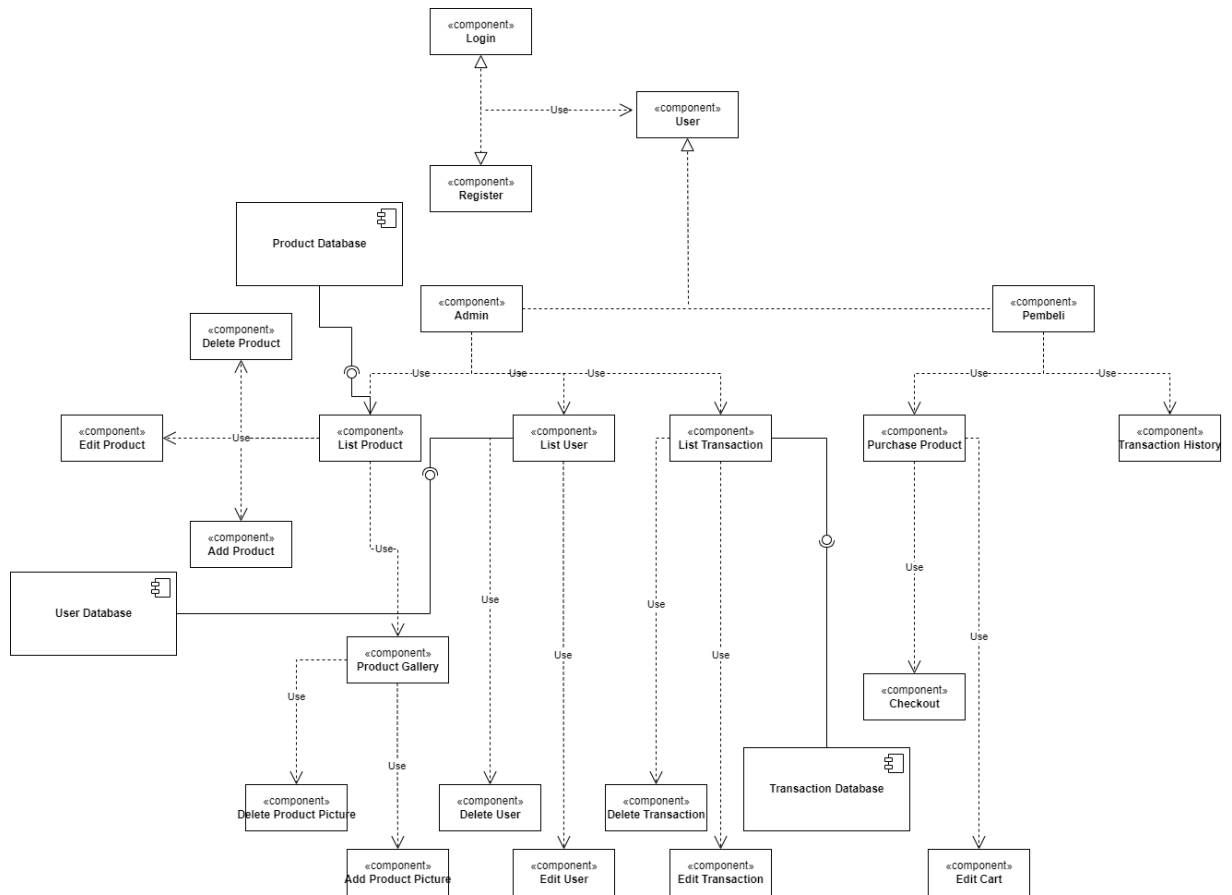
- Sistem Operasi: Windows, Android, Ios, Ubuntu
- RAM: 2 GB
- Processor: Intel Celeron, Intel Pentium, Intel Core i3, Intel Core 2 Duo
- Penyimpanan: SDD, Hard Drive

Perangkat lunak berikut juga akan mendapatkan waktu maintenance yang dimana akan meningkatkan tidak hanya performa tetapi juga kualitas dan kuantitas fitur yang dapat ditambahkan di dalam website klip.

Frontend Development: Tailwind CSS

Backend Development: Laravel, PHP

2.2. Deskripsi Arsitektur Perangkat Lunak

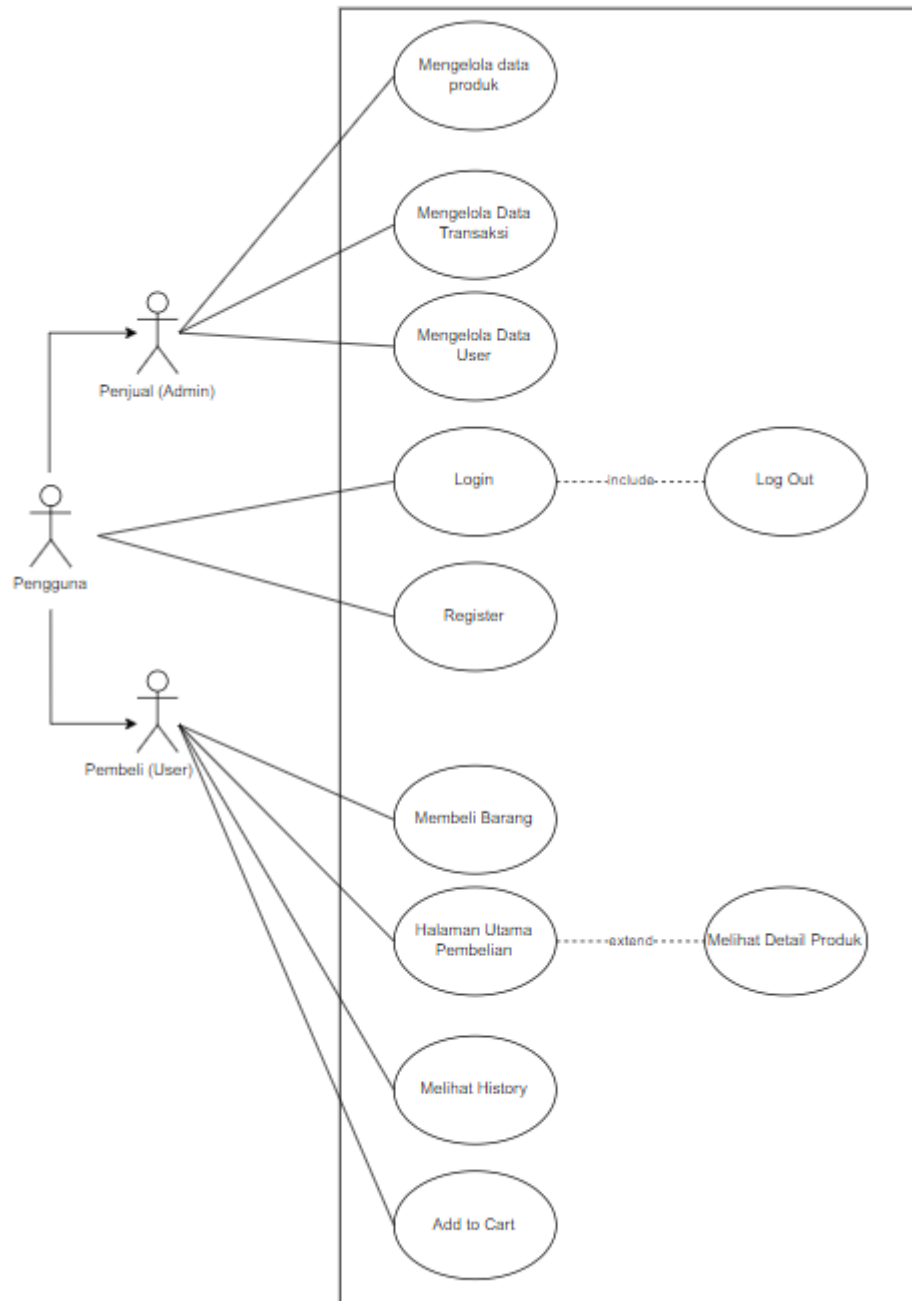


Gambar 1 Arsitektur Perangkat Lunak

3. Perancangan Rinci

3.1. Realisasi Use Case

Kami membuat Use Case Diagram yang menggabungkan dan menggambarkan seluruh use case website klip. Use Case Diagram kami juga dapat digunakan sebagai gambaran apa yang akan dibangun dalam website klip berdasarkan Class Diagram dan Diagram Komponen.



Gambar 2 Use Case Diagram

Berisi TABEL USE CASE sebagai berikut :

No	Nama Use Case	Deskripsi Use Case
#1	Registrasi	Menu pendaftaran akun aplikasi KLIP untuk melakukan login
#2	Login	Menu masuk aplikasi untuk mengakses fitur aplikasi
#3	Logout	Menu untuk mengeluarkan akun dari aplikasi
#4	Mengelola Data Produk	Menu untuk mengelola data produk baik itu tambah, edit, dan hapus produk
#5	Mengelola Data Transaksi	Menu untuk penjual melihat detail dari transaksi dari pembeli dan juga penjual dapat mengubah status transaksi
#6	Membeli Barang	Menu untuk pembeli membeli suatu produk
#7	Halaman Utama Penjualan/Pembelian	Halaman utama yang nantinya berisi berbagai produk yang dijual
#8	Melihat Detail Produk	Halaman untuk pembeli dapat melihat detail dari suatu produk
#9	Mengelola Data User	Menu untuk penjual dapat melihat data user yang terdaftar dan
#10	Add to Cart	Menu untuk pembeli agar dapat menampung terlebih dahulu beberapa produk yang akan dibeli
#11	Melihat History	Halaman untuk melihat history pembelian produk dari seorang user

3.1.1. Use Case #1 Registrasi

3.1.1.1. Use Case Scenario #1 Registrasi

Skenario Use Case #1 :

- i. Pre-Condition: User memiliki email yang belum terdaftar
- ii. Use Case Description
 - a. Primary Flow

No.	Actor	Action Description
1	System	Menampilkan halaman register
2	User	Memasukkan informasi berupa: <ul style="list-style-type: none"> - Nama - Email Address - Password - Confirm Password
3	System	Melakukan pengecekan informasi yang diinputkan user
4	System	Menampilkan halaman dashboard Penjual/Pembeli

b. Alternative Flow

No.	Condition	Action Description
3a	Email yang dimasukkan user sudah terdaftar	System akan menampilkan pesan “Email has already been taken.”
3b	Email tidak komplit	System akan menampilkan pesan “Please enter a part of following ‘@’. ‘x@’ is incomplete.”
3c	Email tidak memiliki ‘@’	System akan menampilkan pesan “Please include an ‘@’ in the email address. ‘x’ is missing an ‘@’.”
3d	Email field kosong	System akan menampilkan pesan “Please fill out this field.”
3e	Password kurang dari 8 character	System akan menampilkan pesan “The password must be at least 8 characters.”

iii. Post-Condition: User berhasil membuat akun

3.1.1.2. UI Design dan Deskripsi Objek UI #1 Registrasi

Gambar 3 Page Registrasi

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>register</i>	<i>Page Registrasi</i>	<i>Halaman registrasi akun untuk pengguna</i>

Page Registrasi

Id_Objek	JENIS	LABEL*	Keterangan**
<i>txt_name</i>	<i>Text Field editable</i>	<i>Name</i>	<i>Text Field yang digunakan untuk memasukkan nama pengguna.</i>
<i>txt_email</i>	<i>Text Field editable</i>	<i>Email</i>	<i>Text field yang digunakan untuk memasukkan email pengguna.</i>
<i>txt_password</i>	<i>Text Field editable</i>	<i>Password</i>	<i>Text field yang digunakan untuk memasukkan password pengguna.</i>
<i>txt_confirm_password</i>	<i>Text Field editable</i>	<i>Confirm Password</i>	<i>Text field yang digunakan untuk memasukkan konfirmasi password pengguna.</i>
<i>btn_already_register</i>	<i>Button clickable</i>	<i>Already Registered?</i>	<i>Jika diklik, akan meredirect ke halaman login</i>
<i>btn_register</i>	<i>Button clickable</i>	<i>Register</i>	<i>Jika diklik akan meredirect ke halaman utama user</i>

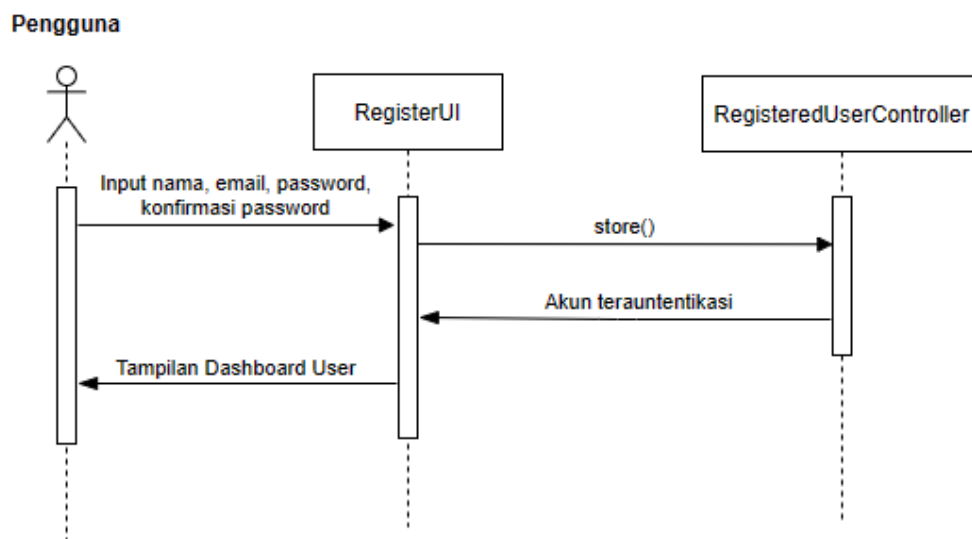
3.1.1.3. Identifikasi Object Baru dan Tipe Kelas #1 Registrasi

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
<i>1</i>	<i>Pengguna</i>	<i>Actor</i>
<i>2</i>	<i>RegisterUI</i>	<i>Boundary</i>
<i>3</i>	<i>UserController</i>	<i>Controller</i>
<i>4</i>	<i>UserRequest</i>	<i>Controller</i>

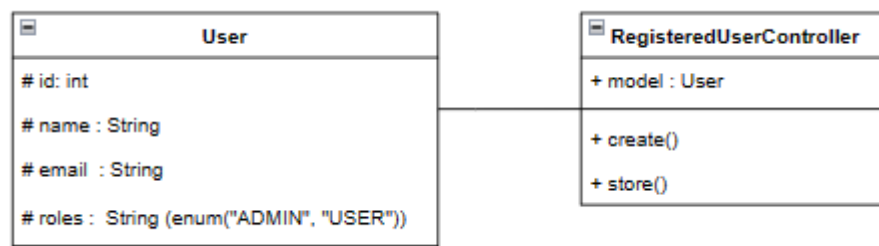
*Tipe kelas diisi dengan *Boundary*(Interface), *Entity*(Database), *Controller*

3.1.1.4. Sequence Diagram #1 Registrasi



Gambar 4 Sequence Diagram Registrasi

3.1.1.5. Class Diagram #1 Registrasi



Gambar 5 Class Diagram Registrasi

3.1.2. Use Case #2 Login

3.1.2.1. Use Case Scenario #2 Login

Skenario Use Case #2 :

- i. Pre-Condition: user memiliki akun yang telah didaftarkan
- ii. Use Case Description
 - a. Primary Flow

No.	Actor	Action Description
1	System	Menampilkan halaman Login
2	User	Memasukkan informasi berupa email address dan password yang telah teregistrasi

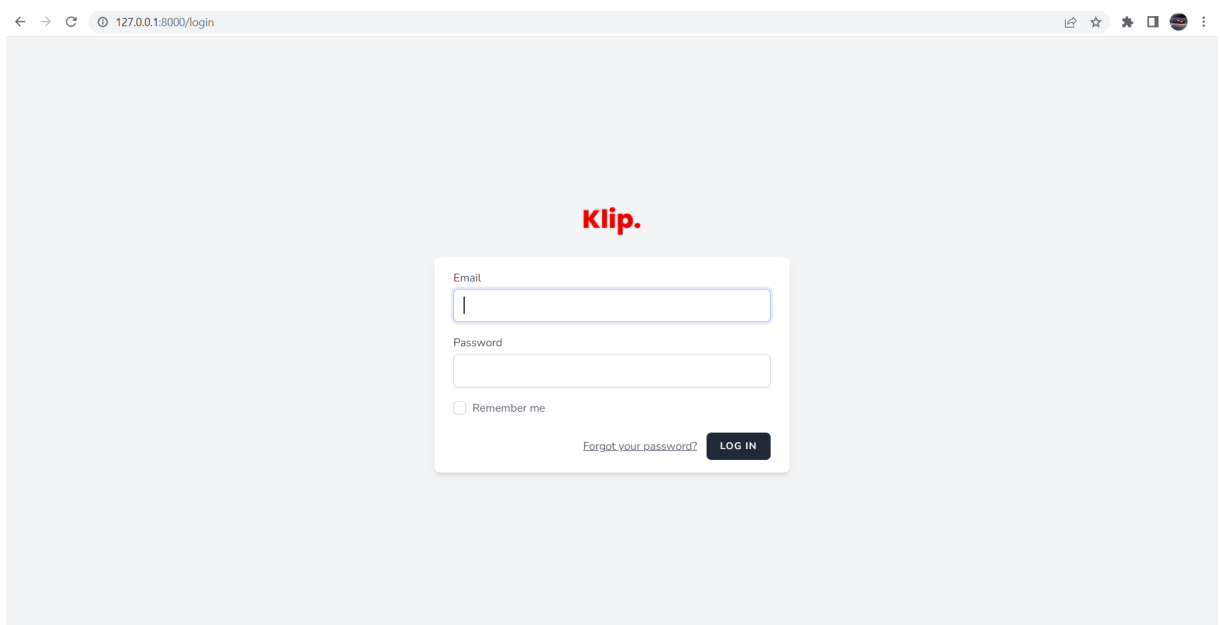
3	System	Mengkonfirmasi data yang dimasukkan dan
4	System	Menampilkan halaman utama

b. Alternative Flow

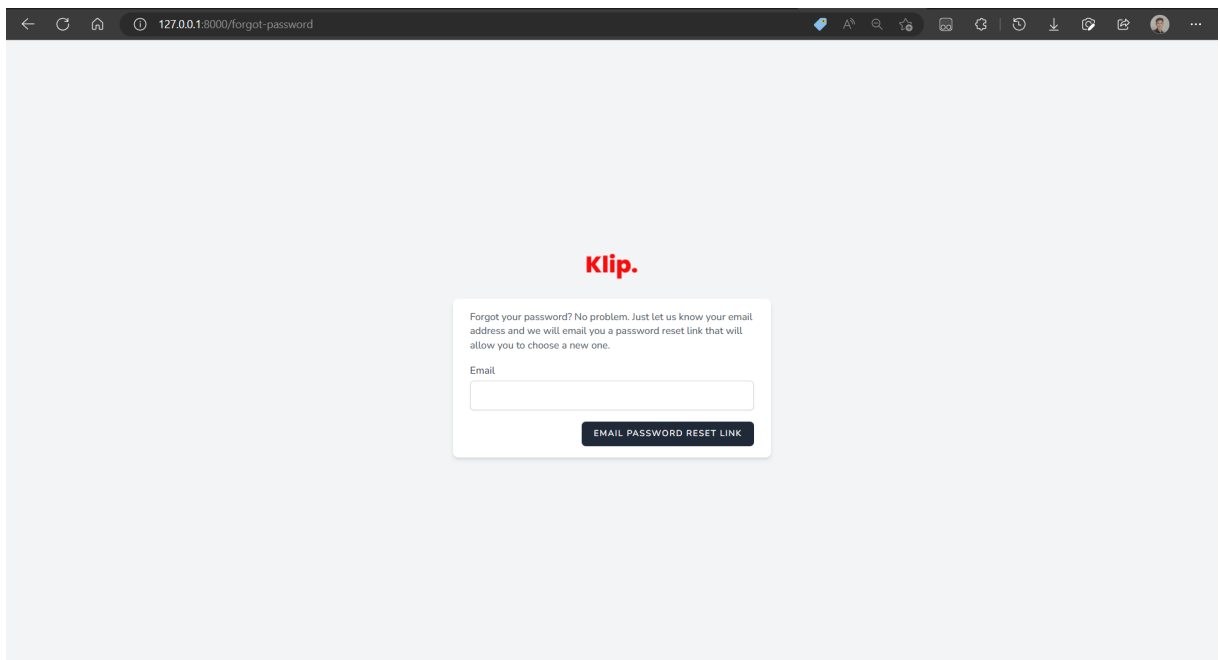
No.	Condition	Action Description
3a	User memasukkan email atau password yang salah	System akan menampilkan pesan “Email atau Password Salah” dan meminta user mengisi kembali email dan password

iii. Post-Condition: User berhasil masuk ke halaman utama

3.1.2.2. UI Design dan Deskripsi Objek UI #2 Login



Gambar 6 Page Login



Gambar 7 Page Forgot Password

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>login</i>	<i>Page Login</i>	<i>Halaman login akun untuk pengguna</i>
<i>forgot_password</i>	<i>Page Forgot Password</i>	<i>Halaman reset password untuk pengguna</i>

Page Login

Id_Objek	JENIS	LABEL*	Keterangan**
<i>txt_email</i>	<i>Text Field editable</i>	<i>Email</i>	<i>Text field yang digunakan untuk memasukkan email pengguna.</i>
<i>txt_password</i>	<i>Text Field editable</i>	<i>Password</i>	<i>Text field yang digunakan untuk memasukkan password pengguna.</i>
<i>check_remember_me</i>	<i>Check Box checkable</i>	<i>Remember Me</i>	<i>Check box yang digunakan untuk mengingat email dan password pengguna di login selanjutnya</i>
<i>btn_login</i>	<i>Button clickable</i>	<i>LogIn</i>	<i>Jika di klik maka akan meredirect ke home page</i>
<i>btn_forgot_password</i>	<i>Button clickable</i>	<i>ForgotPassword</i>	<i>Jika diklik maka akan meredirect ke halaman forgot password</i>

Page Forgot Password

Id_Objek	JENIS	LABEL*	Keterangan**
<i>txt_email</i>	<i>Text Field editable</i>	<i>Email</i>	<i>Text field yang digunakan untuk memasukkan email pengguna yang dipakai untuk mereset password.</i>

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_reset_link</i>	<i>Button clickable</i>	<i>Email Password Reset Link</i>	<i>Jika diklik maka akan dikirimkan link untuk mereset password pada email yang diinputkan</i>

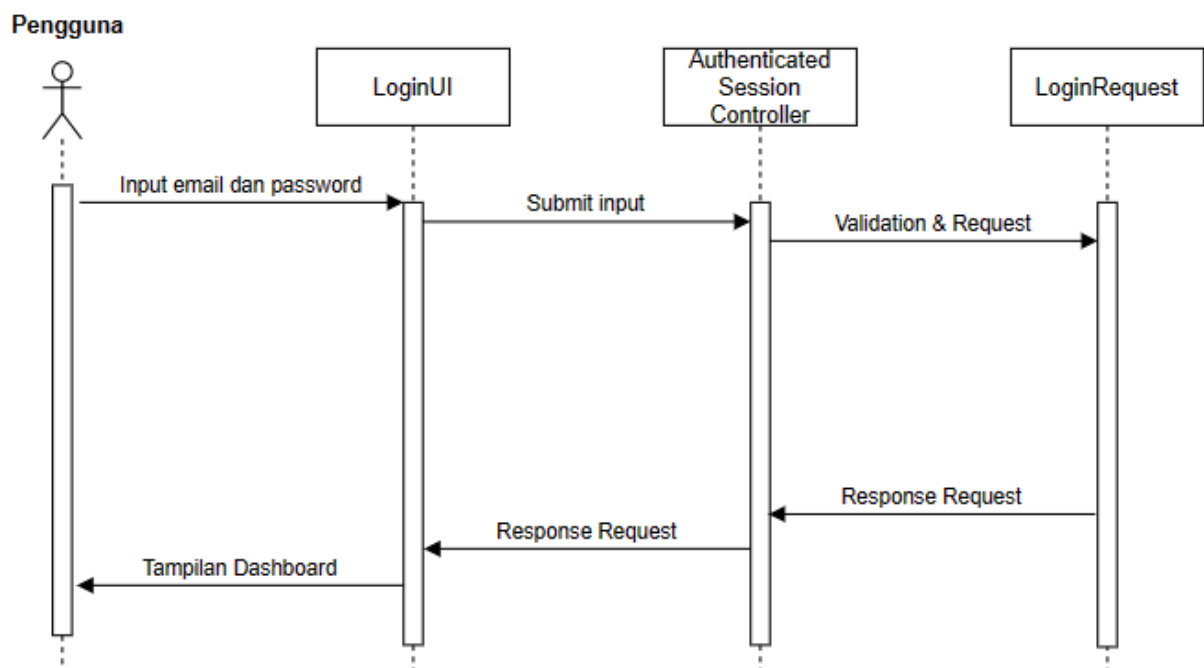
3.1.2.3. Identifikasi Object Baru & Tipe Kelas #2 Login

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	<i>Pengguna</i>	<i>Actor</i>
2	<i>LoginUI</i>	<i>Boundary</i>
3	<i>UserController</i>	<i>Controller</i>
4	<i>UserRequest</i>	<i>Controller</i>

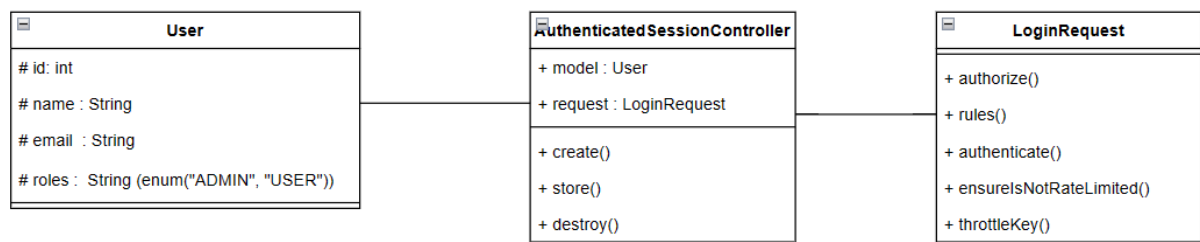
**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

3.1.2.4. Sequence Diagram #2 Login



Gambar 8 Sequence Diagram Login

3.1.2.5. Class Diagram #2 Login



Gambar 9 Class Diagram Registrasi

3.1.3. Use Case #3 Logout

3.1.3.1. Use Case Scenario #3 Logout

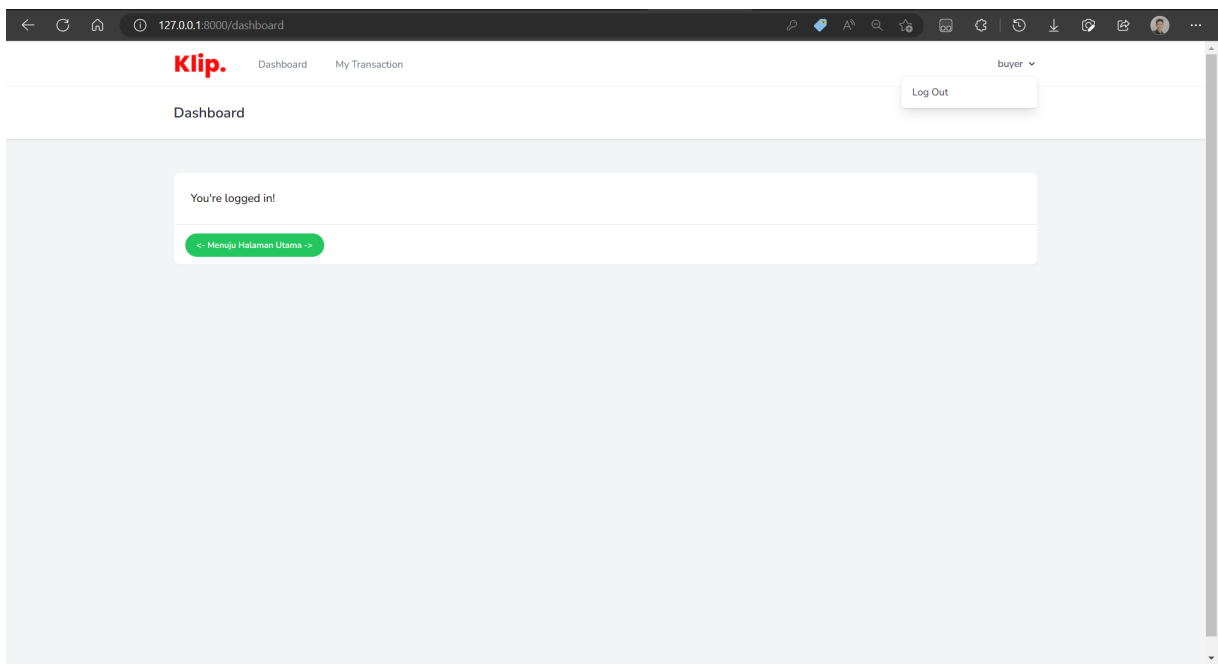
Skenario Use Case #3 :

- i. Pre-Condition: User berhasil masuk ke halaman utama
- ii. Use Case Description
 - a. Primary Flow

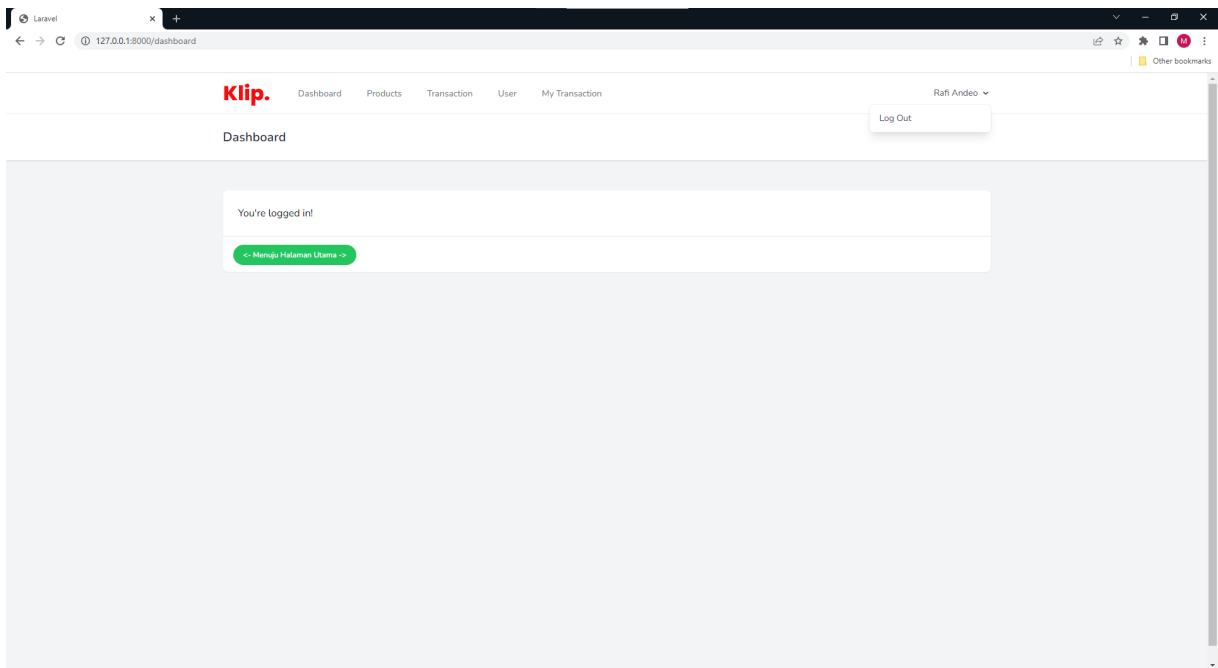
No.	Actor	Action Description
1	User	Menekan tombol logout
2	System	Melakukan logout pada akun User
3	System	Melakukan redirect ke halaman halaman login

- iii. Post-Condition: Berhasil keluar dari akun

3.1.3.2. UI Design dan Deskripsi Objek UI #3 Logout



Gambar 10 Page Dashboard User



Gambar 11 Page Dashboard Admin

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>dashboard user</i>	<i>Page Dashboard User</i>	<i>Halaman dashboard yang dapat mengakses fitur-fitur yang tersedia untuk user</i>
<i>dashboard admin</i>	<i>Page Dashboard Admin</i>	<i>Halaman dashboard yang dapat mengakses fitur-fitur yang tersedia untuk admin</i>

Page Dashboard User

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_logout</i>	<i>Dropdown Button clickable</i>	<i>Log Out</i>	<i>Jika diklik, User akan logout akun dan redirect ke page home</i>
<i>btn_navbar_dashboard</i>	<i>Button clickable</i>	<i>Dashboard</i>	<i>Jika diklik, akan meredirect ke halaman dashboard user</i>
<i>btn_navbar_mytransaction</i>	<i>Button clickable</i>	<i>My Transaction</i>	<i>Jika diklik, akan meredirect ke halaman My Transaction</i>
<i>btn_home</i>	<i>Button clickable</i>	<i>Menuju Halaman Utama</i>	<i>Jika diklik, akan meredirect ke halaman landing page untuk melihat produk-produk yang dijual</i>

Page Dashboard Admin

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_logout</i>	<i>Dropdown Button clickable</i>	<i>Log Out</i>	<i>Jika diklik, User akan logout akun dan redirect ke page home</i>
<i>btn_navbar_dashboard</i>	<i>Button clickable</i>	<i>Dashboard</i>	<i>Jika diklik, akan meredirect ke halaman dashboard admin</i>
<i>btn_navbar_product</i>	<i>Button clickable</i>	<i>Product</i>	<i>Jika diklik, akan meredirect ke halaman mengelola data produk</i>
<i>btn_navbar_transaction</i>	<i>Button clickable</i>	<i>Transaction</i>	<i>Jika diklik, akan meredirect ke halaman mengelola data transaksi</i>
<i>btn_navbar_user</i>	<i>Button clickable</i>	<i>User</i>	<i>Jika diklik, akan meredirect ke halaman mengelola data user</i>
<i>btn_home</i>	<i>Button clickable</i>	<i>Menuju Halaman Utama</i>	<i>Jika diklik, akan meredirect ke halaman landing page untuk melihat produk-produk yang dijual</i>

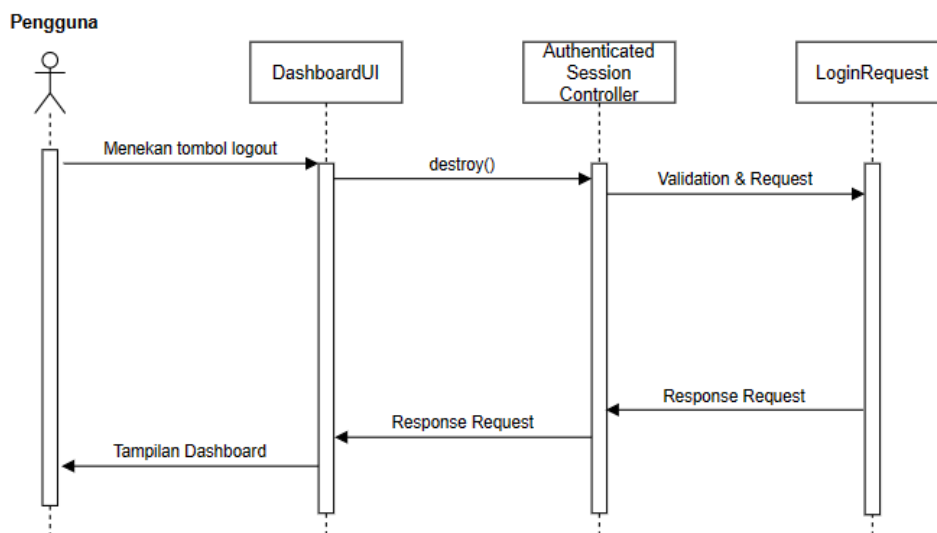
3.1.3.3. Identifikasi Object Baru & Tipe Kelas #3 Logout

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	User	Entity
2	DashboardUI	Boundary
3	UserController	Controller
3	UserRequest	Controller

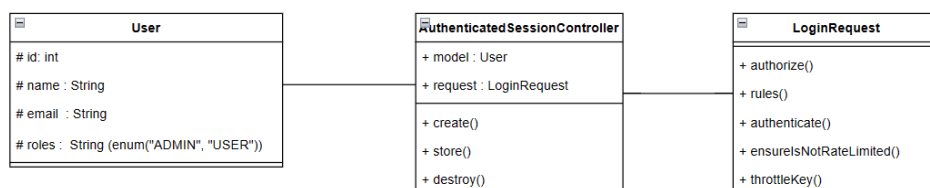
*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.3.4. Sequence Diagram #3 Logout



Gambar 12 Class Diagram Logout

3.1.3.5. Class Diagram #3 Logout



Gambar 13 Class Diagram Logout

3.1.4. Use Case #4 Mengelola Data Produk

3.1.4.1. Use Case Scenario #4 Mengelola Data Produk

Skenario Use Case #4 :

- i. Pre-Condition: Penjual Login dan masuk dashboard admin/penjual
- ii. Use Case Description
 - a. Primary Flow

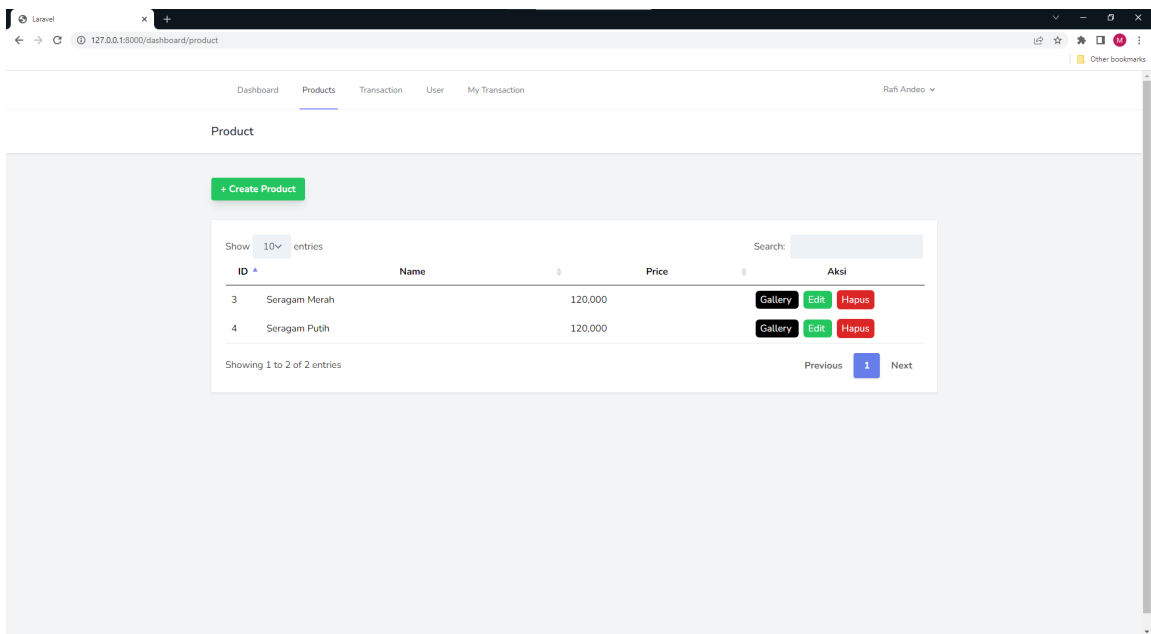
No.	Actor	Action Description
1	Penjual	Menuju ke halaman produk
2	System	Menampilkan data produk
3	Penjual	Melakukan create product
4	System	Memindahkan ke halaman create product
5	Penjual	Mengisi form name, description, dan price lalu tekan tombol save product
6	System	Melakukan tambah data produk & menampilkan data produk
7	Penjual	Menekan hapus pada list data produk
8	System	Melakukan hapus produk & menghilangkan di halaman produk
9	Penjual	Menekan edit pada list data produk
10	System	Memindahkan ke halaman edit produk
11	Penjual	Mengubah form data produk & melakukan save product
12	System	Melakukan edit produk dan menampilkan di halaman produk

b. Alternative Flow

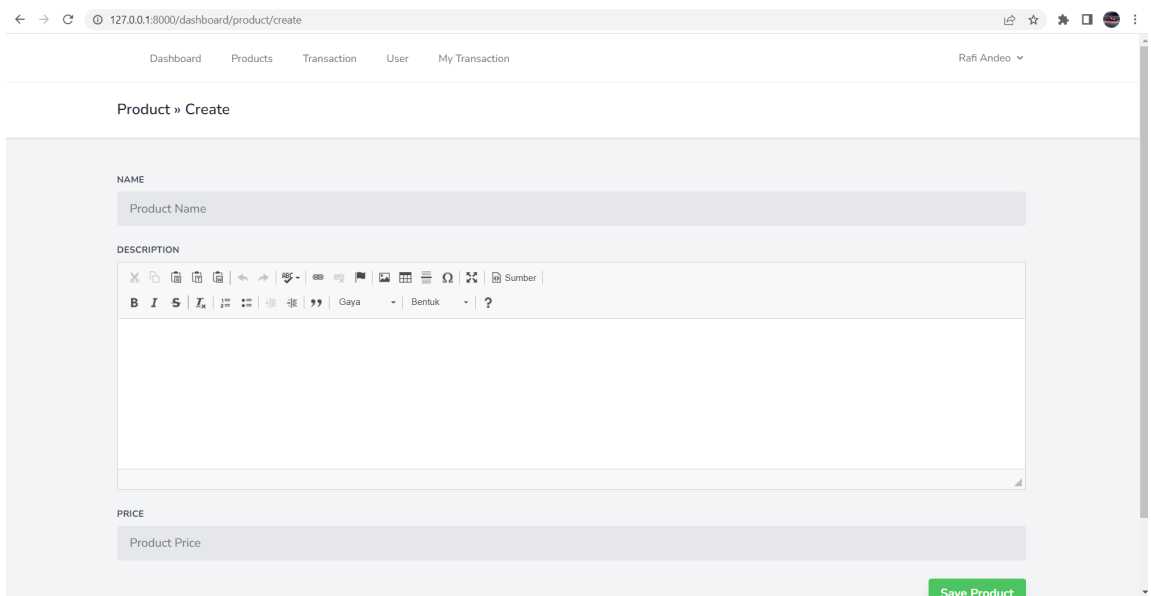
No.	Condition	Action Description
5a	User tidak mengisi form tambah produk	System akan menampilkan pesan/ alert “Harap Mengisi form”

- iii. Post-Condition: System berhasil menampilkan produk, tambah data produk, edit data produk, delete data produk

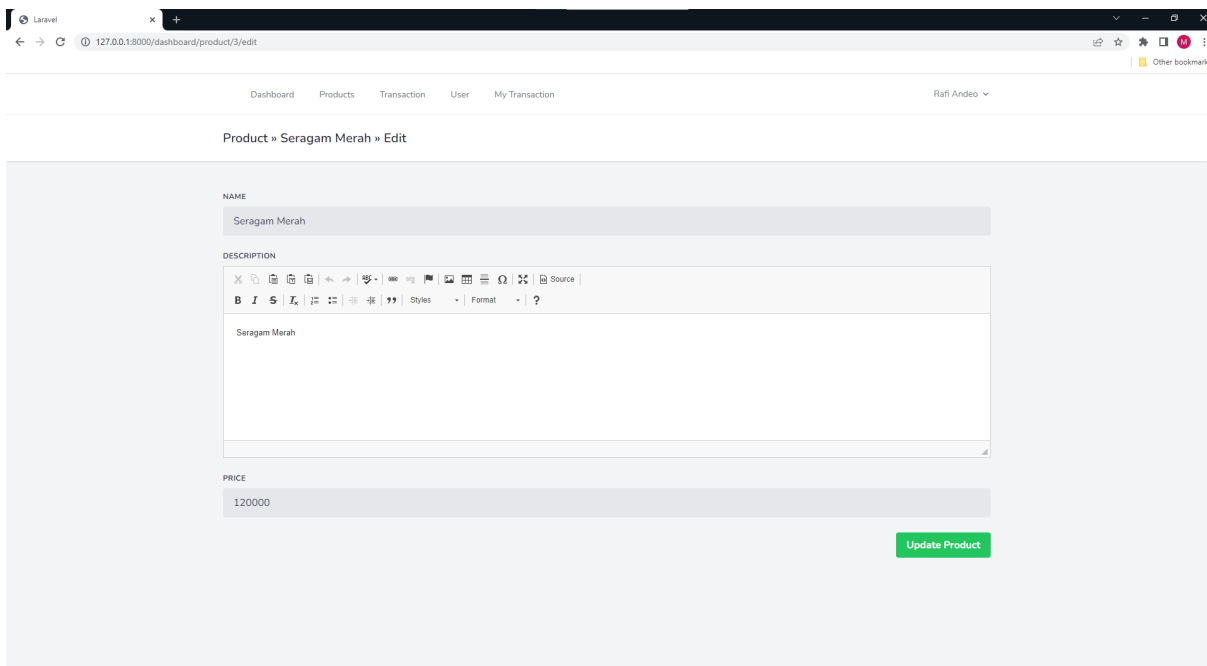
3.1.4.2. UI Design dan Deskripsi Objek UI #4 Mengelola Data Produk



Gambar 14 Page Product



Gambar 15 Page Create Product



Gambar 16 Page Edit Product

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>dashboard_product</i>	<i>Page Product</i>	<i>Halaman untuk menampilkan daftar produk beserta keterangan dan beberapa aksi untuk edit, hapus, dan menambah produk</i>
<i>dashboard_product_create</i>	<i>Page Create Product</i>	<i>Halaman untuk menambahkan dan membuat produk baru</i>
<i>dashboard_product_edit</i>	<i>Page Edit Product</i>	<i>Halaman untuk melakukan edit product berupa nama, harga, dan deskripsi produk</i>

Page Product

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_create</i>	<i>Button clickable</i>	<i>Create Product</i>	<i>Jika diklik, akan redirect ke page create product</i>
<i>btn_hapus</i>	<i>Button clickable</i>	<i>Hapus</i>	<i>Jika diklik akan menghapus product</i>
<i>btn_edit</i>	<i>Button clickable</i>	<i>Edit</i>	<i>Jika diklik akan mengaktifkan fitur mengedit informasi produk</i>

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_gallery</i>	<i>Button clickable</i>	<i>Gallery</i>	<i>Jika diklik akan menampilkan deskripsi produk</i>

Page Create Product

Id_Objek	JENIS	LABEL*	Keterangan**
<i>txt_name</i>	<i>Text Field editable</i>	<i>Name</i>	<i>Text field yang digunakan untuk mengisi nama produk</i>
<i>txt_description</i>	<i>Text Field editable</i>	<i>Description</i>	<i>Text field yang digunakan untuk mengisi deskripsi produk</i>
<i>txt_price</i>	<i>Text Field editable</i>	<i>Price</i>	<i>Text field yang digunakan untuk mengisi harga produk</i>
<i>btn_save</i>	<i>Button clickable</i>	<i>Save Product</i>	<i>Jika di klik akan menyimpan dan menambahkan produk ke dalam daftar produk</i>

Page Edit Product

Id_Objek	JENIS	LABEL*	Keterangan**
<i>txt_name</i>	<i>Text Field editable</i>	<i>Name</i>	<i>Text field yang digunakan untuk mengisi nama produk</i>
<i>txt_description</i>	<i>Text Field editable</i>	<i>Description</i>	<i>Text field yang digunakan untuk mengisi deskripsi produk</i>
<i>txt_price</i>	<i>Text Field editable</i>	<i>Price</i>	<i>Text field yang digunakan untuk mengisi harga produk</i>
<i>btn_update</i>	<i>Button clickable</i>	<i>Update Product</i>	<i>Jika di klik akan menyimpan dan menambahkan produk ke dalam daftar produk</i>

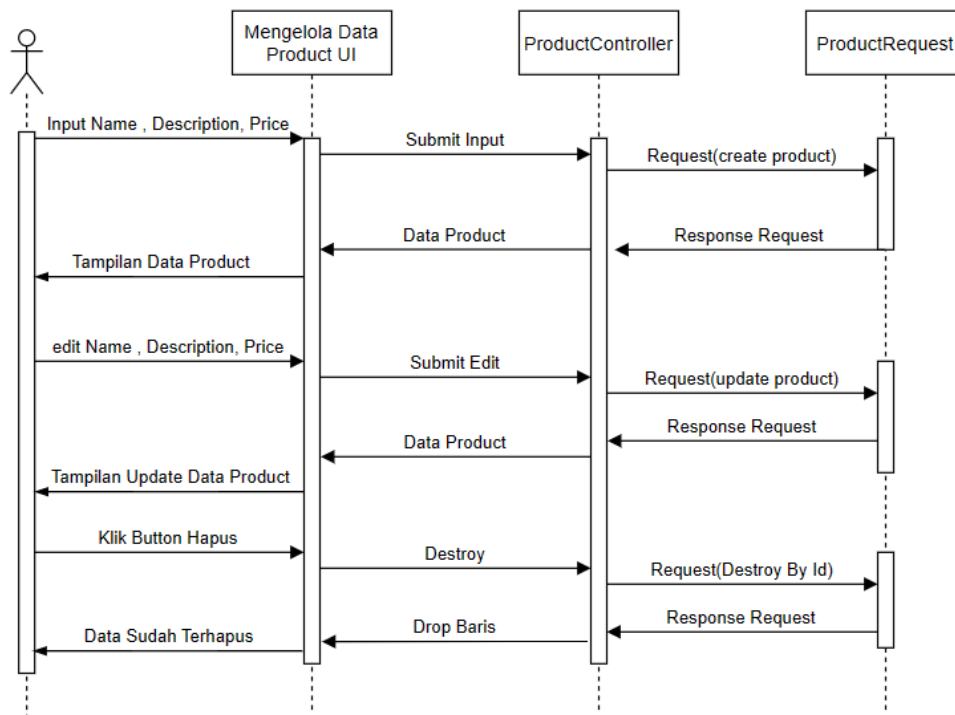
3.1.4.3. Identifikasi Object Baru & Tipe Kelas #4 Mengelola Data Produk

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
<i>1</i>	<i>User</i>	<i>Entity</i>
<i>2</i>	<i>Mengelola Data Product UI</i>	<i>Boundary</i>
<i>3</i>	<i>ProductController</i>	<i>Controller</i>
<i>4</i>	<i>ProductRequest</i>	<i>Controller</i>

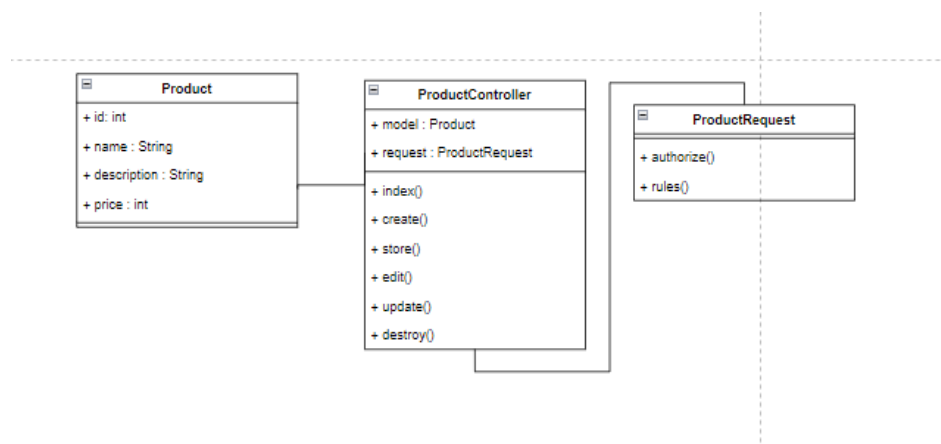
**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

3.1.4.4. Sequence Diagram #4 Mengelola Data Produk



Gambar 17 Sequence Diagram Mengelola Data Produk

3.1.4.5. Class Diagram #4 Mengelola Data Produk



Gambar 18 Class Diagram Mengelola Data Produk

3.1.5. Use Case #5 Mengelola Data Transaksi

3.1.5.1. Use Case Scenario #5 Mengelola Data Transaksi

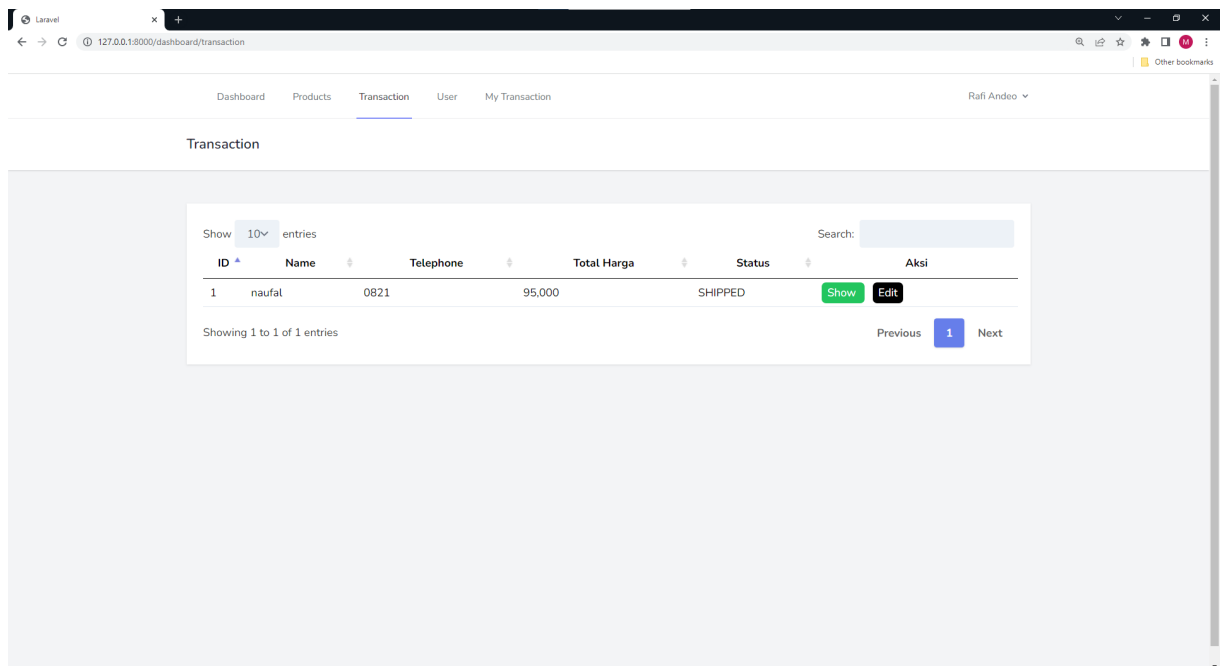
Skenario Use Case #5 :

- i. Pre-Condition: Penjual telah melakukan login
- ii. Use Case Description
 - a. Primary Flow

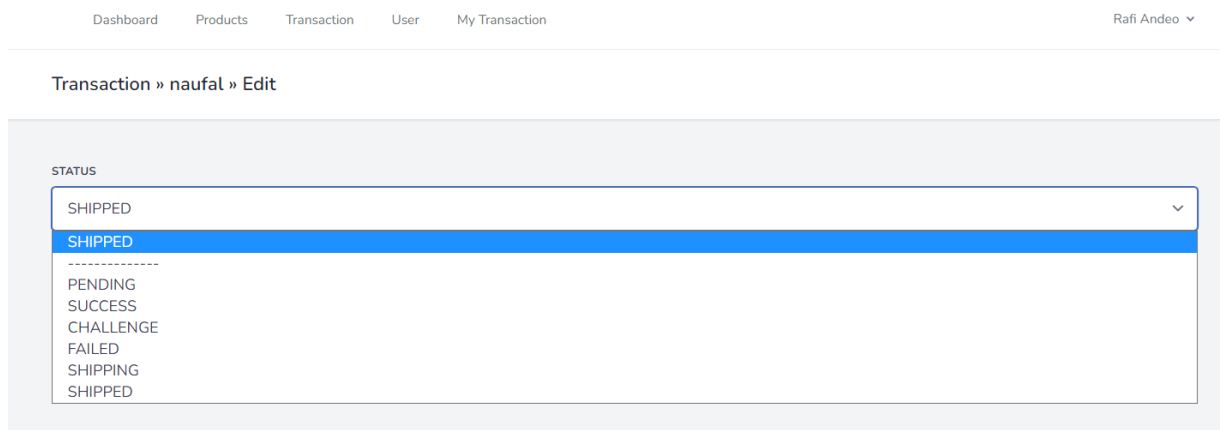
No.	Actor	Action Description
1	Penjual	Mengklik bagian navbar transaksi & halaman transaksi
2	System	Menampilkan halaman transaksi
3	Penjual	Menekan tombol edit transaksi
4	System	Menampilkan halaman edit transaksi pembelian
5	Penjual	Mengubah Status yang terdapat pada transaksi
6.	System	Melakukan perubahan Status
7.	Penjual	Menekan tombol show
8.	System	Melakukan redirect ke halaman transaction detail
9.	Penjual	Melihat detail transaction

- iii. Post-Condition: Detail transaksi dapat ditampilkan di halaman transaksi dan penjual dapat mengubah status transaksi

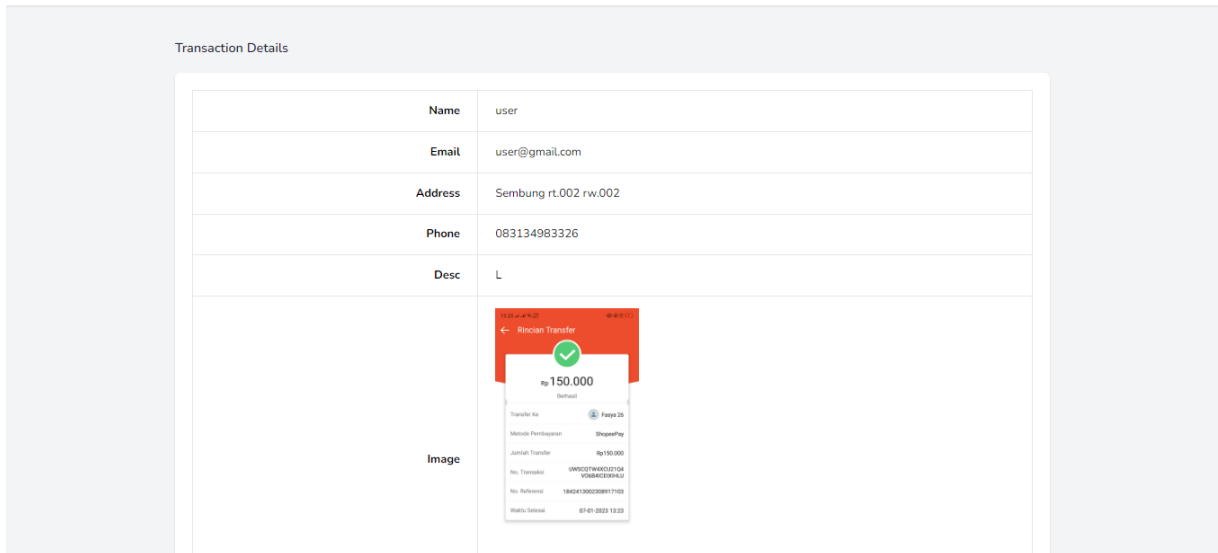
3.1.5.2. UI Design dan Deskripsi Objek UI #5 Mengelola Data Transaksi



Gambar 19 Page Transaction



Gambar 20 Page Edit Transaction



Gambar 21 Page Show Transaction

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>dashboard_transaction</i>	<i>Page Transaction</i>	<i>Halaman untuk menampilkan daftar transaksi dari seluruh pembeli dengan beberapa detail dan fungsi untuk edit</i>
<i>dashboard_edit_transaction</i>	<i>Page Edit Transaction</i>	<i>Halaman untuk mengubah status transaksi pembeli</i>
<i>dashboard_detail_transaction</i>	<i>Page Show Transaction</i>	<i>Halaman untuk menampilkan detail pembelian berupa product , jumlah ,total harga, data diri dan bukti pembayaran dari pembeli</i>

Page Transaction

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_edit</i>	<i>Button clickable</i>	<i>Edit</i>	<i>Jika diklik, akan meredirect ke halaman</i>
<i>btn_show</i>	<i>Button clickable</i>	<i>Show</i>	<i>Jika diklik, akan meredirent ke halaman</i>

Page Edit Transaction

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_edit</i>	<i>Button clickable</i>	<i>Edit</i>	<i>Jika diklik, akan meredirect ke halaman</i>

Page Show Transaction

Id_Objek	JENIS	LABEL*	Keterangan**
-	-	-	-

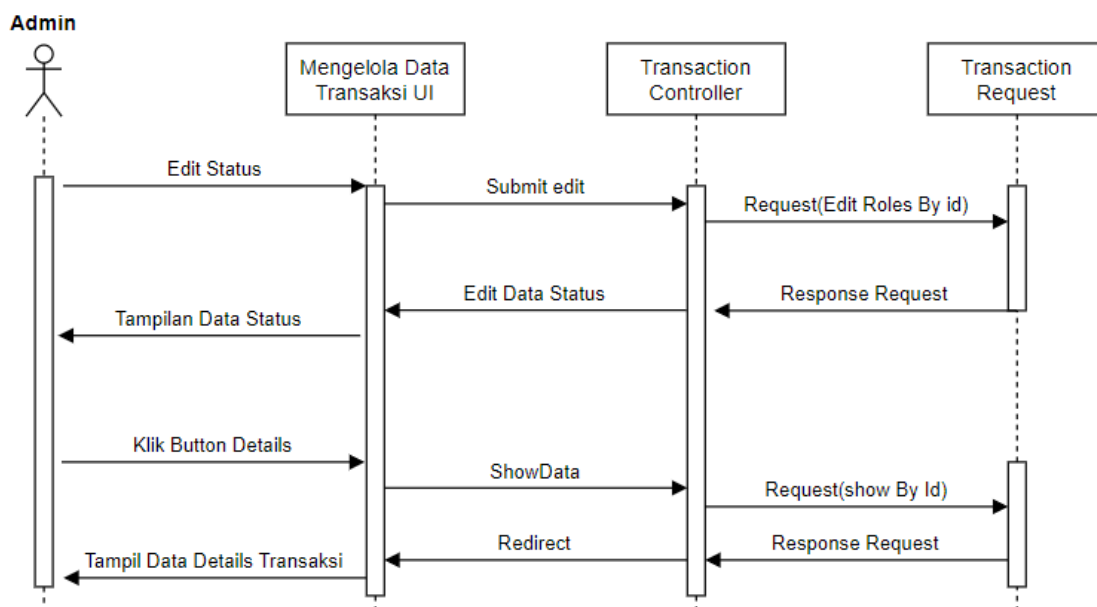
3.1.5.3. Identifikasi Object Baru & Tipe Kelas #5 Mengelola Data Transaksi

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	<i>Admin</i>	<i>Actor</i>
2	<i>Mengelola Data Transaksi UI</i>	<i>Boundary</i>
3	<i>TransactionController</i>	<i>Controller</i>
4	<i>TransactionRequest</i>	<i>Controller</i>

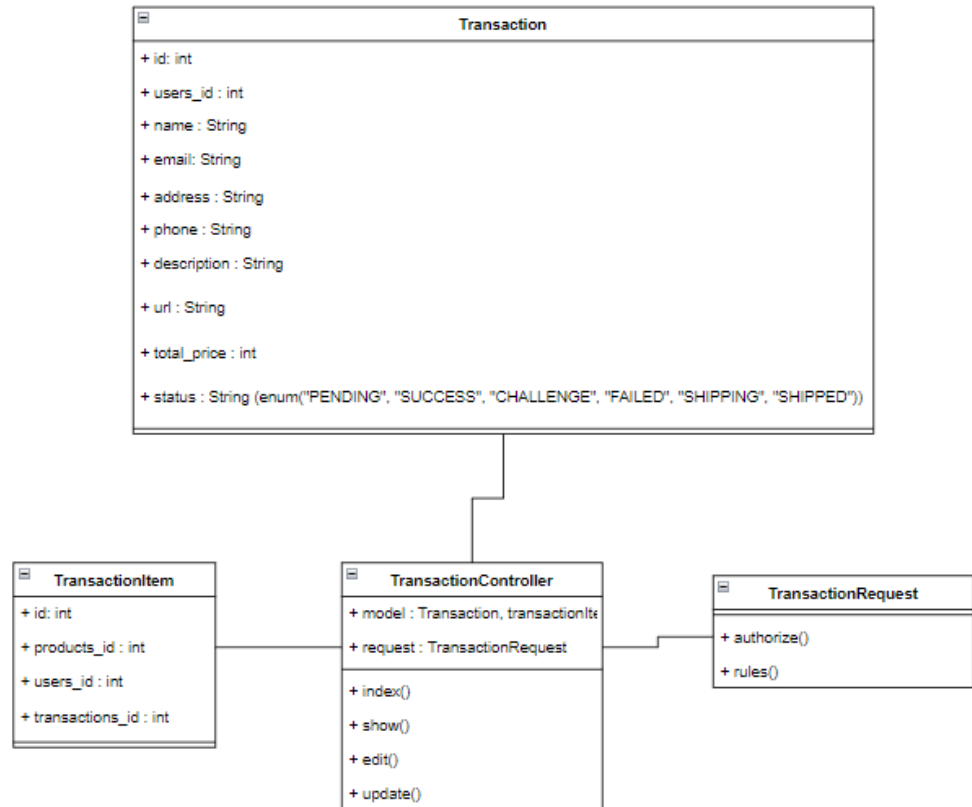
*Tipe kelas seperti *Boundary(Interface)*, *Entity(Database)*, *Controller*

3.1.5.4. Sequence Diagram #5 Mengelola Data Transaksi



Gambar 22 Sequence Diagram Mengelola Data Transaksi

3.1.5.5. Class Diagram #5 Mengelola Data Transaksi



Gambar 23 Class Diagram Mengelola Data Transaksi

3.1.6. Use Case #6 Membeli Barang

3.1.6.1. Use Case Scenario #6 Membeli Barang

Skenario Use Case #6 :

- i. Pre-Condition : Pembeli berhasil melakukan login
- ii. Use Case Description
 - a. Primary Flow

No	Actor	Action Description
1	System	Menampilkan halaman konfirmasi pembelian
2	Pembeli	Mengisi data diri berupa : <ul style="list-style-type: none"> ● Nama Lengkap ● Email ● Alamat

		<ul style="list-style-type: none"> • Nomor Telepon • Deskripsi / ukuran • Bukti Pembayaran Lalu menekan tombol checkout now.
3	System	Memindahkan halaman menuju ke halaman sukses


iii. Post-Condition : Pembeli berhasil melakukan pembelian barang

3.1.6.2. UI Design dan Deskripsi Objek UI #6 Membeli Barang

Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya.

Home / Shopping Cart

Shopping Cart

Photo	Product	Price	Action
	Seragam Telkom Merah Office Room	IDR 120,000	X

Detail Pembelian

Nama Lengkap

Email

Alamat

Nomor telepon

Deskripsi / Ukuran

Upload Bukti
 No file chosen

Gambar 24 Page Transaction

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>transaction_page</i>	<i>Page Pembelian</i>	<i>Halaman untuk melakukan pembelian/pembayaran yang terdapat product yang dibeli dan inputan yang harus di isi oleh pembeli untuk melengkapi data</i>

Page Pembelian

Id_Objek	JENIS	LABEL*	Keterangan**
<i>field_nama_lengkap</i>	<i>Text Field editable</i>	<i>Nama Lengkap</i>	<i>Untuk memasukkan nama pembeli</i>

Id_Objek	JENIS	LABEL*	Keterangan**
<i>field_email</i>	<i>Text Field editable</i>	<i>Email</i>	<i>Untuk memasukkan email pembeli</i>
<i>field_alamat</i>	<i>Text Field editable</i>	<i>Alamat</i>	<i>Untuk memasukkan email pembeli</i>
<i>field_nohp</i>	<i>Text Field editable</i>	<i>No telepon</i>	<i>Untuk memasukkan nomor telepon pembeli</i>
<i>field_desc</i>	<i>Text Field editable</i>	<i>Description</i>	<i>Untuk memasukkan deskripsi atau size yang dipilih pembeli</i>
<i>file_uploaded_bukti</i>	<i>File uploadable</i>	<i>Upload Bukti</i>	<i>Untuk mengunggah bukti pembayaran yang telah dilakukan oleh pembeli</i>
<i>button_checkout_now</i>	<i>Button clickable</i>	<i>Checkout Now</i>	<i>Untuk submit data yang telah diinputkan oleh pembeli</i>

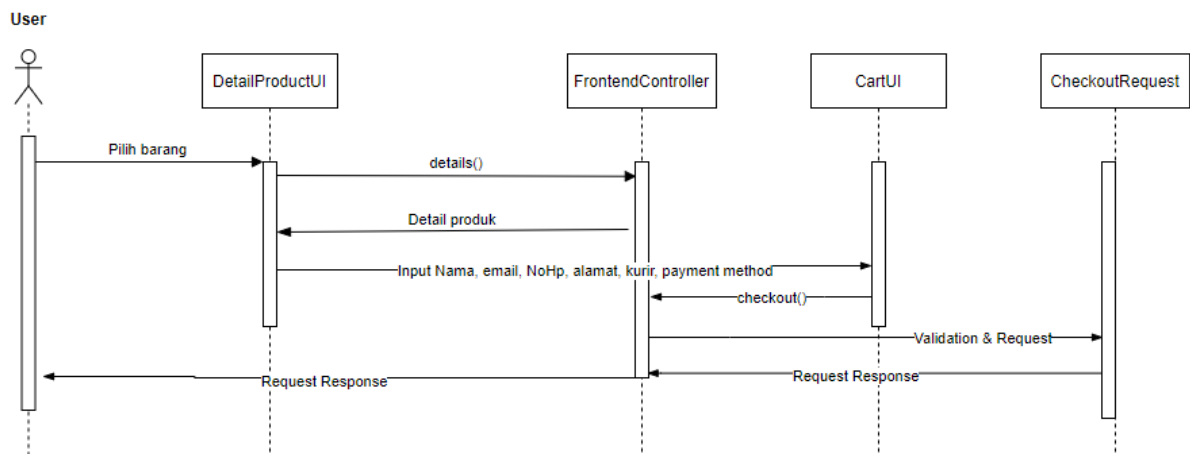
3.1.6.3. Identifikasi Object Baru & Tipe Kelas #6 Membeli Barang

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
<i>1</i>	<i>User</i>	<i>Actor</i>
<i>2</i>	<i>DetailProductUI</i>	<i>Boundary</i>
<i>3</i>	<i>CartUI</i>	<i>Boundary</i>
<i>4</i>	<i>FrontendController</i>	<i>Controller</i>
<i>5</i>	<i>CheckoutRequest</i>	<i>Controller</i>

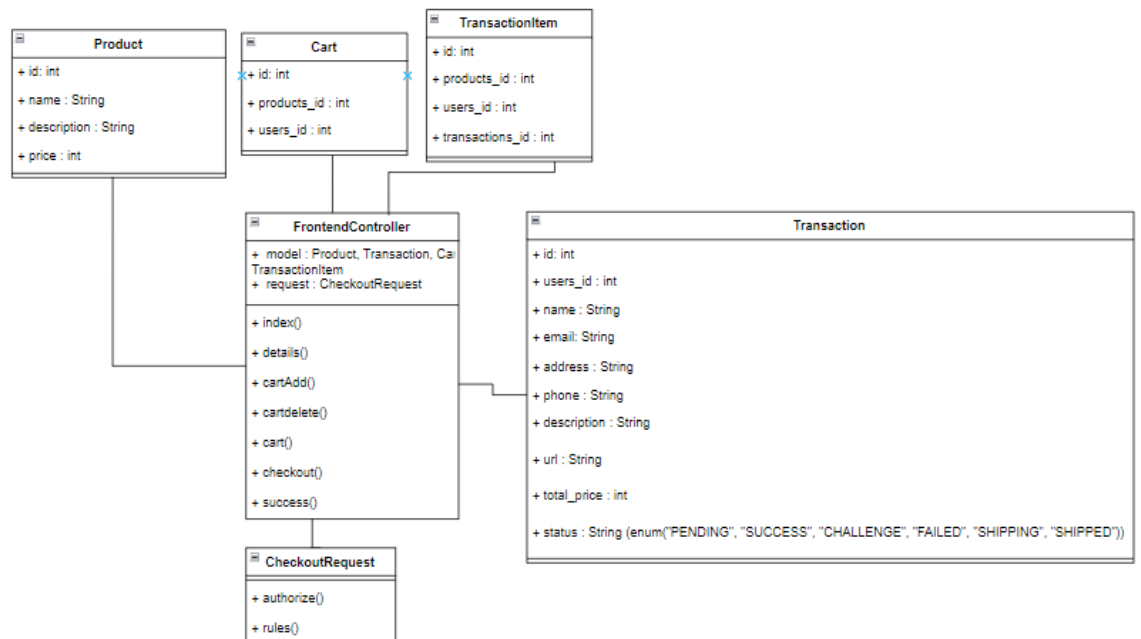
**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

3.1.6.4. Sequence Diagram #6 Membeli Barang



Gambar 25 Sequence Diagram Membeli Barang

3.1.6.5. Class Diagram #6 Membeli Barang



Gambar 26 Class Diagram Membeli Barang

3.1.7. Use Case #7 Halaman Utama Penjualan/Pembelian

3.1.7.1. Use Case Scenario #7 Halaman Utama Pembelian

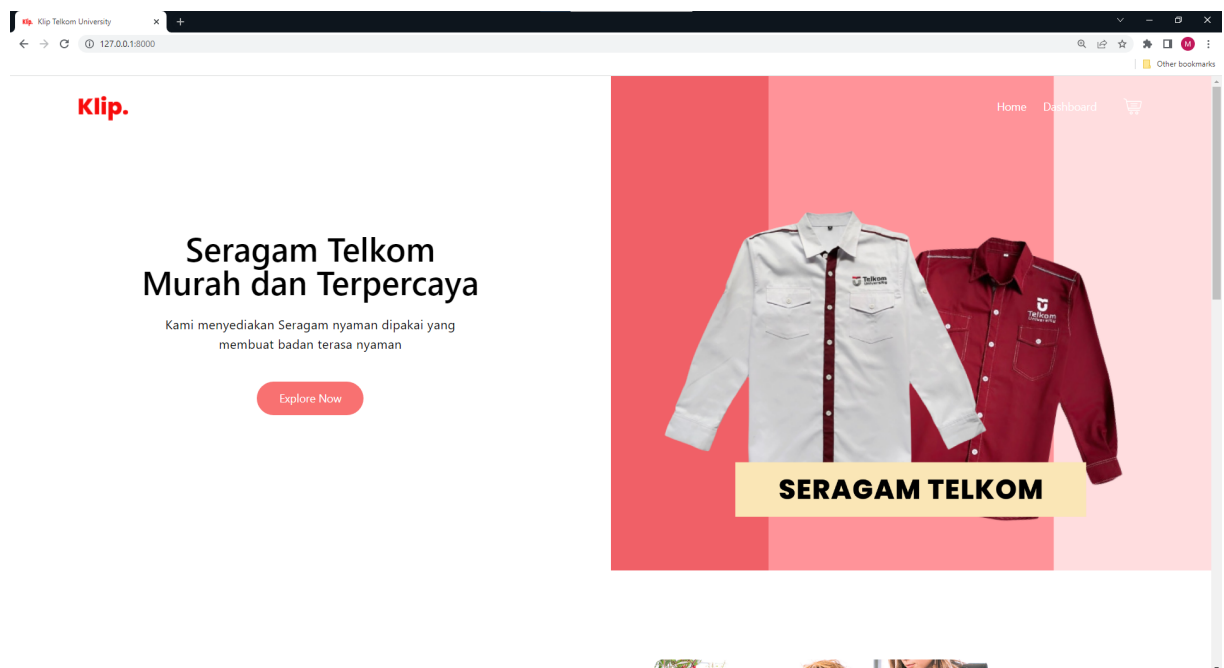
Skenario Use Case #7 :

- i. Pre-Condition: Pengguna sudah melakukan Registrasi dan Log In
- ii. Use Case Description
 - a. Primary Flow

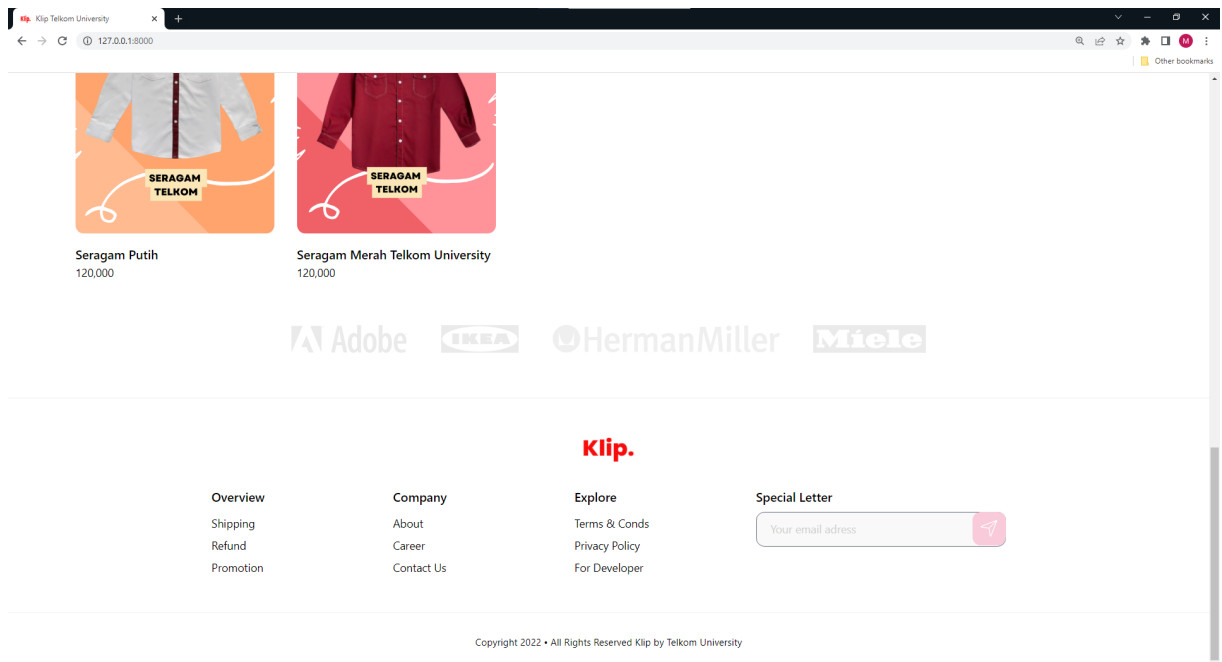
No.	Actor	Action Description
1	System	Menampilkan halaman utama pembeli
2	User	Melihat dan mengakses halaman utama pembelian

- iii. Post-Condition: Pengguna dapat mengakses halaman utama penjual/pembeli

3.1.7.2. UI Design dan Deskripsi Objek UI #7 Halaman Utama Penjualan/Pembelian



Gambar 27 Page Home



Gambar 28 Page Home

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
home	Page Home	Halaman landing page saat User pertama kali membuka web Klip

Page Home

Id_Objek	JENIS	LABEL*	Keterangan**
navbar_dashboard	Button clickable	Dashboard	Jika di klik, akan mengalihkan atau memindahkan ke halaman dashboard pembeli/penjual sesuai role masing-masing
card	Container clickable	Seragam Putih / Seragam merah Telkom University	Jika diklik akan memindahkan ke halaman detail product sesuai product yang dipilih pembeli

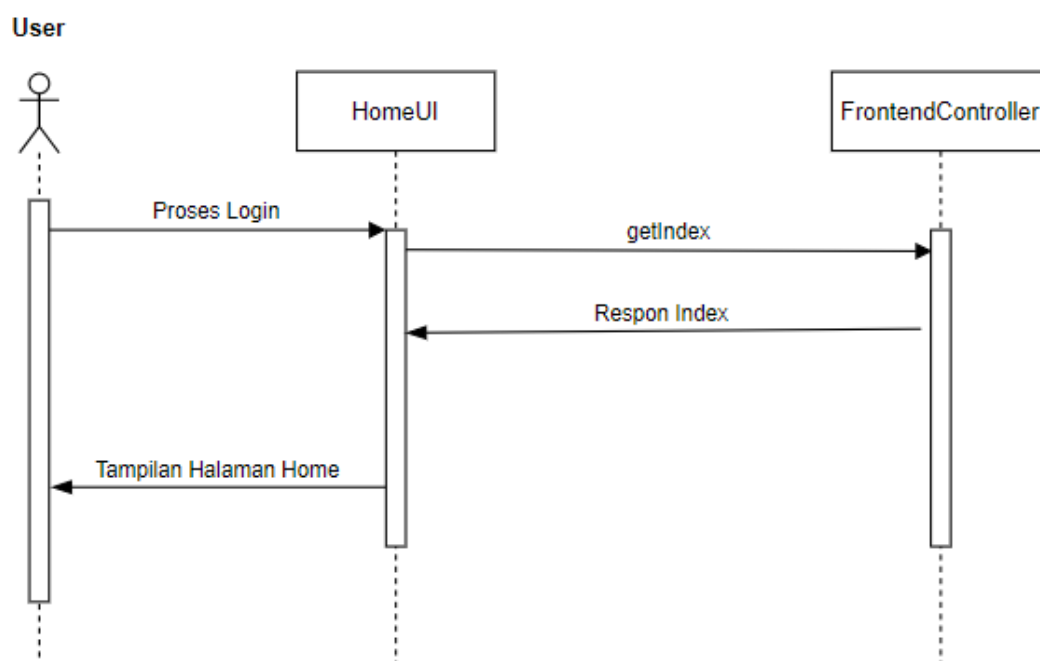
3.1.7.3. Identifikasi Object Baru & Tipe Kelas #7 Halaman Utama Penjualan/Pembelian

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	User	Actor
2	HomeUI	Boundary
3	FrontendController	Controller

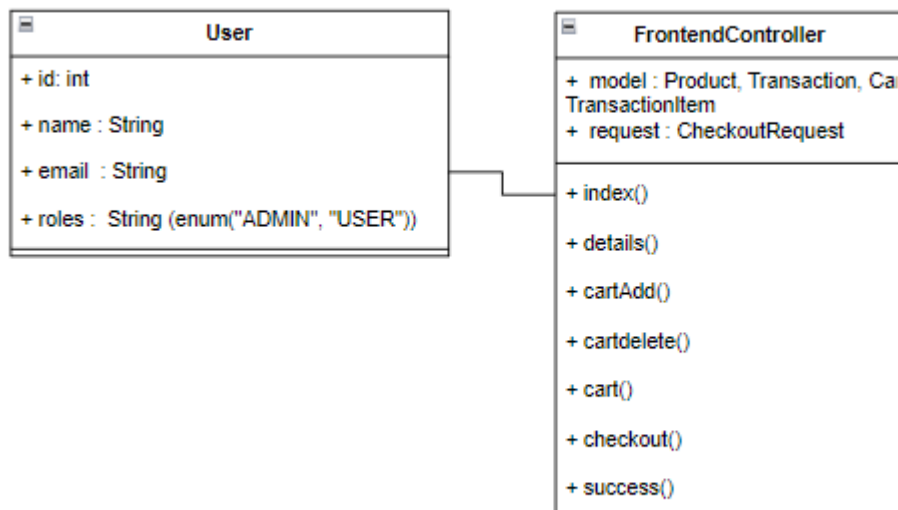
**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

3.1.7.4. Sequence Diagram #7 Halaman Utama Penjualan/Pembelian



Gambar 29 Sequence Diagram Halaman Utama Penjualan/Pembelian

3.1.7.5. Class Diagram #7 Halaman Utama Penjualan/Pembelian



Gambar 30 Class Diagram Halaman Utama Penjualan/Pembelian

3.1.8. Use Case #8 Melihat Detail Produk

3.1.8.1. Use Case Scenario #8 Melihat Detail Produk

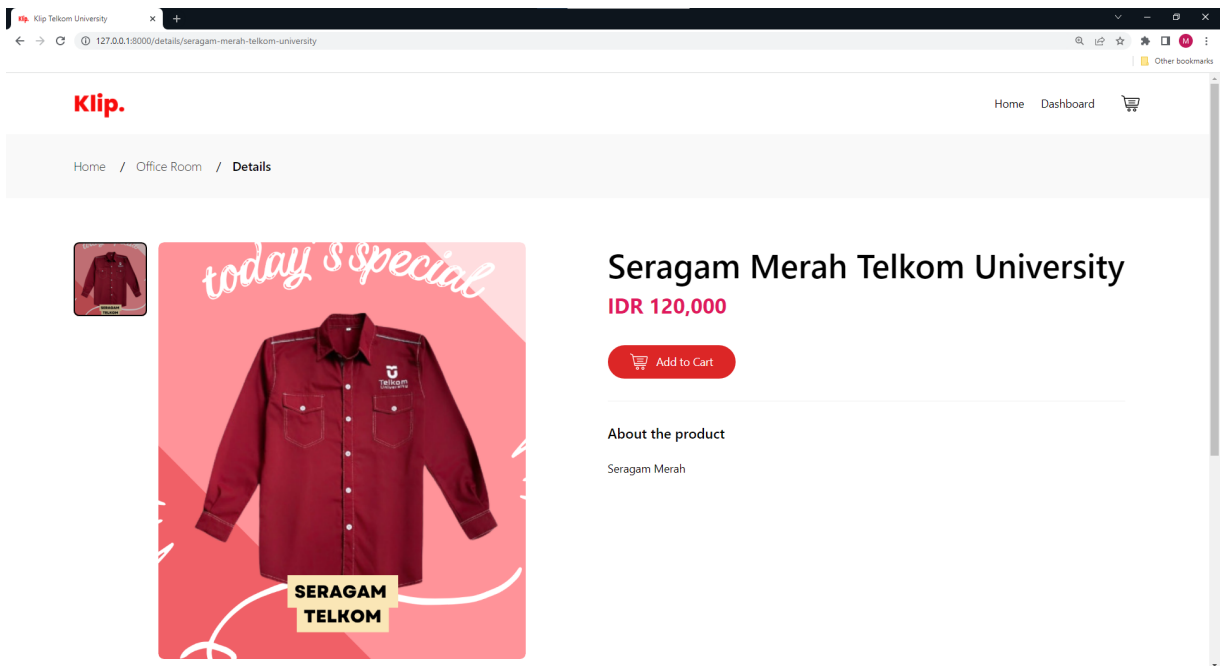
Skenario Use Case #8 :

- i. Pre-Condition : Pembeli berhasil masuk ke halaman utama
- ii. Use Case Description
 - a. Primary Flow

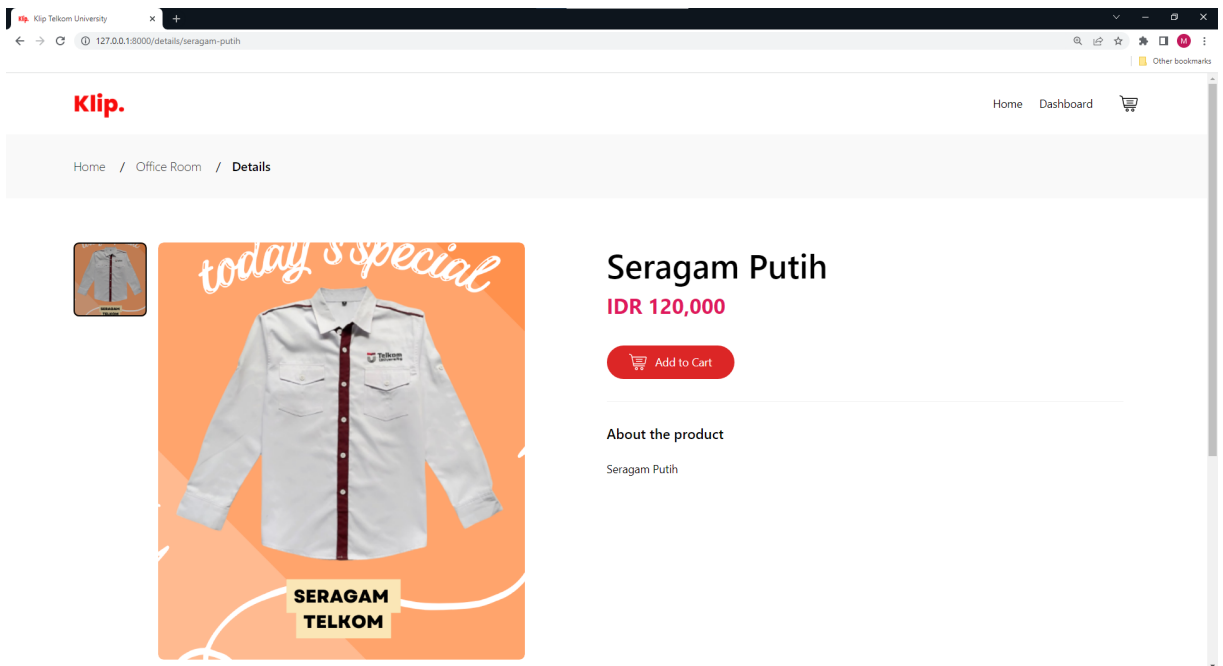
No.	Actor	Action Description
1	Pembeli	Menampilkan detail dari produk
2	System	Menampilkan detail dari produk
3	Pembeli	Dapat melihat detail dari produk

- iii. Post-Condition : Sistem berhasil menampilkan halaman detail produk

3.1.8.2. UI Design dan Deskripsi Objek UI #8 Melihat Detail Produk



Gambar 31 Page Detail Product 1



Gambar 32 Page Detail Product 2

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>details_product name</i>	<i>Page Detail Product</i>	<i>Halaman untuk menampilkan detail suatu produk</i>

Page Detail Product

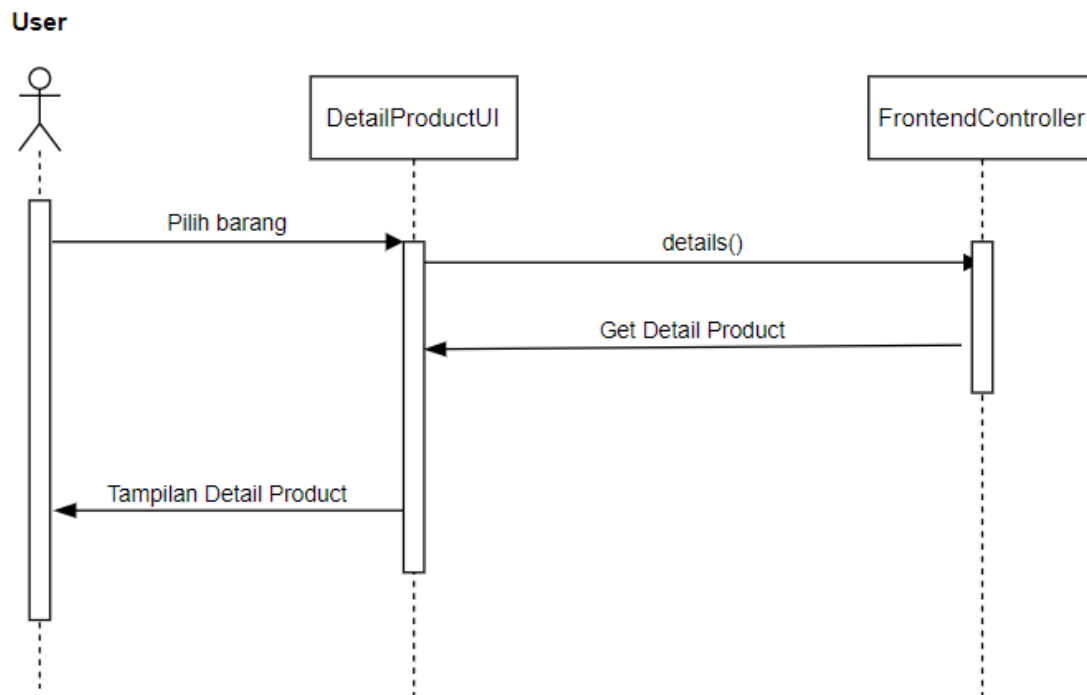
Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_home</i>	<i>Button clickabl e</i>	<i>Home</i>	<i>Jika di klik maka akan melanjutkan ke Home Page Website</i>
<i>btn_add_t o_cart</i>	<i>Button clickabl e</i>	<i>AddToCart</i>	<i>Jika di klik maka akan meredirect ke halaman Cart untuk melanjutkan pembelian produk</i>

3.1.8.3. Identifikasi Object Baru & Tipe Kelas #8 Melihat Detail Produk Penjualan/Pembelian

TABEL OBJECT PERANCANGAN

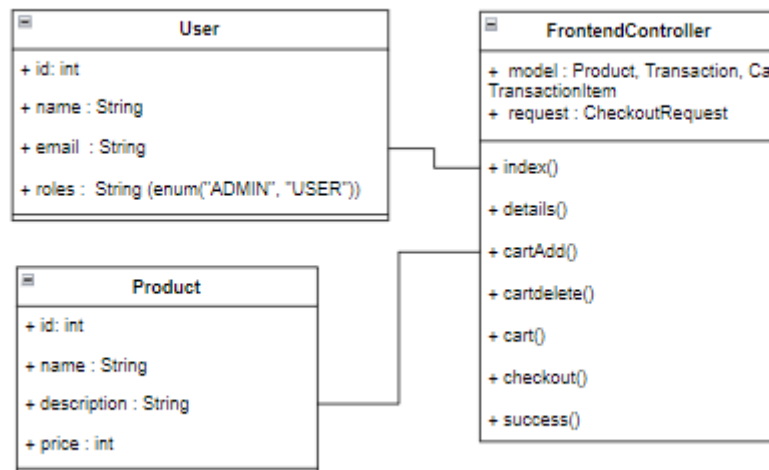
No	Nama Object Baru	Jenis / Tipe Kelas
<i>1</i>	<i>User</i>	<i>Actor</i>
<i>2</i>	<i>DetailProductUI</i>	<i>Boundary</i>
<i>3</i>	<i>FrontendController</i>	<i>Controller</i>

3.1.8.4. Sequence Diagram #8 Melihat Detail Produk



Gambar 33 Sequence Diagram Melihat Detail Produk

3.1.8.5. Class Diagram #8 Melihat Detail Produk



3.1.9. Use Case #9 Mengelola Data User

3.1.9.1. Use Case Scenario #9 Mengelola Data User

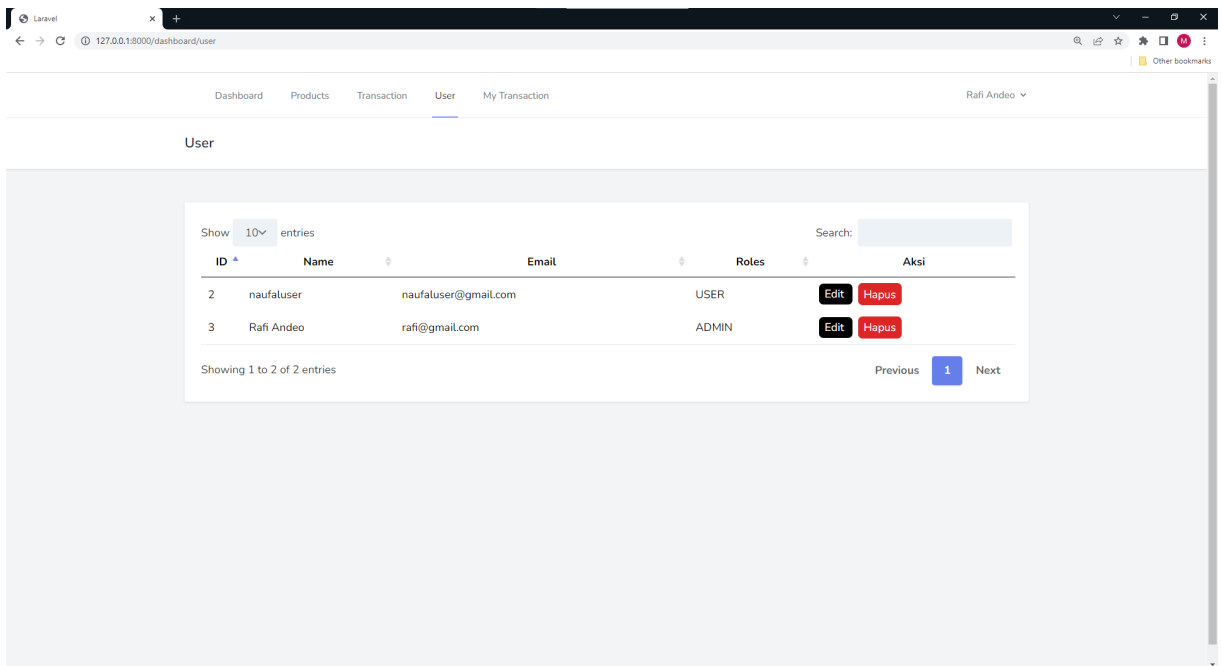
Skenario Use Case #9 :

- i. Pre-Condition: Penjual berada di halaman dashboard User
- ii. Use Case Description
 - a. Primary Flow

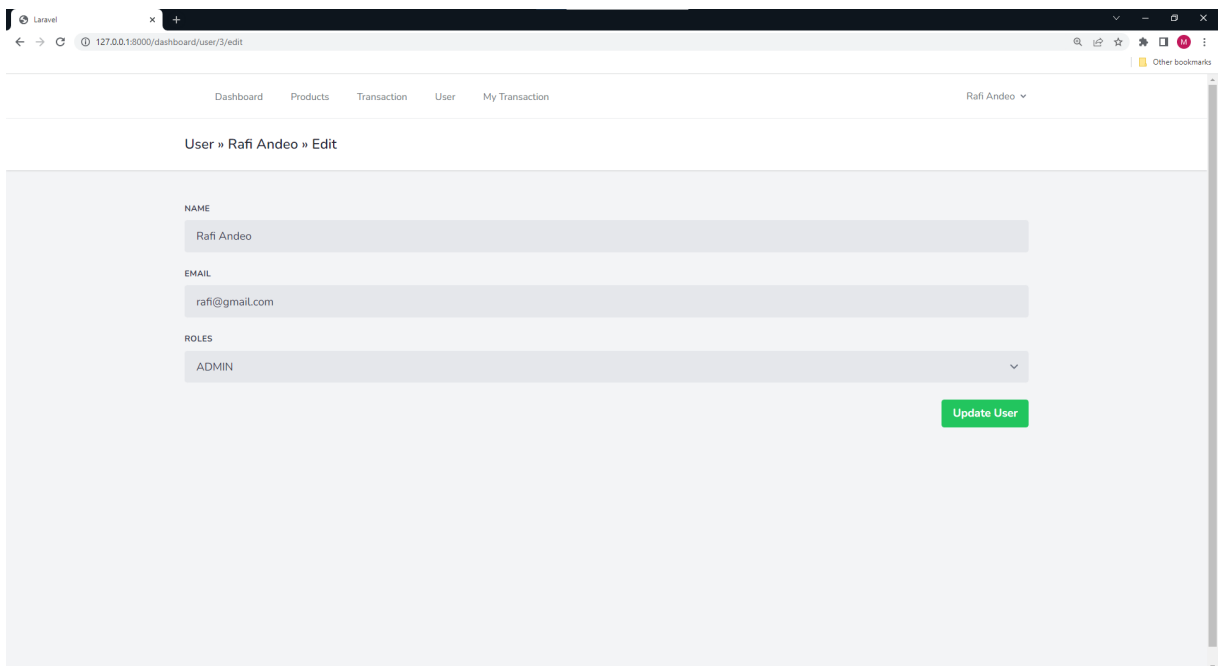
No.	Actor	Action Description
1	Penjual	Menekan “user” di bagian navbar
2	System	Melakukan redirect ke halaman dashboard user & menampilkan data user.
3	Penjual	Mengubah roles user pada tombol option
4	System	Melakukan perubahan roles user dan menampilkan data user yang telah berubah.
5	Penjual	Menekan tombol hapus
6	System	Melakukan destroy akun

- iii. Post-Condition: Penjual berhasil mengubah status User dan berhasil tampil di dashboard

3.1.9.2. UI Design dan Deskripsi Objek UI #9 Mengelola Data User



Gambar 35 Page User



Gambar 36 Page Edit User

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>dashboard user</i>	<i>Page User</i>	<i>Halaman untuk mengelola data user seperti menghapus dan trigger untuk page edit user</i>
<i>dashboard user edit</i>	<i>Page Edit User</i>	<i>Halaman untuk mengedit data user</i>

Page User

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_edit_user</i>	<i>Field Dropdown</i>	<i>Edit Roles</i>	<i>Jika diklik maka akan redirect ke halaman edit product</i>
<i>btn_delete_user</i>	<i>Button clickable</i>	<i>Hapus</i>	<i>Jika diklik, maka akan menghapus data user yang dipilih</i>

Page Edit User

Id_Objek	JENIS	LABEL*	Keterangan**
<i>txt_name</i>	<i>Text Field editable</i>	<i>Name</i>	<i>Text field yang digunakan untuk mengisi nama user</i>
<i>txt_email</i>	<i>Text Field editable</i>	<i>Email</i>	<i>Text field yang digunakan untuk mengisi email user</i>
<i>btn_roles</i>	<i>Button Dropdown clickable</i>	<i>Roles</i>	<i>Jika diklik, maka akan muncul dropdown menu untuk memilih role user</i>
<i>btn_update</i>	<i>Button clickable</i>	<i>Update User</i>	<i>Jika diklik, maka akan mengupdate data user sesuai perubahan yang dilakukan</i>

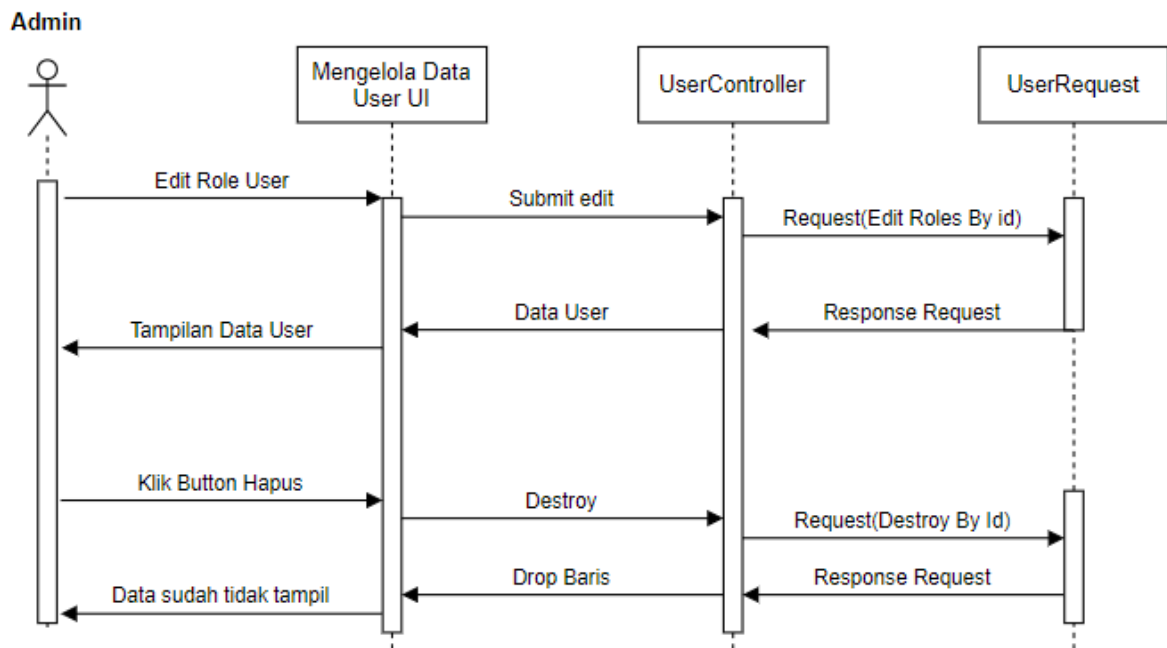
3.1.9.3. Identifikasi Object Baru & Tipe Kelas #9 Mengelola Data User

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	<i>Admin</i>	<i>Actor</i>
2	<i>Mengelola Data User UI</i>	<i>Boundary</i>
3	<i>UserController</i>	<i>Controller</i>
4	<i>UserRequest</i>	<i>Controller</i>

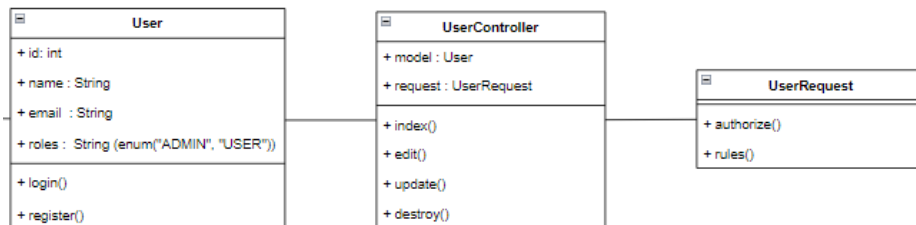
*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.9.4. Sequence Diagram #9 Mengelola Data User



Gambar 37 Sequence Diagram Mengelola Data User

3.1.9.5. Class Diagram #9 Mengelola Data User



Gambar 38 Class Diagram Mengelola User

3.1.10. Use Case #10 Add to Cart

3.1.10.1. Use Case Scenario #10 Add to Cart

Skenario Use Case #10 :

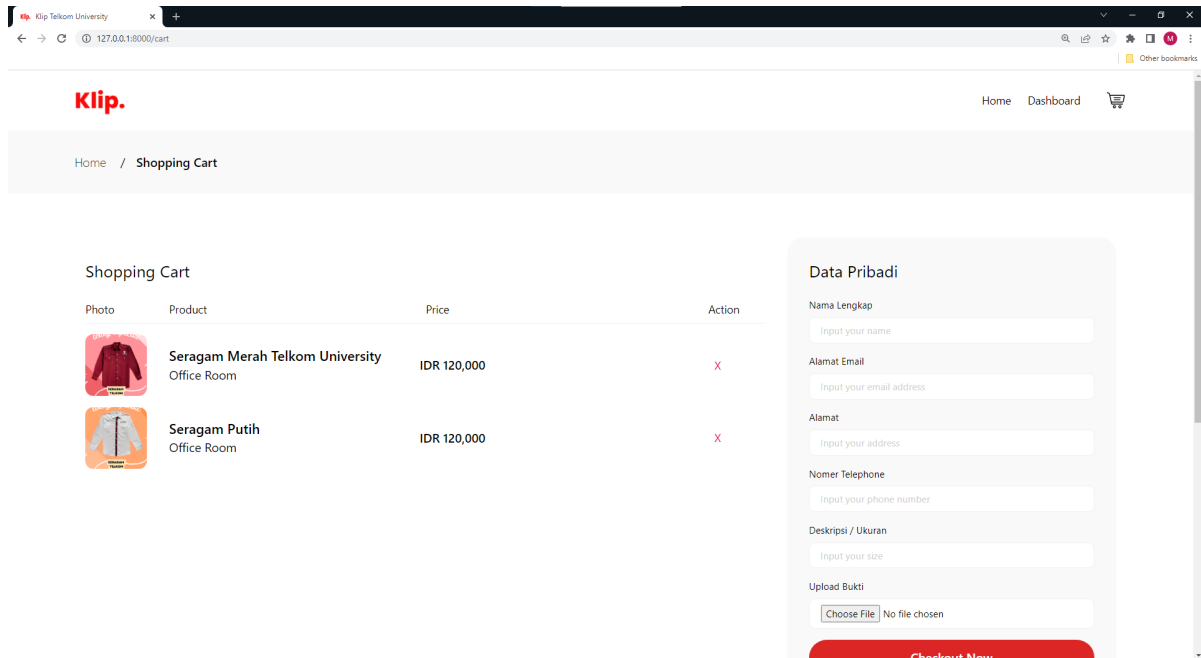
- i. Pre-Condition: Pembeli berada di halaman produk
- ii. Use Case Description
 - a. Primary Flow

No.	Actor	Action Description
1	Pembeli	Menekan tombol tambah ke keranjang

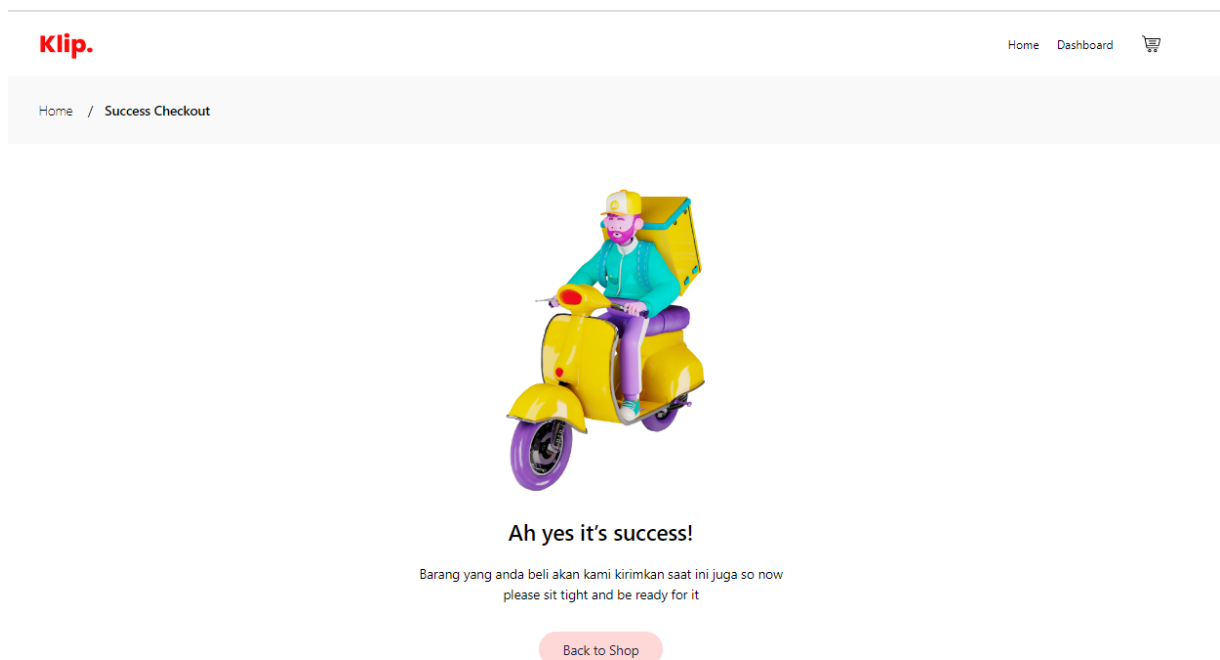
2	System	Memasukkan barang ke dalam keranjang
---	--------	--------------------------------------

iii. Post-Condition: Pembeli berhasil menambahkan barang ke keranjang

3.1.10.2. UI Design dan Deskripsi Objek UI #10 Add to Cart



Gambar 39 Page Shopping Cart



Gambar 40 Page Success

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>cart</i>	<i>Page Shopping Cart</i>	<i>Halaman untuk menampilkan keranjang berisi beberapa produk yang akan dibeli beserta detail pengiriman</i>
<i>success</i>	<i>Page Success</i>	<i>Halaman untuk menampilkan keterangan bahwa pemesanan telah berhasil</i>

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detail

Page Shopping Cart

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_cancel</i>	<i>Button clickable</i>	<i>X</i>	<i>Jika diklik, akan menghapus produk dari keranjang atau cart</i>
<i>txt_name</i>	<i>Text Field editable</i>	<i>Nama Lengkap</i>	<i>Text field yang digunakan untuk mengisi nama lengkap user pembeli</i>
<i>txt_email</i>	<i>Text Field editable</i>	<i>Email</i>	<i>Text field yang digunakan untuk mengisi alamat email</i>
<i>txt_addresses</i>	<i>Text Field editable</i>	<i>Address</i>	<i>Text field yang digunakan untuk mengisi alamat lengkap user pembeli</i>
<i>txt_phone number</i>	<i>Text Field editable</i>	<i>Phone Number</i>	<i>Text field yang digunakan untuk mengisi nomor telepon user pembeli</i>
<i>txt_description</i>	<i>Text Field editable</i>	<i>Description</i>	<i>Text field yang digunakan untuk mengisi deskripsi/ukuran barang</i>
<i>file_upload_bukti</i>	<i>File uploadable</i>	<i>Bukti Transfer</i>	<i>form field tipe file yang digunakan pembeli untuk mengunggah gambar upload bukti</i>
<i>btn_checkout</i>	<i>Button clickable</i>	<i>Checkout now</i>	<i>Jika diklik, maka akan redirect ke halaman page success</i>

Page Success

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_back_to_shop</i>	<i>Button clickable</i>	<i>Back To Shop</i>	<i>Jika diklik, akan merirect ke halaman landing page untuk melihat produk-produk yang dijual</i>

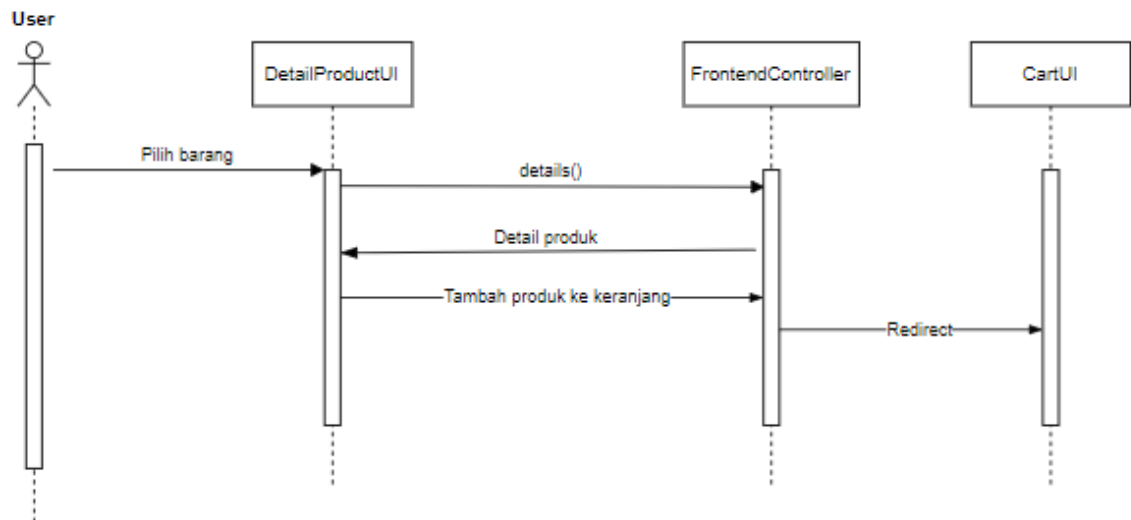
3.1.10.3. Identifikasi Object Baru & Tipe Kelas #10 Add to Cart

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	<i>User</i>	<i>Actor</i>
2	<i>DetailProductUI</i>	<i>Boundary</i>
3	<i>CartUI</i>	<i>Boundary</i>
4	<i>FrontendController</i>	<i>Controller</i>

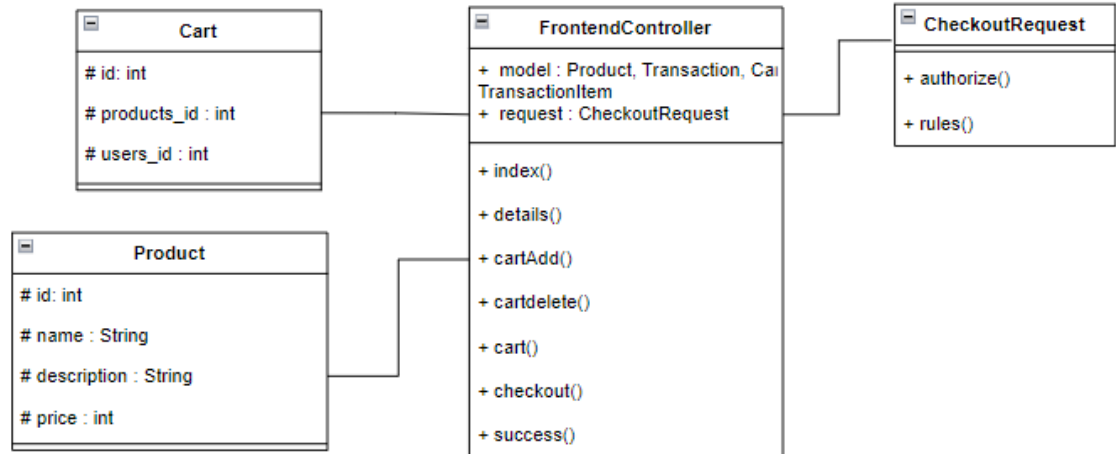
**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

3.1.10.4. Sequence Diagram #10 Add to Cart



Gambar 41 Sequence Diagram Add to Cart

3.1.11.5. Class Diagram #10 Add to Cart



Gambar 42 Class Diagram Add to Cart

3.1.12. Use Case #11 Melihat History

3.1.12.1. Use Case Scenario #11 Melihat History

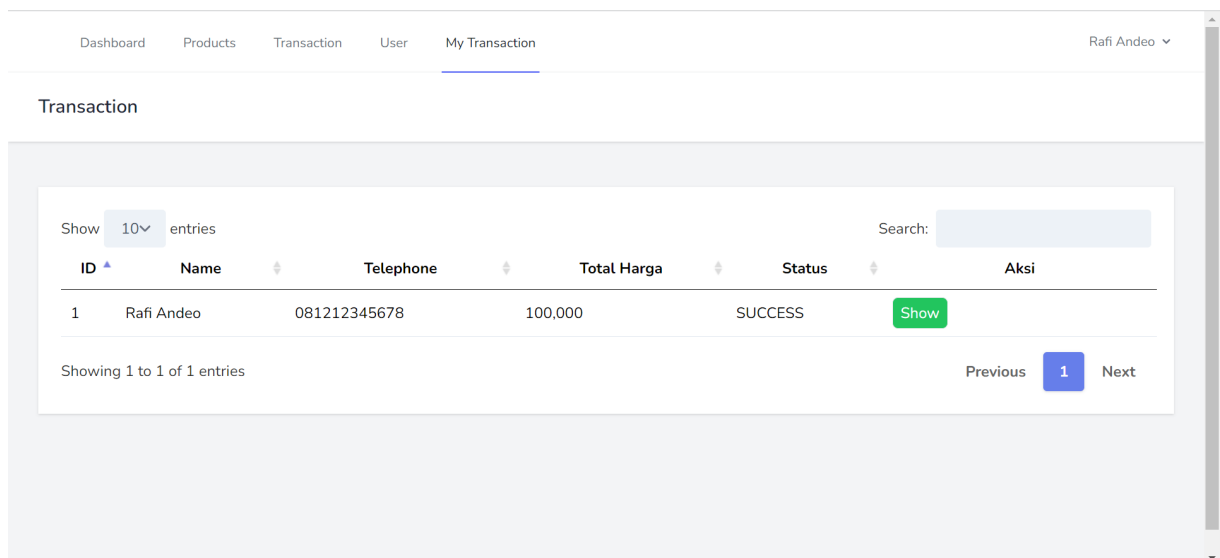
Skenario Use Case #11 :

- i. Pre-Condition: Pembeli berada di halaman produk
- ii. Use Case Description
 - a. Primary Flow

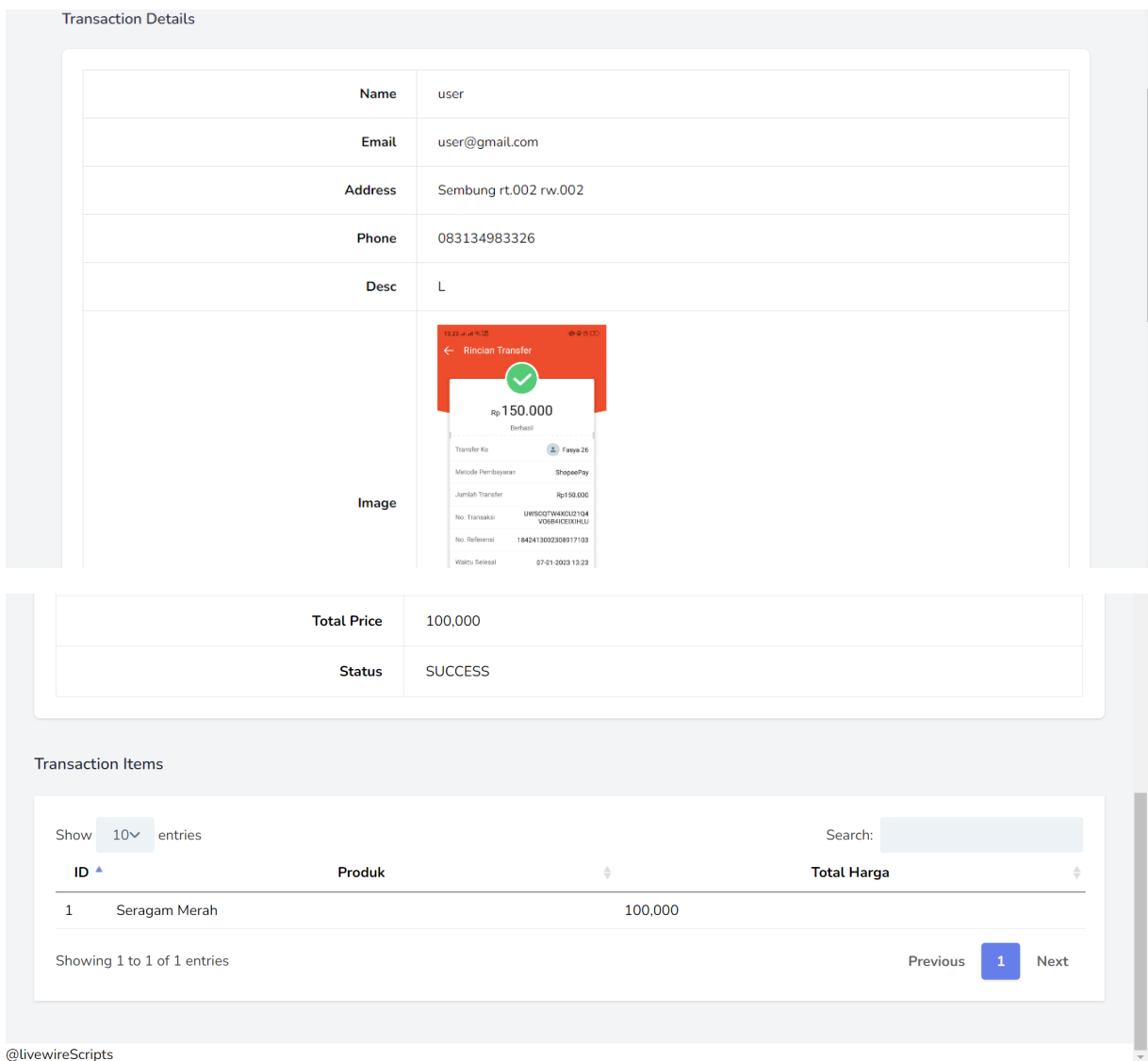
No.	Actor	Action Description
1	Pembeli	Menekan tombol my transaction pada navbar
2	System	Menampilkan halaman history pembelian
3	Pembeli	Menekan Tombol Show
4	System	System akan melakukan redirect pembeli ke detail pembelian

- iii. Post-Condition: Pembeli berhasil melihat daftar history dari pembelian

3.1.12.2. UI Design dan Deskripsi Objek UI #11 Melihat History



Gambar 43 Page My Transaction



Gambar 44 Page Show My Transaction

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
<i>dashboard_my-transaction</i>	<i>Page MyTransaction</i>	<i>Halaman yang menampilkan history atau riwayat transaksi pembelian dari pembeli</i>
<i>dashboard_my-transaction_show</i>	<i>Page Show Transaction</i>	<i>Halaman yang menampilkan detail transaksi pembelian dari pembeli</i>

Page My Transaction

Id_Objek	JENIS	LABEL*	Keterangan**
<i>btn_show</i>	<i>Button Show</i>	<i>Show</i>	<i>Jika di klik, akan menampilkan halaman detail dari pembelian atau transaksi</i>

Page Show My Transaction

Id_Objek	JENIS	LABEL*	Keterangan**
-	-	-	-

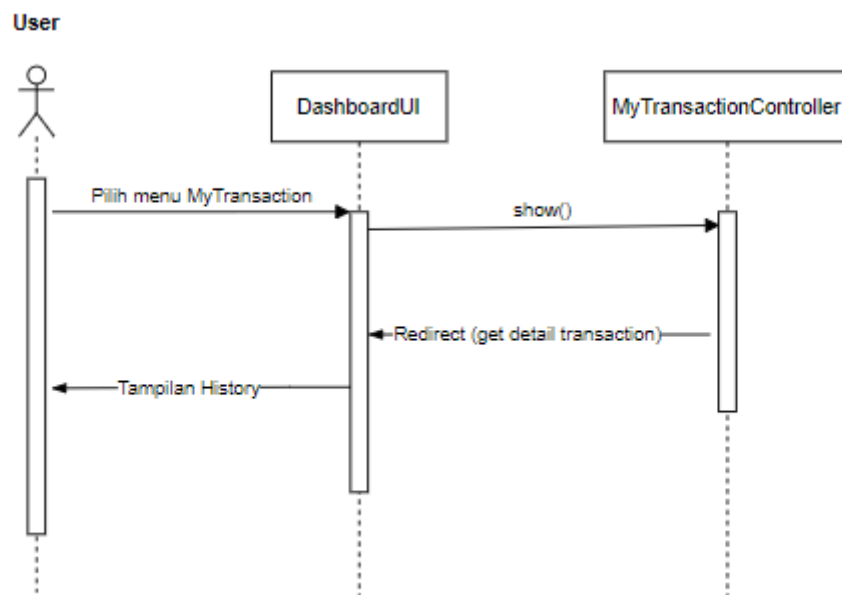
3.1.12.3. Identifikasi Object Baru & Tipe Kelas #11 Melihat History

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	<i>User</i>	<i>Actor</i>
2	<i>DashboardUI</i>	<i>Boundary</i>
3	<i>MyTransactionController</i>	<i>Controller</i>

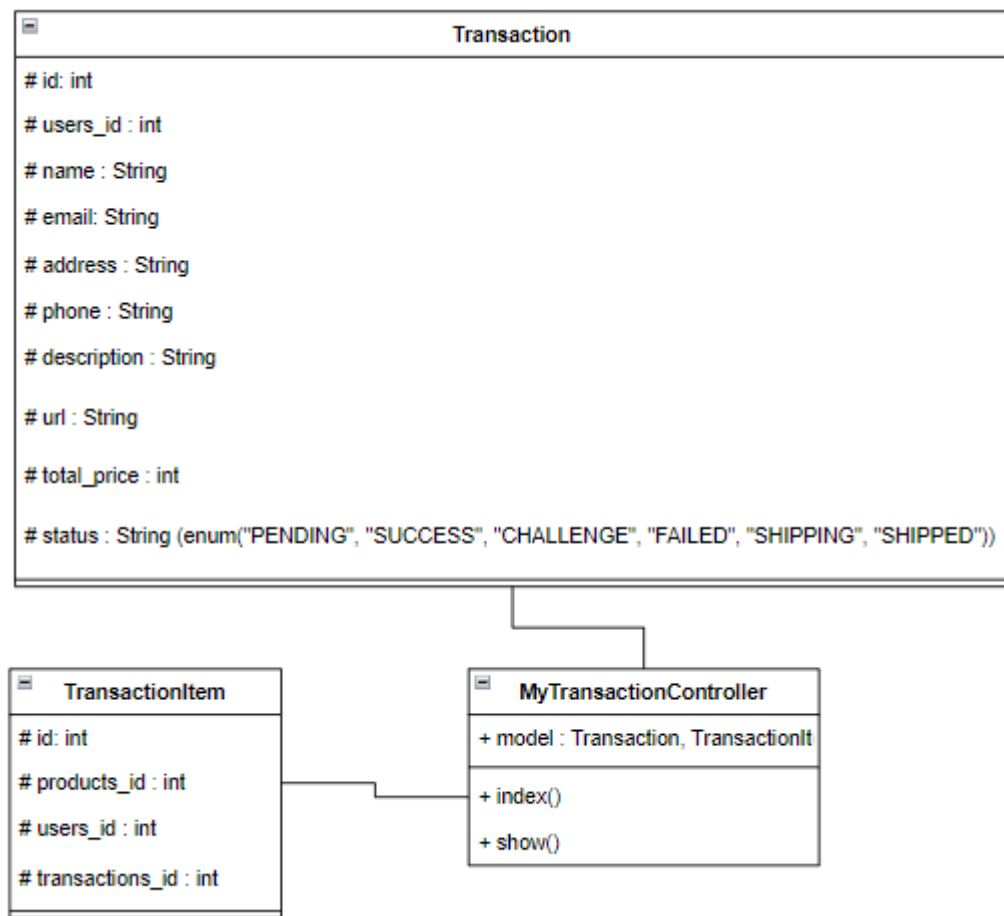
*Tipe kelas seperti *Boundary*(Interface), *Entity*(Database), *Controller*

3.1.12.4. Sequence Diagram #11 Melihat History



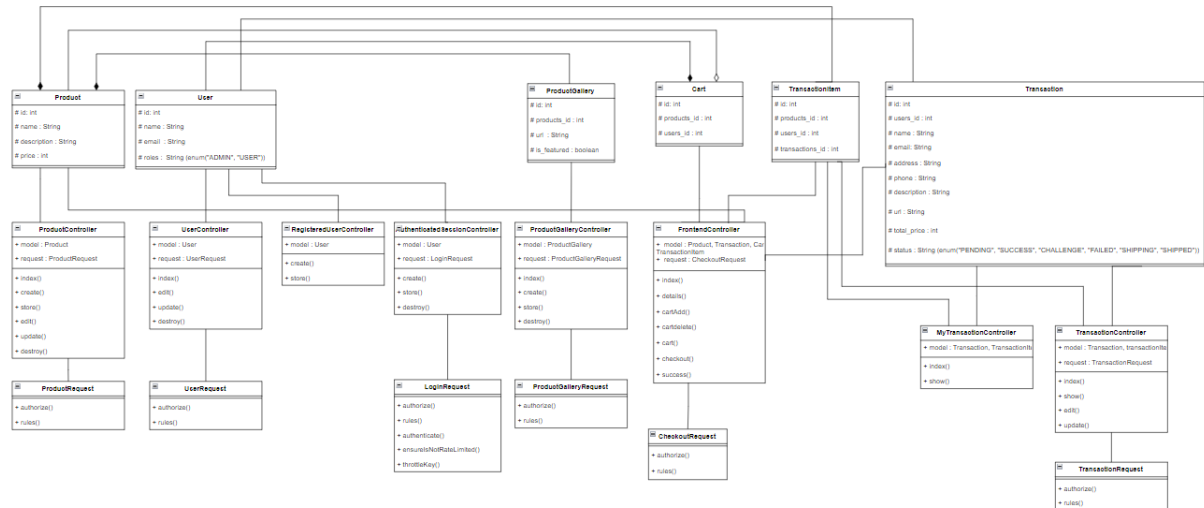
Gambar 45 Sequence Diagram Melihat History

3.1.12.5. Class Diagram #11 Melihat History



Gambar 46 Class Diagram Melihat History

3.2. Diagram Kelas Keseluruhan



Gambar 47 Class Diagram Klip

Link Class Diagram :

https://drive.google.com/file/d/1xTuUL-_yeemts43VwsKfhYCoWRdtthdQ/view?usp=sharing

TABEL KELAS :

ID Kelas	Nama Kelas Perancangan	Atribut (visibility)	Method / Operation
<i>user</i>	<i>User</i>	<ul style="list-style-type: none"> id : int name : String email : String roles : String (enum("ADMIN", "USER")) 	-
<i>user_controller</i>	<i>UserController</i>	<ul style="list-style-type: none"> model : User request : UserRequest 	<ul style="list-style-type: none"> index() edit() update() destroy()
<i>user_request</i>	<i>UserRequest</i>	-	<ul style="list-style-type: none"> authorize() rules()
<i>product</i>	<i>Product</i>	<ul style="list-style-type: none"> id : int name : String description : String price : int 	-
<i>product_controller</i>	<i>ProductController</i>	<ul style="list-style-type: none"> model : Product request : ProductRequest 	<ul style="list-style-type: none"> index() create() store() edit() update() destroy()

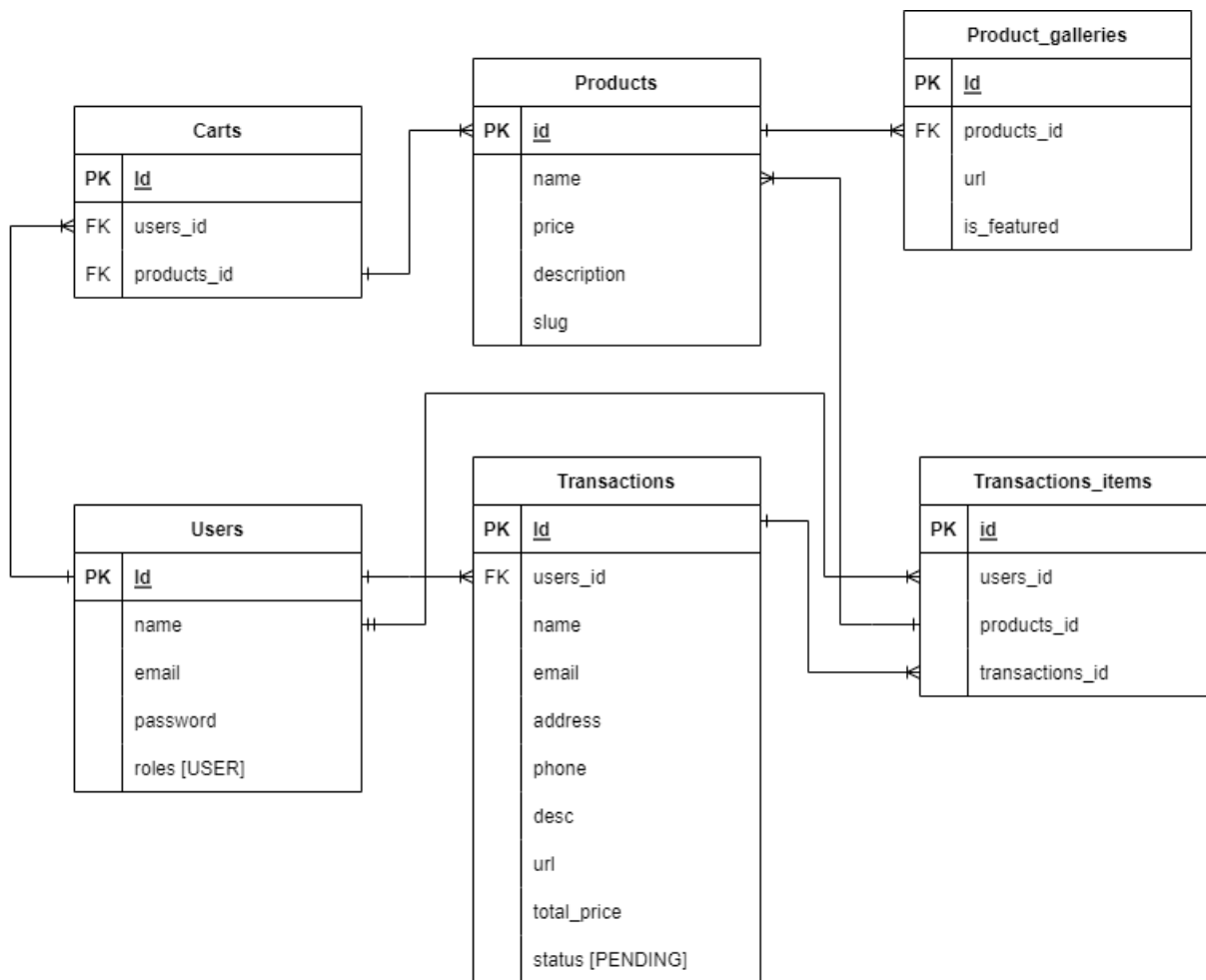
<i>product_request</i>	<i>ProductRequest</i>	-	<ul style="list-style-type: none"> • authorize() • rules()
<i>product_gallery</i>	<i>ProductGallery</i>	<ul style="list-style-type: none"> • id : int • products_id : int • url : String • is_featured : boolean 	-
<i>product_gallery_controller</i>	<i>ProductGalleryController</i>	<ul style="list-style-type: none"> • model : ProductGallery • request : ProductGalleryRequest 	<ul style="list-style-type: none"> • index() • edit() • update() • destroy()
<i>product_gallery_request</i>	<i>ProductGalleryRequest</i>	-	<ul style="list-style-type: none"> • authorize() • rules()
<i>cart</i>	<i>Cart</i>	<ul style="list-style-type: none"> • id : int • products_id : int • users_id : int 	-
<i>transaction_item</i>	<i>TransactionItem</i>	<ul style="list-style-type: none"> • id : int • products_id : int • users_id : int • transactions_id : int 	-
<i>frontend_controller</i>	<i>FrontendController</i>	<ul style="list-style-type: none"> • model : Product, Transaction, Cart • request : CheckoutRequest 	<ul style="list-style-type: none"> • index() • details() • cartAdd() • cartdelete() • cartIndex() • checkout() • success()
<i>checkout_controller</i>	<i>CheckoutRequest</i>	-	<ul style="list-style-type: none"> • authorize() • rules()

<i>transaction</i>	<i>Transaction</i>	<ul style="list-style-type: none"> • Id : int • users_id : int • name : String • email : String • address : String • phone : String • courier : String • payment : String • payment_url : String • total_price : int • status : String (enum("PENDING", "SUCCESS", "CHALLENGE", "FAILED", "SHIPPING", "SHIPPED")) 	-
<i>my_transaction_controller</i>	<i>MyTransactionController</i>	<ul style="list-style-type: none"> • model: Transaction 	<ul style="list-style-type: none"> • index() • show()
<i>transaction_controller</i>	<i>TransactionController</i>	<ul style="list-style-type: none"> • model: Transaction • request: TransactionRequest 	<ul style="list-style-type: none"> • index() • edit() • update() • destroy()
<i>transaction_request</i>	<i>TransactionRequest</i>	-	<ul style="list-style-type: none"> • authorize() • rules()
<i>registered_user_controller</i>	<i>RegisteredUserController</i>	<ul style="list-style-type: none"> • model: User 	<ul style="list-style-type: none"> • create() • store()
<i>authenticated_session_controller</i>	<i>AuthenticatedSessionController</i>	<ul style="list-style-type: none"> • model: User • request: LoginRequest 	<ul style="list-style-type: none"> • create() • store() • destroy()
<i>login_request</i>	<i>LoginRequest</i>	-	<ul style="list-style-type: none"> • authorize() • rules() • authenticate() • ensureIsNotRateLimited() • throttleKey()

Untuk setiap kelas:

- identifikasi operasi (mengacu pada tanggung-jawab kelas), identifikasi atribut, termasuk visibility-nya

3.3. Perancangan Data / Basis Data



Gambar 48 Entity Relationship Diagram Klip

3.4. Perancangan Algoritma dan/atau Query

Algoritma #1

Contoh:

Nama Kelas : User

Nama Operasi : Register

Algoritma :

```

name = str(input("Name: "))
email = str(input("Email: "))
password = str(input("Password: "))
confirm_password = str(input("Confirm Password"))

new_user, user_data = User
  
```

```

new_user = User.find("SELECT * FROM User WHERE "name" = name AND
"email" = email AND "password" = password AND "password" =
confirm_password")

if new_user != null:
    return error

user_data = User.create("INSERT INTO User VALUES (name, email,
password)").save()

return user_data

```

Algoritma #2

Contoh:

Nama Kelas : Product

Nama Operasi : Membuat Product

Algoritma :

```

name = str(input("Product Name: "))
description = str(input("Description: "))
price = float(input("Product Price: "))

new_product = Product

new_product = Product.create("INSERT INTO Product VALUES (name, description,
price)").save()

return new_product

```

Algoritma #3

Contoh:

Nama Kelas : Product

Nama Operasi : Menghapus Product

Algoritma :

```

name = str(input("Product Name: "))
description = str(input("Description: "))
price = float(input("Product Price: "))

new_product = Product

new_product = Product.find("SELECT * FROM Product WHERE "name" = name
AND "description" = description AND "price" = price")

if new_product != null:
    new_product = Product.destroy("DELETE FROM Product WHERE "name" =
name AND "description" = description AND "price" = price")

elif new_product == null:
    return error

```

Algoritma #4

Contoh:

Nama Kelas : Product

Nama Operasi : Edit Product

Algoritma :

```
name = str(input("Product Name: "))
description = str(input("Description: "))
price = float(input("Product Price: "))

new_product = Product

new_product = Product.find("SELECT * FROM Product WHERE "name" = name
AND "description" = description AND "price" = price")

if new_product != null:
    name = str(input("Product Name: "))
    description = str(input("Description: "))
    price = float(input("Product Price: "))
    new_product = Product.edit("UPDATE Product SET "name" = name,
"description" = description, "price" = price WHERE "name" = name AND
"description" = description AND "price" = price")

elif new_product == null:
    return error
```

Algoritma #5

Contoh:

Nama Kelas : User

Nama Operasi : Login

Algoritma :

```
email = str(input("Email: "))
password = str(input("Password: "))

new_user = User

new_user = User.find("SELECT * FROM User WHERE "email" = email AND
"password" = password")

return new_user
```

Algoritma #6

Contoh:

Nama Kelas : Product Gallery

Nama Operasi : Membuat Product Gallery

Algoritma :

```

new_product = Product

url = str(input("Choose Files: "))
product_id = str(input())

product_id = Product.find("SELECT * FROM Product WHERE 'id Product' =
product_id")

new_product_picture = Product Gallery

if product_id == null:
    return error

new_product_picture = Product Gallery.create("INSERT INTO Product Gallery
VALUES (product_id, url)").save()

return new_product

```

Algoritma #7

Contoh:

Nama Kelas : Product Gallery

Nama Operasi : Menghapus Product Gallery

Algoritma :

```

url = str(input("Choose Files: "))

new_product_picture = Product Gallery

new_product_picture = Product Gallery.find("SELECT * FROM Product Gallery
WHERE 'url' = url")

if new_product_picture != null:
    new_product_picture = Product Gallery.destroy("DELETE FROM Product
Gallery WHERE 'url' = url")
elif new_product_picture == null:
    return error

```

Algoritma #8

Contoh:

Nama Kelas : User

Nama Operasi : Delete User

Algoritma :

```

name = str(input("Name: "))
email = str(input("Email: "))
roles = str(input("Roles: "))

new_user = User

new_user = User.find("SELECT * FROM User WHERE 'name' = name AND
'email' = email AND 'roles' = roles")

```

```

if new_user != null:
    name = str(input("Name: "))
    email = str(input("Email: "))
    roles = str(input("Roles: "))
    new_user = User.destroy("DELETE FROM User WHERE "name" = name,
"email" = email, "roles" = roles")

elif new_user == null:
    return error

```

Algoritma #9

Contoh:

Nama Kelas : User

Nama Operasi : Edit User

Algoritma :

```

name = str(input("Name: "))
email = str(input("Email: "))
roles = str(input("Roles: "))

new_user = User

new_user = User.find("SELECT * FROM User WHERE "name" = name AND
"email" = email AND "roles" = roles")

if new_user != null:
    name = str(input("Name: "))
    email = str(input("Email: "))
    roles = str(input("Roles: "))
    new_user = User.edit("UPDATE User SET "name" = name, "email" = email,
"roles" = roles WHERE "name" = name AND "email" = email AND "roles" =
roles")

elif new_user == null:
    return error

```

Algoritma #10

Contoh:

Nama Kelas : Transaction

Nama Operasi : Edit Transaction

Algoritma :

```

new_user = User
new_product = Product

status = str(input())
user_id = str(input())
product_id = str(input())

new_transaction = Transaction

```



```

user_id = User.find("SELECT * FROM User WHERE "id User" = user_id")

product_id = Product.find("SELECT * FROM Product WHERE "id Product" =
product_id")

if new_user == null or new_product == null:
    return error

status = str(input())
new_transaction = Transaction.edit("UPDATE Transaction SET "status" = status
WHERE "status" = status")

```

Algoritma #11

Contoh:

Nama Kelas : Transaction

Nama Operasi : Menghapus Transaction

Algoritma :

```

new_user = User
new_product = Product

status = str(input())
user_id = str(input())
product_id = str(input())

new_transaction = Transaction

user_id = User.find("SELECT * FROM User WHERE "id User" = user_id")

product_id = Product.find("SELECT * FROM Product WHERE "id Product" =
product_id")

if new_user == null or new_product == null:
    return error

status = str(input())
new_transaction = Transaction.destroy("DELETE FROM Transaction WHERE
"status" = status")

```

Algoritma #12

Contoh:

Nama Kelas : Product

Nama Operasi : Membeli Product

Algoritma :

```

new_product_picture = Product Gallery
new_transaction = Transaction

url = str(input())
name = str(input("Product Name: "))

```

```
add = str(input("Add to Cart: "))

new_product = Product

url = Product Gallery.find("SELECT * FROM Product Gallery WHERE "url" = url")

new_product = Product.find("SELECT * FROM Product WHERE "name" = name")

new_transaction = Transaction.update("(UPDATE Transaction SET "status" = "PENDING"))
```

Query #1

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Users

SELECT * FROM Users

Keterangan : Query untuk menyimpan data Users

Query #2

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Transaction_items

SELECT * FROM Transaction_items

Keterangan : Query untuk menyimpan data produk yang masuk transaksi user

Query #3

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Transactions

SELECT * FROM Transactions

Keterangan : Query untuk menyimpan data transaksi user berupa data pribadi user

Query #4

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Product_galleries

SELECT * FROM Product_galleries

Keterangan : Query untuk menyimpan data gambar produk

Query #5

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Products

SELECT * FROM Products

Keterangan : Query untuk menyimpan data produk

Query #6

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Carts

SELECT * FROM Carts

Keterangan : Query untuk menyimpan seluruh data produk yang masuk transaksi user

4. Matriks Keruntutan (Requirement Traceability Matrix)

Kode FR	Nama Functional Requirement	Nama Use Case
(SKPL-KLIP-001)	Pembeli dan penjual dapat melakukan login dan registrasi	1. Register 2. Login
(SKPL-KLIP-002)	Pembeli dapat melihat produk dari penjual	1. Halaman Utama Penjualan
(SKPL-KLIP-003)	Pembeli dapat melihat detail produk dari penjual	1. Halaman Utama Penjualan 2. Melihat Detail Produk
(SKPL-KLIP-004)	Pembeli dapat menambahkan barang ke keranjang	1. Login 2. Halaman Utama Penjualan 3. Melihat Detail Produk 4. Add to Cart
(SKPL-KLIP-005)	Pembeli dapat membayar produk yang akan dibeli	1. Login 2. Membeli Barang
(SKPL-KLIP-006)	Penjual dapat mengelola data produk	1. Login 2. Mengelola Data Produk
(SKPL-KLIP-007)	Penjual dapat mengelola data transaksi	1. Login 2. Mengelola Data Transaksi
(SKPL-KLIP-008)	Penjual dapat mengelola data user	1. Login 2. Mengelola Data User
(SKPL-KLIP-009)	Pembeli dapat melihat history pembelian	1. Login 2. Melihat History