

## BAB III

### ANALISA DAN PERANCANGAN

Penelitian ini menggunakan metode pengembangan *Rational Unified Process* (RUP), yang melibatkan empat tahap utama: Tahap *Inception*, Tahap *Elaboration*, Tahap *Collaboration*, dan Tahap *Transition*.

#### 3.1 Tahap Inception

Tahap *inception* adalah inialisasi pendefinisian sistem yang akan dibuat seperti analisa masalah dan solusi dari permasalahan tersebut.

##### 3.1.1 Identifikasi Masalah

Selama ini FIDI Laundry masih menggunakan metode konvensional sehingga memungkinkan terjadi human error. Berangkat dari permasalahan tersebut, berikut beberapa identifikasi masalah yang ditemukan :

- a. Pencatatan setiap transaksi yang terjadi hanya di tulis pada nota yang dapat hilang atau terselip.
- b. Pencarian data pelanggan atau transaksi yang masih manual pada buku rekap membutuhkan waktu lama dengan data yang banyak.
- c. Perhitungan pemasukkan setiap bulan yang masih dihitung secara manual menggunakan kalkulator memungkinkan terjadi kesalahan perhitungan jika jumlah datanya banyak dan membutuhkan waktu lama.

- d. Pemesanan jasa laundry hanya melalui aplikasi chatting sehingga pelanggan kurang mendapat informasi penjemputan sudah sampai mana.
- e. Kurangnya informasi terupdate progress cucian pelanggan yang menyebabkan pelayanan pada pelanggan kurang maksimal.
- f. Pembayaran dapat dilakukan baik secara tunai maupun untuk memudahkan pelanggan.

### 3.1.2 Pemecahan Masalah

Berdasarkan identifikasi masalah diatas, ditemukan solusi pemecahan masalah dengan membuat aplikasi mobile berbasis android menggunakan Rational Unified Process untuk proses pengembangannya.

Tabel 3.1 Pemecahan Masalah

Permasalahan	Solusi
Pencatatan setiap transaksi yang terjadi hanya di tulis pada nota yang dapat hilang atau terselip.	Fitur pemesanan secara online dari pelanggan dan scan QR code untuk mencatat transaksi pelanggan saat datang langsung.
Pencarian data pelanggan atau transaksi yang masih manual pada buku rekap membutuhkan waktu lama dengan data yang banyak.	Fitur report transaksi yang dapat di filter berdasarkan nama, bulan, dan tahun sehingga muncul transaksi yang terjadi secara online maupun offline.
Perhitungan pemasukkan setiap bulan yang masih dihitung secara manual menggunakan kalkulator memungkinkan terjadi kesalahan perhitungan jika jumlah datanya	Fitur report transaksi yang terdapat perhitungan jumlah total pembayaran dari data yang di cari.

banyak dan membutuhkan waktu lama.	
Pemesanan jasa laundry hanya melalui aplikasi chatting sehingga pelanggan kurang mendapat informasi penjemputan sudah sampai mana.	Fitur pemesanan secara online yang terdapat peta dan marker dari petugas penjemputan yang dapat dipantau dari aplikasi pelanggan.
Kurangnya informasi terupdate progress cucian pelanggan yang menyebabkan pelayanan pada pelanggan kurang maksimal.	Fitur detail order yang terdapat informasi mengenai progress cucian dan fitur komplain setelah pemesanan selesai.
Pembayaran secara tunai dan tidak tercatat apakah sudah lunas atau belum	Fitur pembayaran secara non tunai dengan payment gateway atau tunai langsung ke tempat laundry.

### 3.2 Tahap Elaboration (Perancangan)

Pada tahap ini merupakan tahapan pembuatan rancangan sistem dari hasil analisa yang dilakukan pada tahap *inception*. Proses yang dilakukan terdiri dari identifikasi aktor, pembuatan use case diagram, dan pemodelan UML seperti activity dan sequence diagram.

#### 3.2.1 Iterasi Pertama

Pada iterasi pertama, peneliti membuat rancangan analisis dan desain sistem dari tahap sebelumnya. Berikut rancangan dari tahap elaboration iterasi pertama :

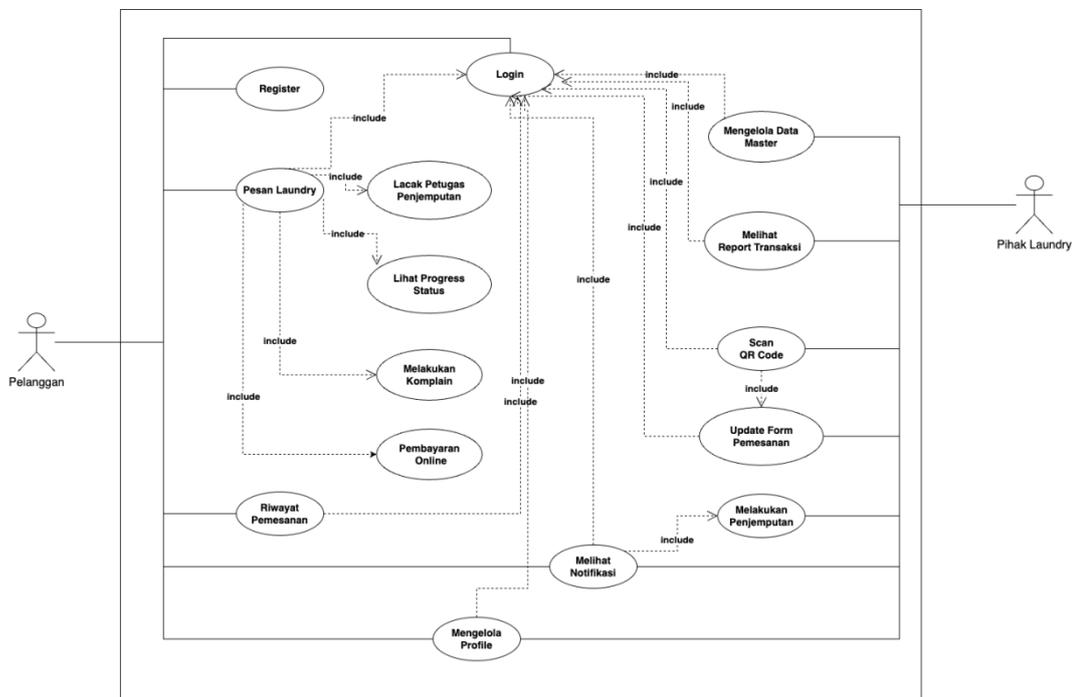
### 3.2.1.1 Use Case Diagram

Tujuan dari use case diagram adalah untuk menunjukkan interaksi antara aktor dan sistem. Pada tabel dibawah ini, menunjukkan identifikasi aktor beserta deskripsinya.

Tabel 3.2 Identifikasi Aktor

Nama Aktor	Deskripsi
Pihak Laundry	Merupakan pemilik laundry yang mempunyai hak akses ke semua menu.
Pelanggan	Merupakan pelanggan laundry yang dapat memesan jasa melalui aplikasi.

Interaksi antara aktor dan sistem digambarkan dalam use case diagram, seperti yang ditunjukkan pada Gambar 3.1:



Gambar 3.1 Use Case Diagram Aplikasi Android FIDI Laundry

Dari gambar diatas kebutuhan tiap aktor dapat dijabarkan sebagai berikut :

Tabel 3.3 Penjelasan *Use Case Diagram*

Aktor	Fitur	Deskripsi
Pihak Laundry	Dapat mengelola data master	Menambah, mengubah, atau menghapus data master.
	Dapat men-scan QR Code	Meng-handle pelanggan yang datang langsung dan sudah mempunyai akun untuk di proses pesannya.
	Dapat melakukan antar-jemput pesanan	Meng-handle pelanggan yang memesan jasa melalui aplikasi.
	Dapat mengelola profile	Mengelola akun sendiri seperti ubah profile dan ubah password.
	Dapat melihat report transaksi	Melihat report transaksi berdasarkan bulan atau tahun.
Pelanggan	Dapat melakukan register dan login	Untuk menggunakan aplikasi pelanggan harus mempunyai akun terlebih dahulu.
	Dapat memesan jasa laundry	Memesan jasa laundry secara online, melacak petugas penjemputan, melihat progress pencucian, serta memberikan feedback kepada pihak laundry.
	Dapat membayar laundry secara online	Membayar laundry secara online (payment gateway) dan offline (cash).
	Dapat melihat riwayat transaksi	Melihat transaksi yang aktif maupun yang telah selesai dikerjakan.
	Dapat mengelola profile	Mengelola akun sendiri seperti ubah profile dan ubah password.

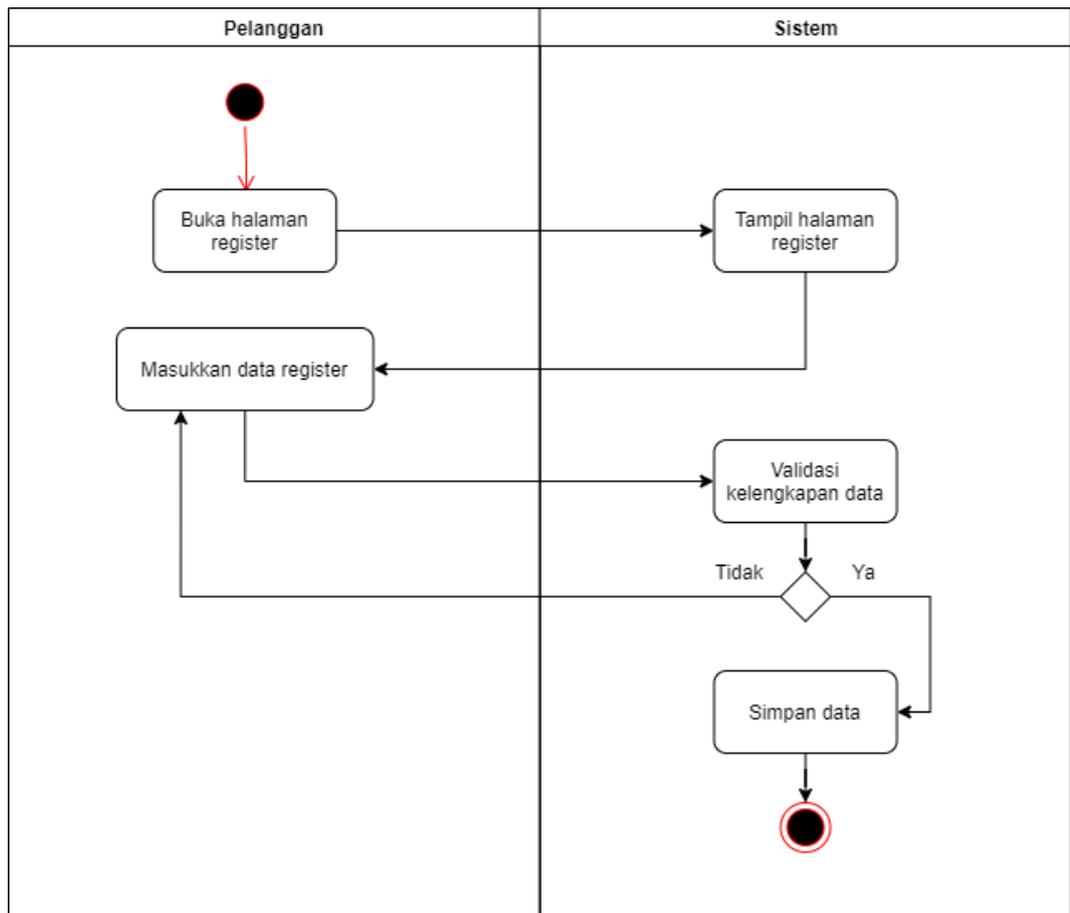
### 3.2.1.2 Activity Diagram

Rancangan sistem pada metode RUP (Tahap *Elaboration*) dituangkan ke dalam melalui *activity* diagram. *Activity* diagram adalah jenis diagram untuk

memodelkan proses-proses yang berlangsung dalam suatu sistem. Berikut beberapa *activity* diagram yang menjelaskan setiap prosesnya.

### 1. *Activity Diagram Register*

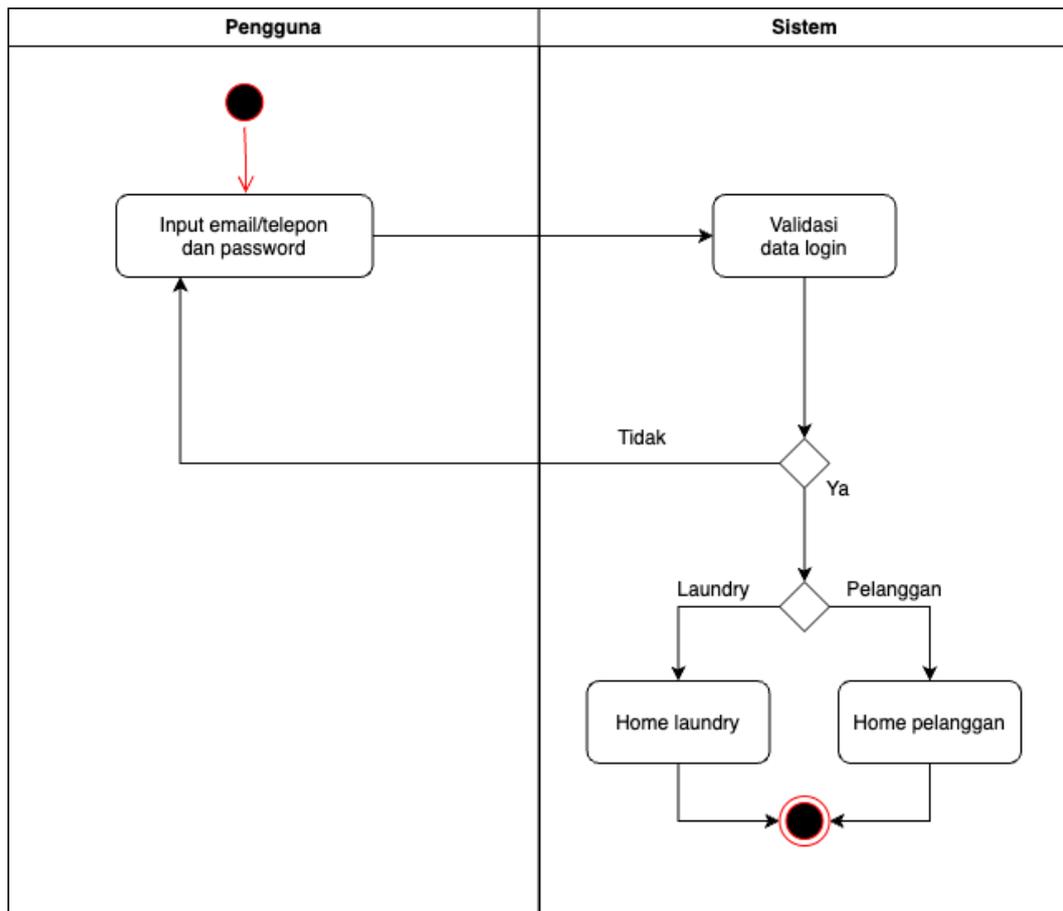
Pada gambar 3.2, dijelaskan bahwa setiap pengguna harus mempunyai akun terlebih dahulu dengan *register* akun yang akan digunakan.



Gambar 3.2 *Activity Diagram Register*

### 2. *Activity Diagram Login*

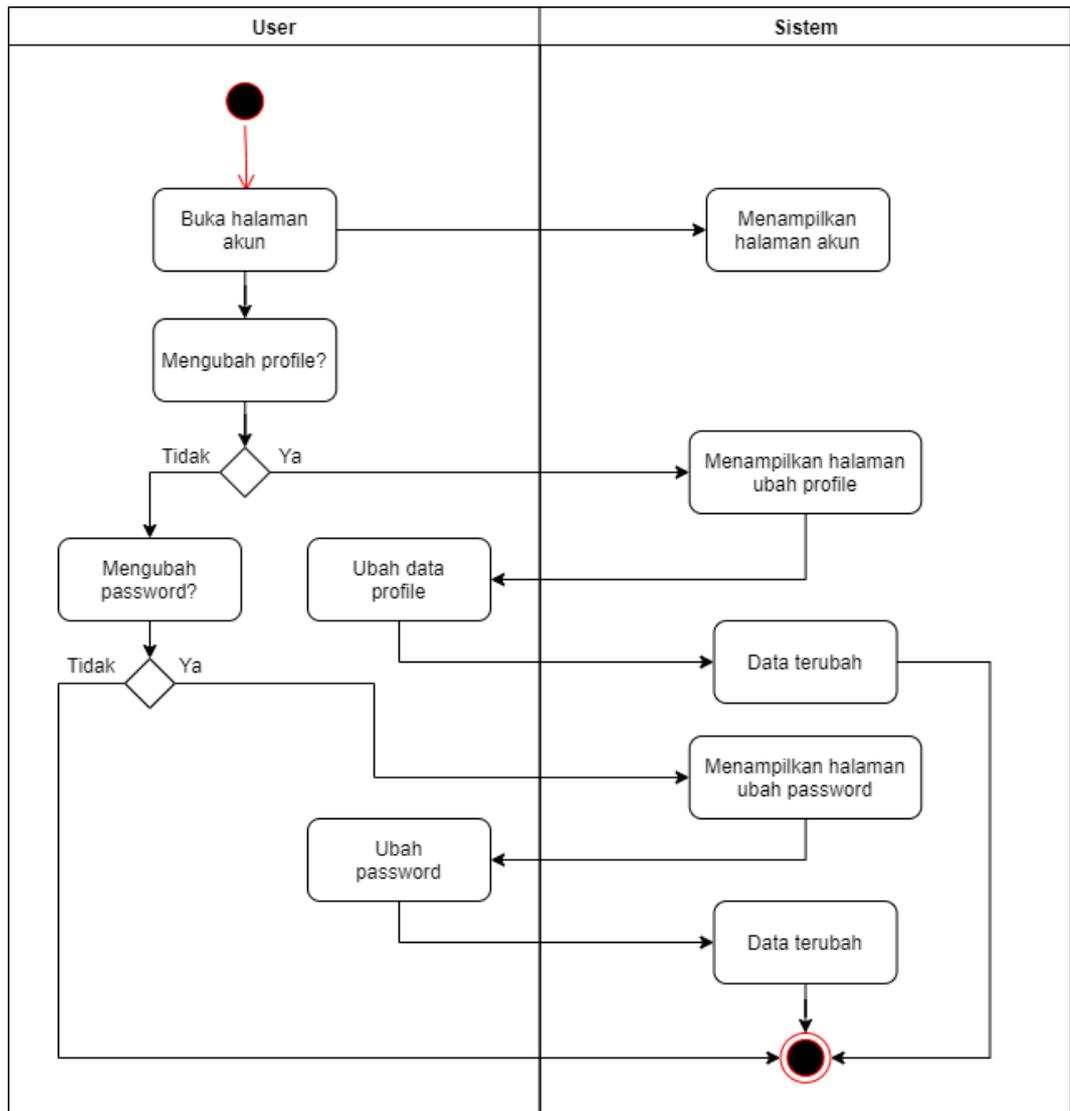
Setelah akun terdaftar, pengguna dapat masuk ke aplikasi melalui halaman *login*. Saat *login*, data yang di masukkan akan di cek apakah akun tersebut sebagai pengguna atau laundry serta akan arahkan ke halaman home masing-masing.



Gambar 3.3 Activity Diagram Login

### 3. Activity Diagram Profile

Pada gambar 3.4, dijelaskan bahwa pengguna dapat mengubah *profile* dan *password* pada halaman *Profile*.



Gambar 3.4 Activity Diagram Profile

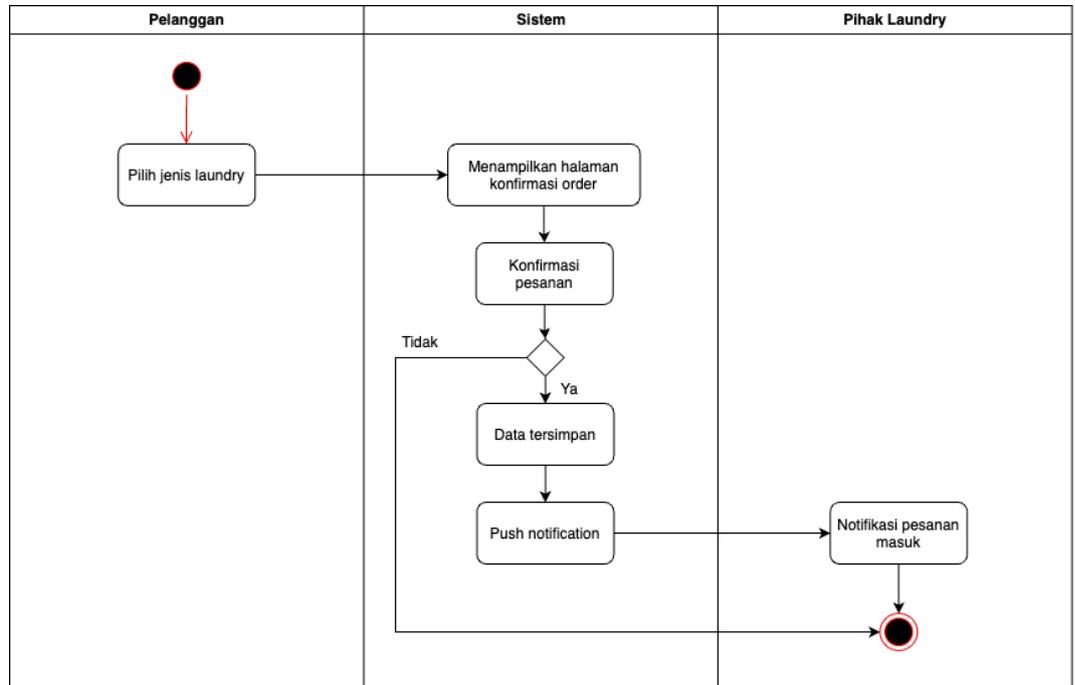
#### 4. Activity Diagram Pelanggan

Fitur yang dapat diakses oleh pelanggan adalah sebagai berikut :

##### a. Pemesanan

Pada gambar 3.5, menjelaskan alur pemesanan melalui aplikasi, pelanggan memilih jenis laundry yang akan dipesan. Lalu, sistem akan menampilkan halaman konfirmasi pesanan. Jika pelanggan melanjutkan

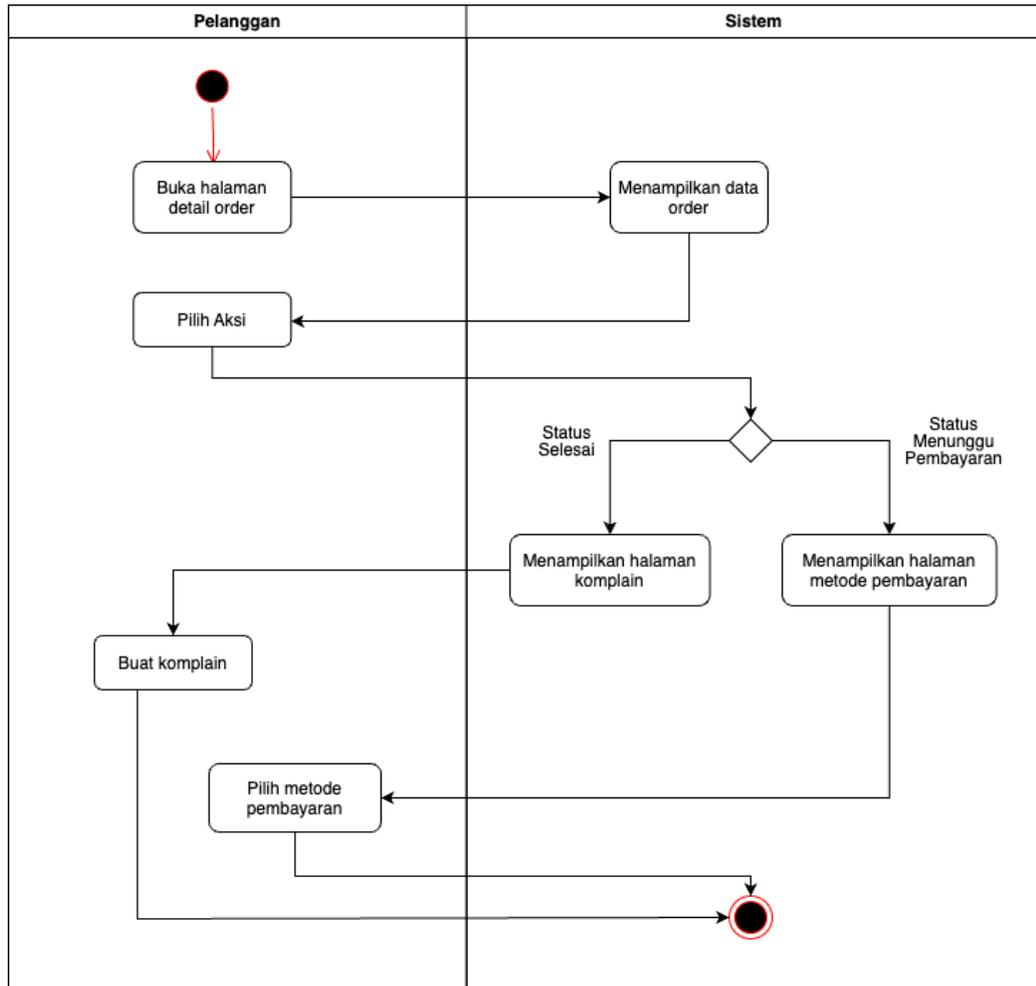
pesanan maka sistem akan mengirimkan *push notification* kepada pihak laundry.



Gambar 3.5 Activity Diagram Pemesanan

b. Detail Pemesanan

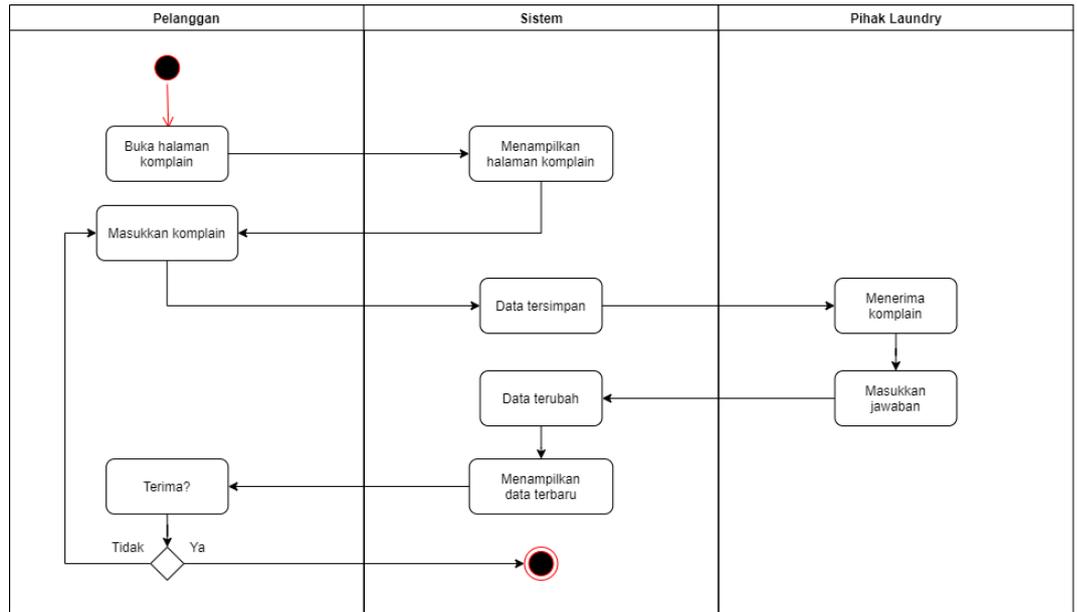
Pada gambar ini, menjelaskan bahwa pelanggan dapat melihat detail pesanannya seperti data pemesanan dan rincian item laundry. Selain itu, terdapat pilihan aksi sesuai status terakhir pemesanan.



Gambar 3.6 *Activity* Diagram Detail Pemesanan

c. Komplain

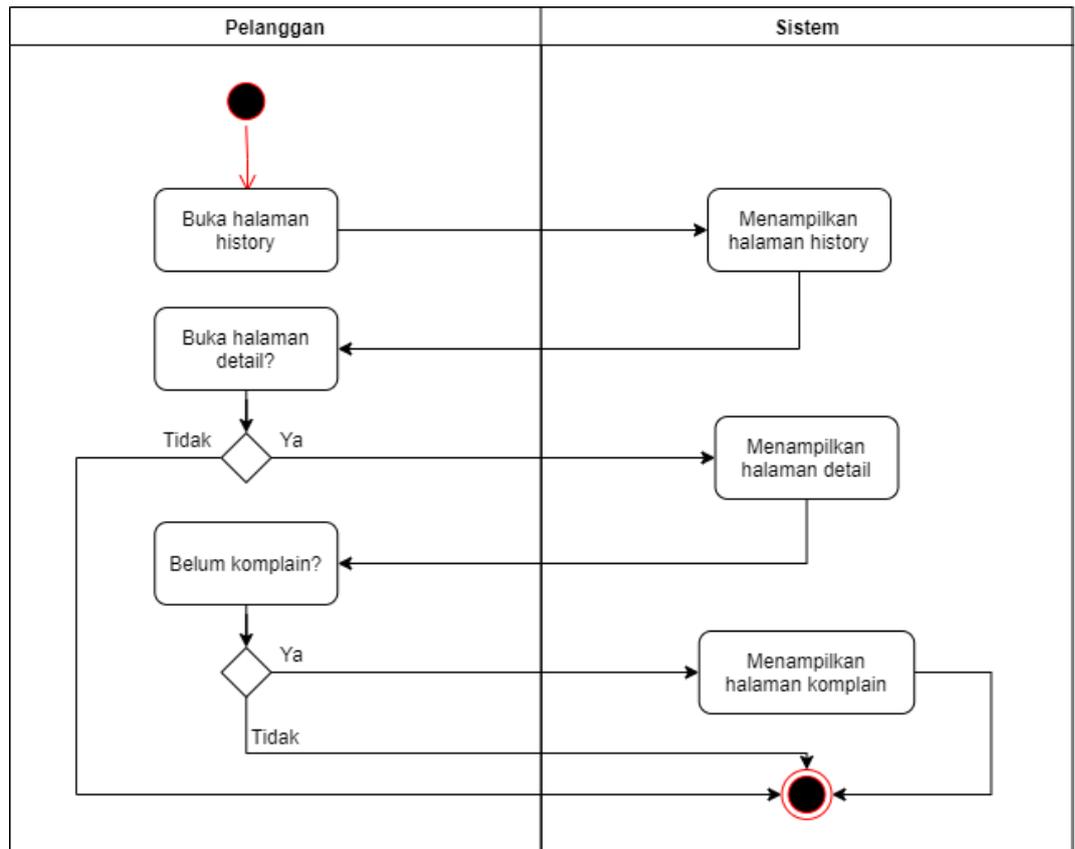
Pada gambar 3.7, dijelaskan bahwa pelanggan dapat mengajukan komplain pada pemesanannya dengan mengisi data terlebih dahulu. Data tersebut akan masuk ke pihak laundry dan di proses. Proses akan berulang jika pelanggan belum menganggap selesai.



Gambar 3.7 Activity Diagram Komplain

d. *History*

Pada gambar 3.8, dijelaskan bahwa pengguna dapat melihat progress laundrynya dengan membuka detail pada salah satu list. Dari halaman tersebut juga dapat membuat pengaduan apabila masih dalam batas waktu yang ditentukan.



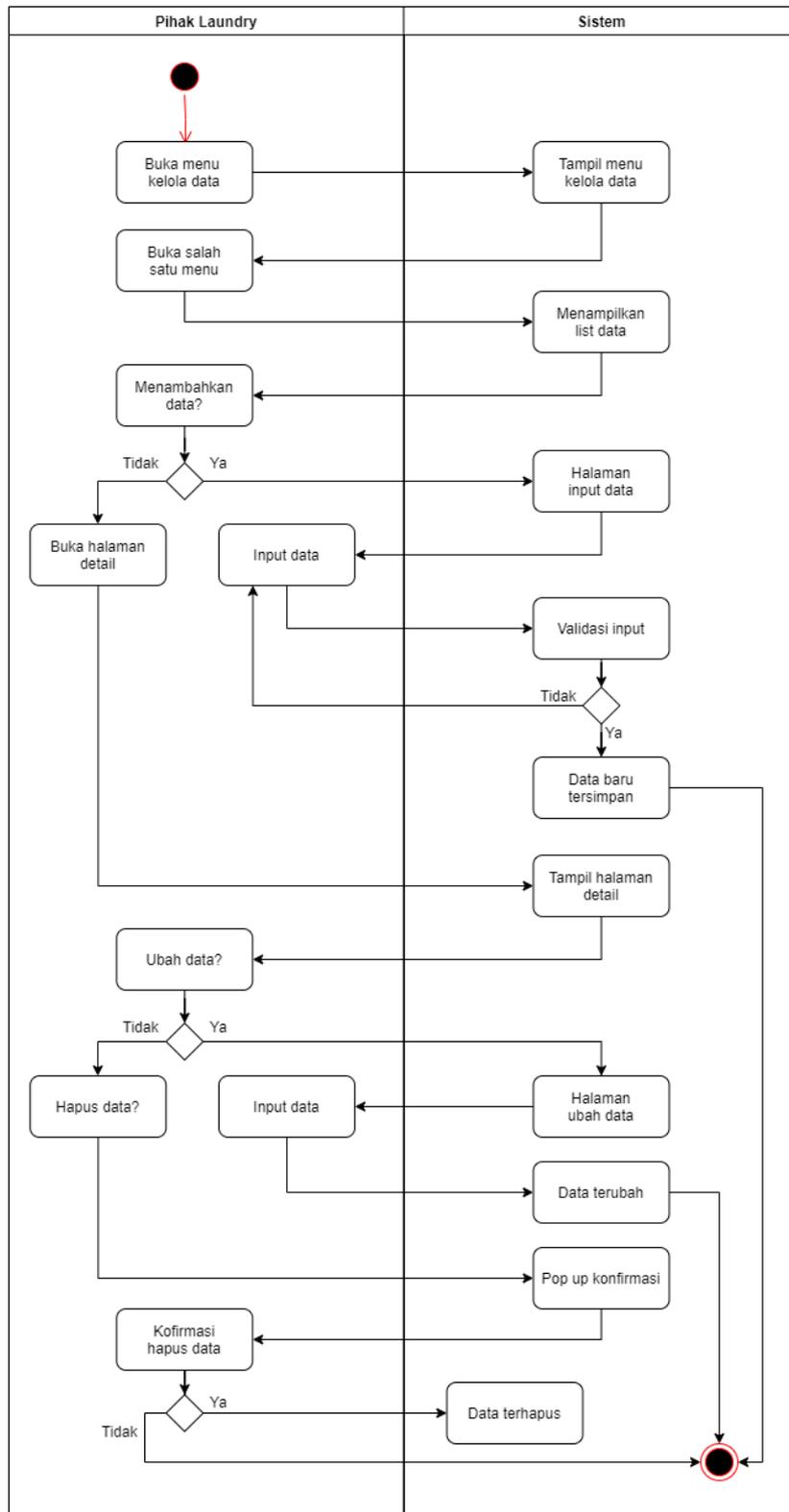
Gambar 3.8 Activity Diagram History

## 5. Activity Diagram Pihak Laundry

Fitur yang dapat diakses oleh pihak laundry adalah sebagai berikut :

### a. Kelola Data Master

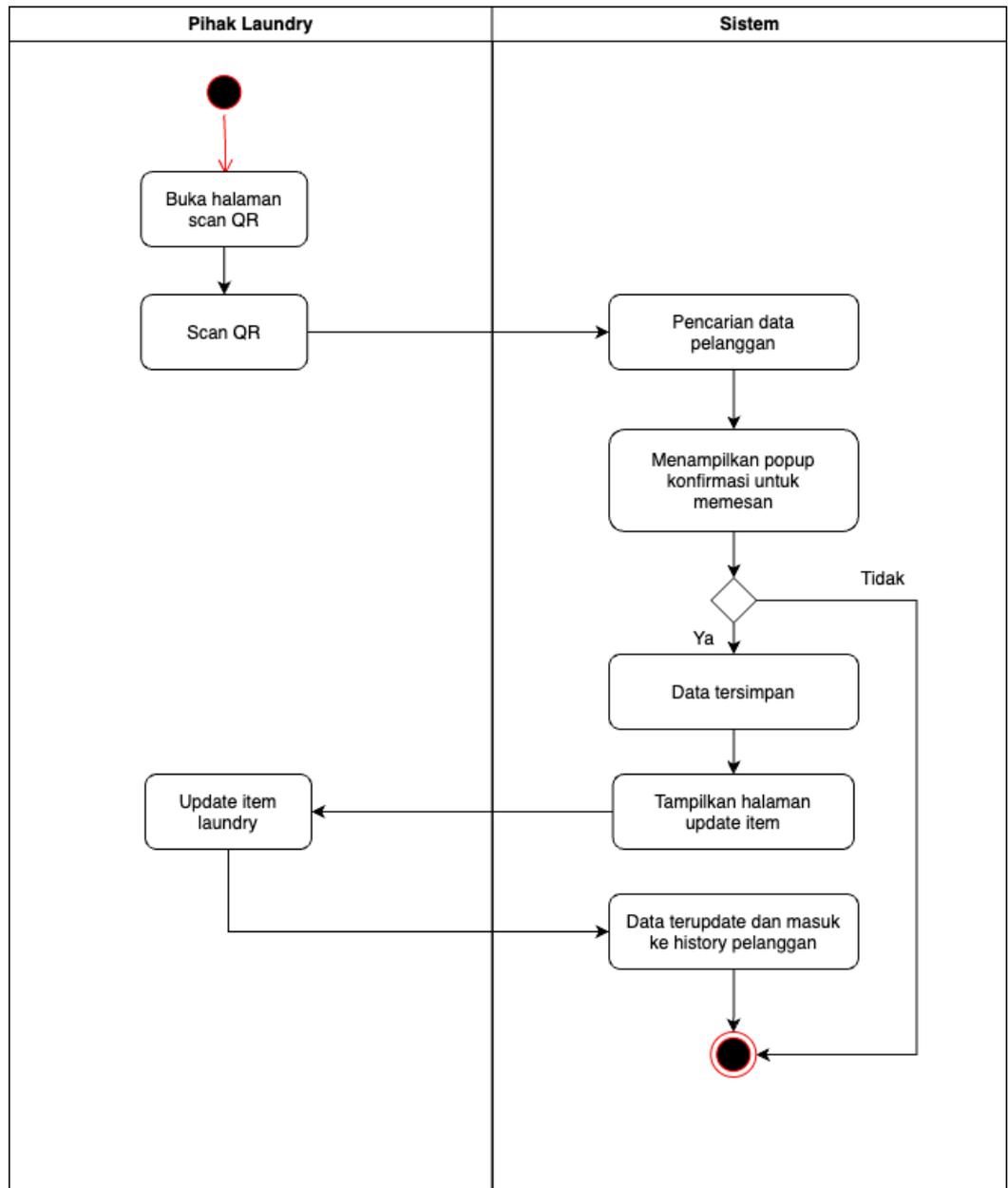
Pada gambar ini, dijelaskan pihak laundry dapat mengelola data master seperti melihat, menambahkan, mengubah, dan menghapus data.



Gambar 3.9 Activity Diagram Kelola Data Master

b. *Scan QR Code*

Pada gambar ini, dijelaskan bahwa pihak laundry tetap dapat mencatat transaksi laundry yang dilakukan secara offline (pelanggan datang langsung). Pihak laundry harus men-*scan barcode* dari pengguna terlebih dahulu, sistem akan menampilkan konfirmasi pemesanan. Jika ya, maka sistem menampilkan halaman *update item* pemesanan dan data akan tersimpan pada sistem serta data pemesanan akan masuk ke riwayat pelanggan.

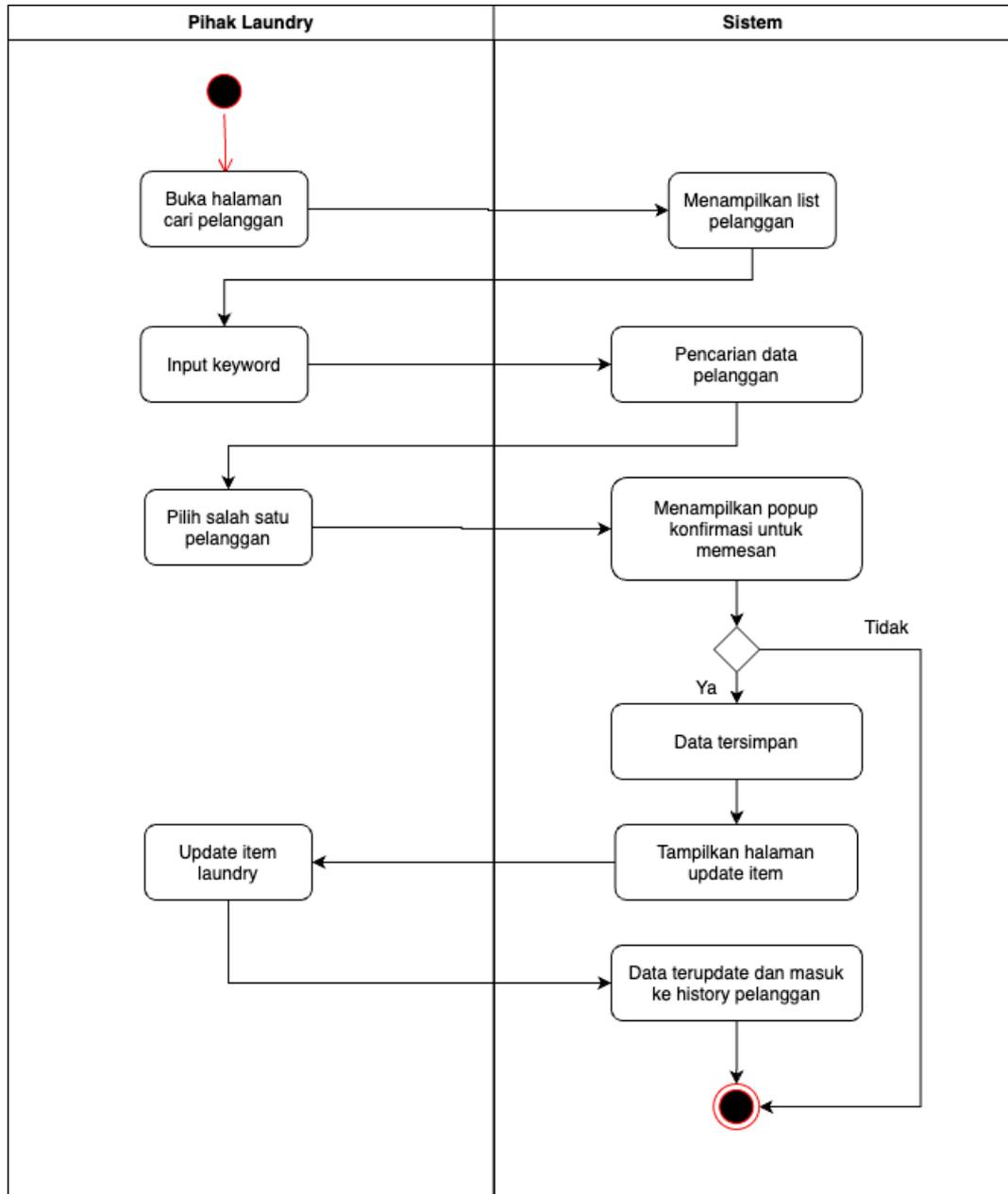


Gambar 3.10 Activity Diagram Scan QR Code

c. Order By Search

Pada gambar 3.11, dijelaskan bahwa pihak laundry dapat mencari akun pelanggan untuk menginput pemesanan. Fitur ini untuk menangani pelanggan

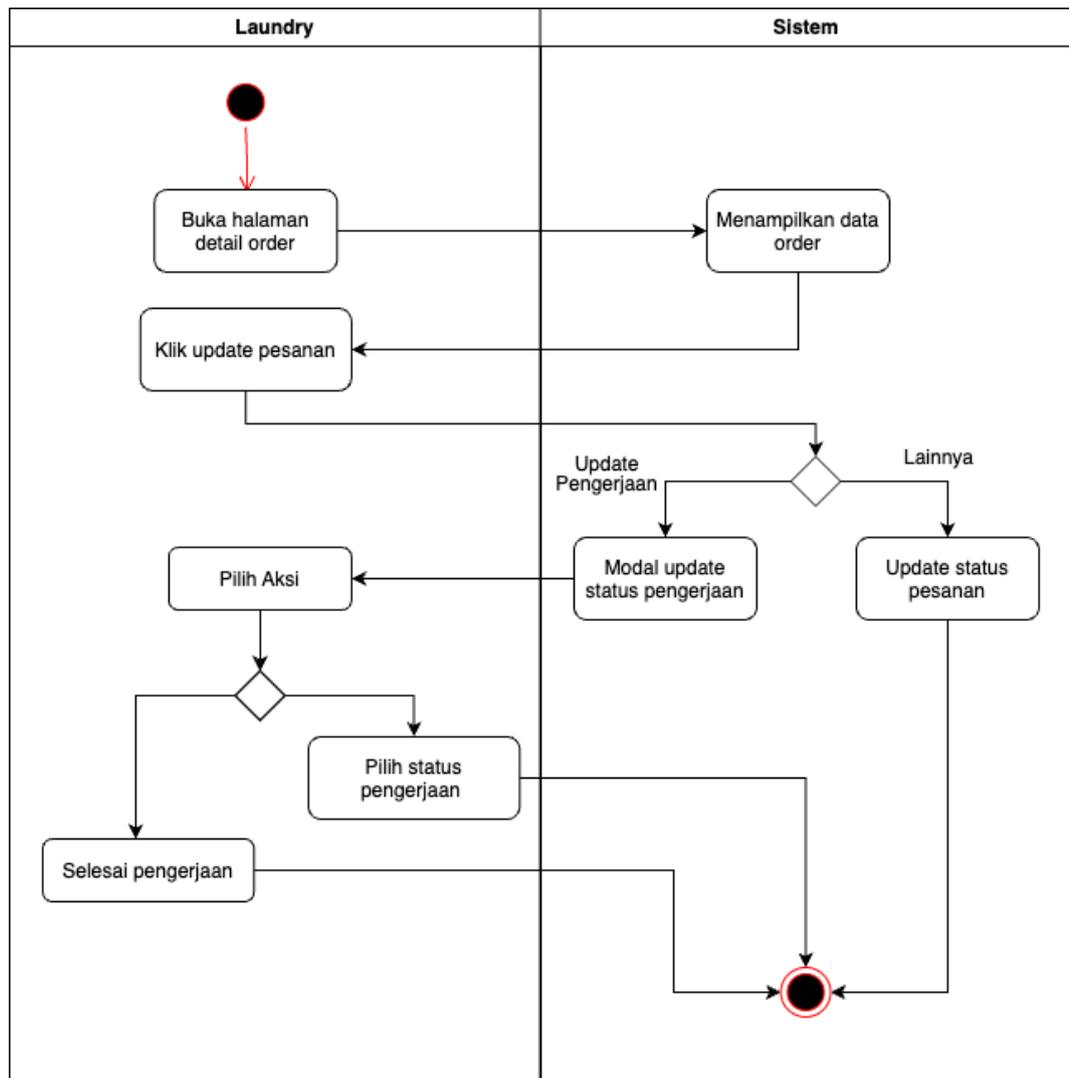
yang datang langsung namun tidak membawa hp. Laundry memasukkan nama pelanggan, lalu sistem akan menampilkan konfirmasi pesanan. Laundry meng-*update item* pesanan dan data akan tersimpan.



Gambar 3.11 Activity Diagram Order By Search

d. Detail Pemesanan

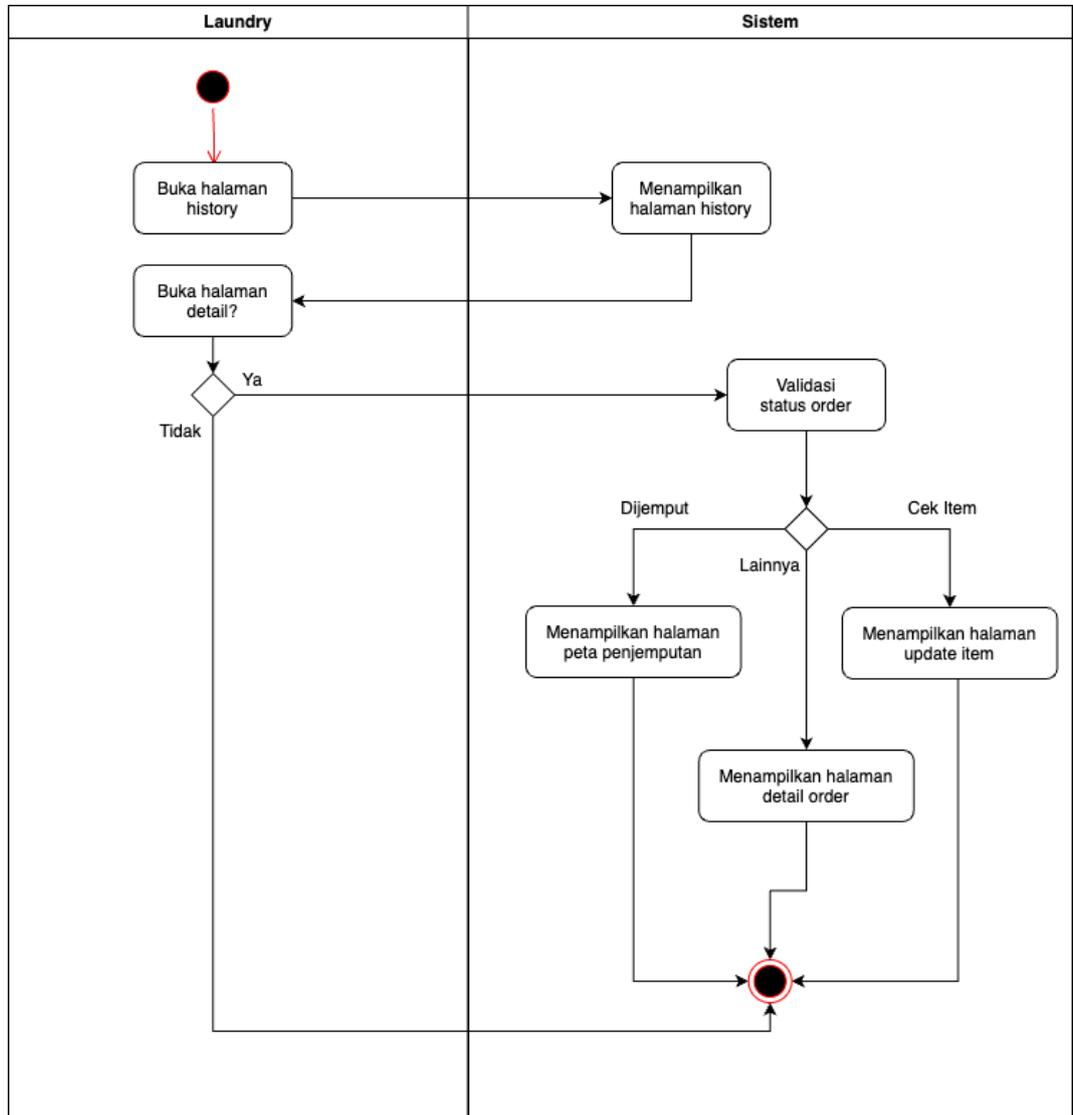
Gambar 3.12 menjelaskan bahwa laundry dapat melihat data pesanan seperti data pemesanan dan rincian item laundry serta meng-*update* status pesanan.



Gambar 3.12 Activity Diagram Detail Pemesanan (Laundry)

e. History

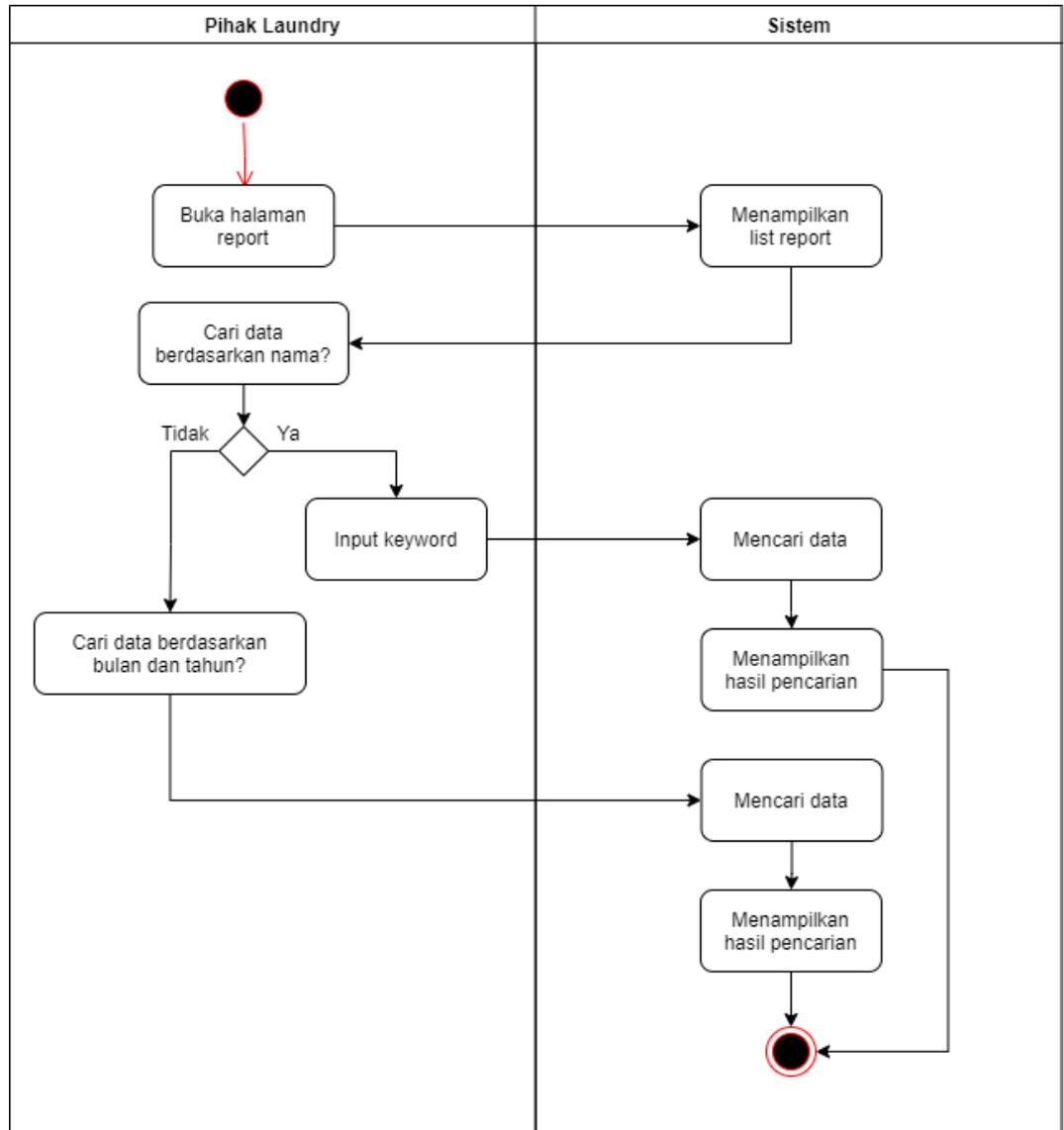
Pada gambar 3.13, dijelaskan bahwa laundry dapat menampilkan list history. Ketika membuka halaman detail akan diarahkan berdasarkan status order pelanggan.



Gambar 3.13 Activity Diagram History

f. *Report*

Gambar 3.14 menjelaskan bahwa pihak laundry dapat mencari data transaksi dan melihat laporan yang dapat di *filter* berdasarkan tanggal tahun yang dimasukkan.



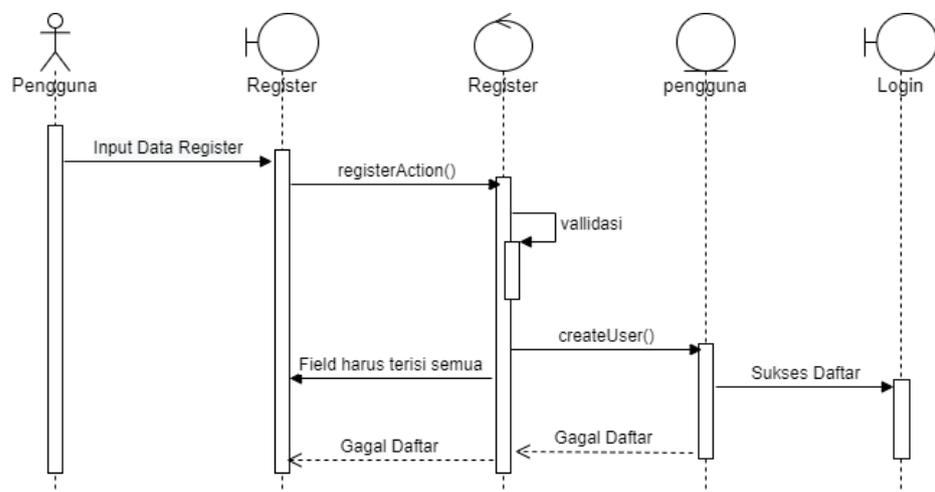
Gambar 3.14 Activity Diagram Report

### 3.2.1.3 Sequence Diagram

Setelah membuat *activity* diagram, langkah selanjutnya adalah menyusun *sequence* diagram. *Sequence* diagram berfungsi untuk menggambarkan dan menunjukkan secara rinci interaksi antara berbagai objek dalam sistem. Berikut *sequence* diagram dari *activity* diagram yang telah dibuat sebelumnya :

#### 1. Register

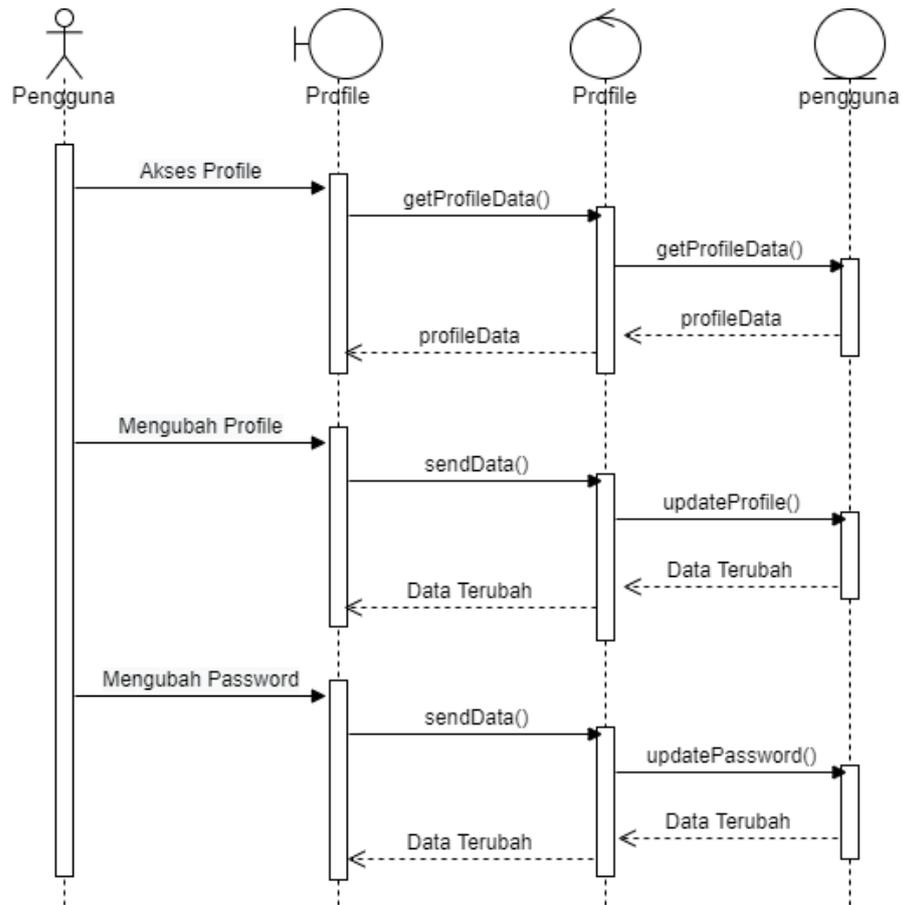
Pada *sequence* diagram 3.15, pelanggan harus memasukkan data register seperti nama lengkap, email, telepon, dan *password*. Data akan di proses pada control class dan di validasi. Jika data tidak valid, sistem akan mengirimkan pemberitahuan mengenai ketidakvalidan data tersebut. Jika data valid, maka proses diteruskan pada *database* untuk di simpan lalu di arahkan ke halaman *login*.



Gambar 3.15 *Sequence* Diagram Register







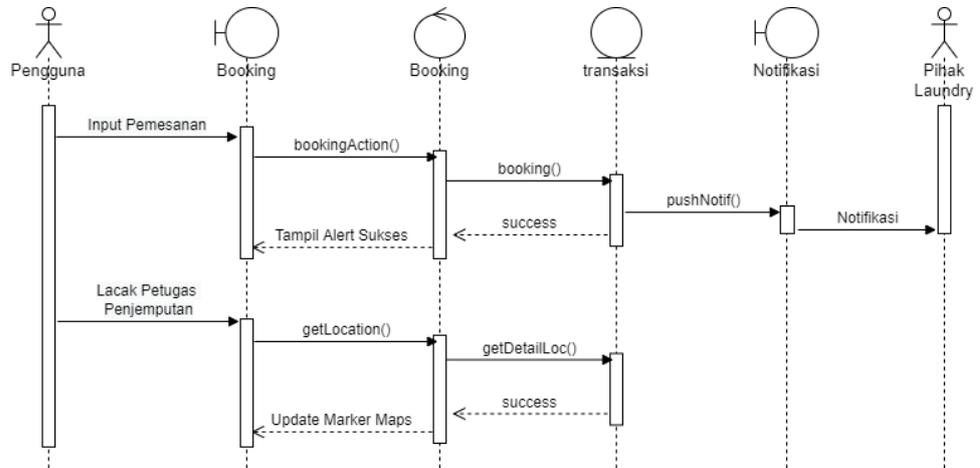
Gambar 3.17 *Sequence Diagram Profile*

#### 4. Hak Akses Pelanggan

##### a. Pemesanan

Pada *sequence* diagram 3.18, pelanggan melakukan transaksi yang akan di proses pada sistem dan di simpan pada *database*. Jika sukses menyimpan data, maka akan mengirimkan notifikasi sukses ke pelanggan. Sistem juga akan mengirimkan *push notification* ke *device* pihak laundry. Setelah itu, sistem akan mengarahkan

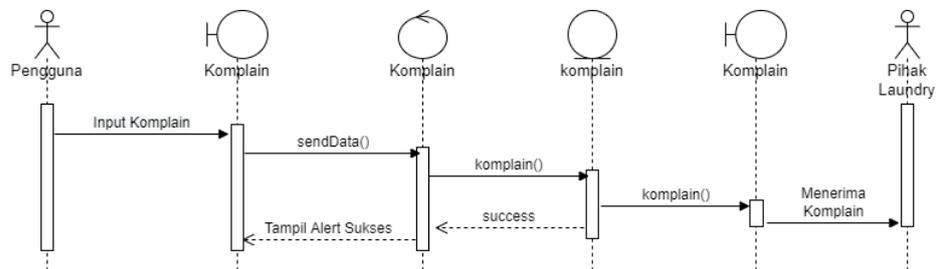
pelanggan ke halaman *tracking*. Pelanggan mendapat lokasi dari *database* dan sistem akan meng-*update marker* pada peta.



Gambar 3.18 *Sequence Diagram Pemesanan*

b. **Komplain**

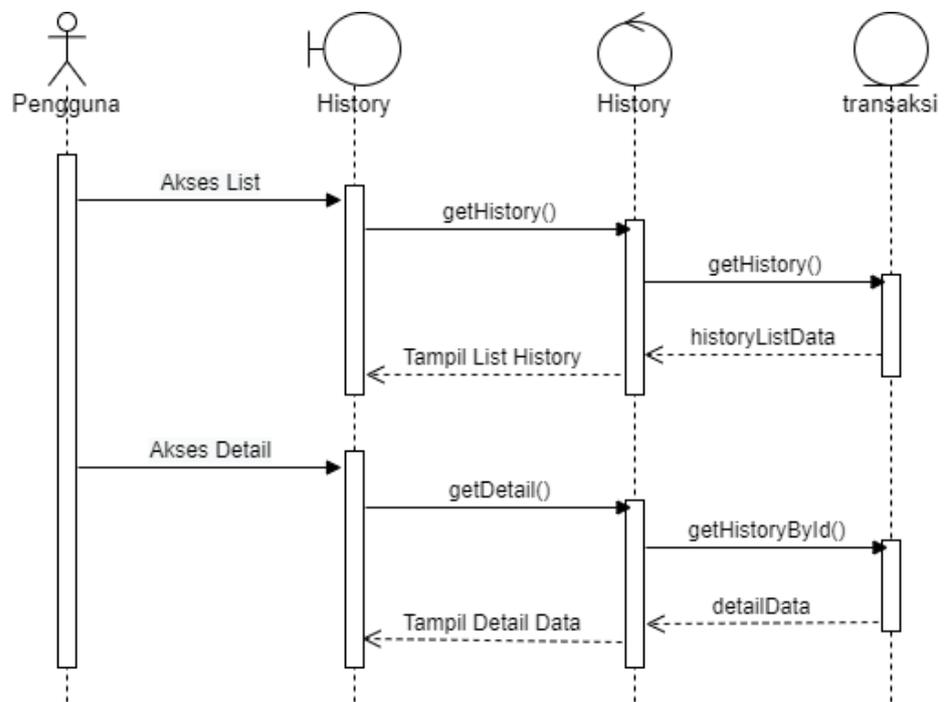
Pada *sequence diagram 3.19*, pelanggan memasukkan data *komplain* seperti kategori pengaduan, deskripsi, dan foto. Sistem akan memproses data tersebut dan di simpan pada *database*. Jika data berhasil di simpan, maka sistem akan mengirimkan *notifikasi* ke aplikasi. Selain itu, pihak laundry mendapat *list komplain* dari *database*.



Gambar 3.19 *Sequence Diagram Komplain*

c. *History*

Pada sequence diagram 3.20, pengguna mengakses halaman *list* yang akan di proses sistem dan mengeluarkan data dari *database*. Selain itu, pengguna dapat melihat detail dari transaksi yang telah dilakukan. Sistem memproses dengan mengirimkan ID ke database sebagai parameter. Jika data berhasil di dapatkan, sistem akan menampilkannya di aplikasi.

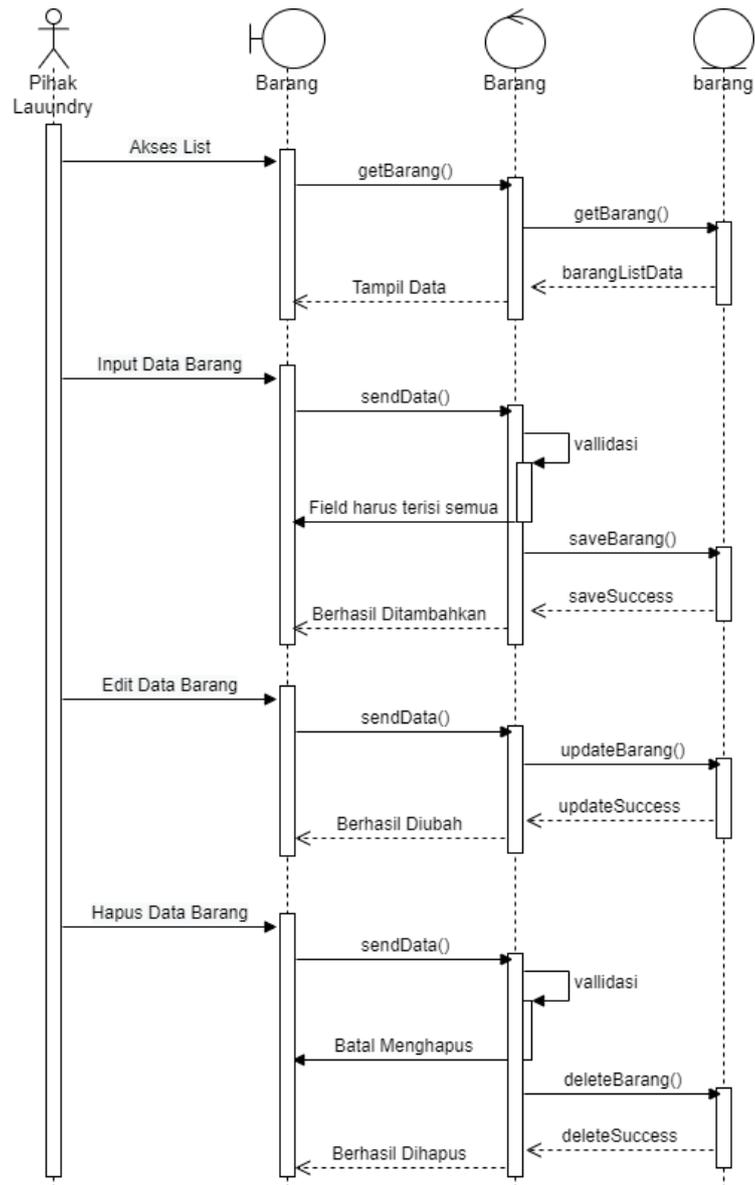


Gambar 3.20 *Sequence Diagram History*

5. Hak Akses Pihak Laundry

a. Kelola Data Master

Pada sequence diagram 3.21, pihak laundry halaman *list* yang akan di proses sistem dan mengeluarkan data dari *database*. Pihak laundry dapat menambahkan item dengan memasukkan data pada form. Sistem akan memproses data tersebut dengan memvalidasi semua *field* terisi atau tidak. Jika semua *field* terisi, sistem akan menambahkan data ke tabel. Jika *field* belum terisi, sistem akan mengirimkan pemberitahuan bahwa semua *field* harus diisi. Selain itu, pihak laundry juga dapat mengubah maupun menghapus data pada tabel items.

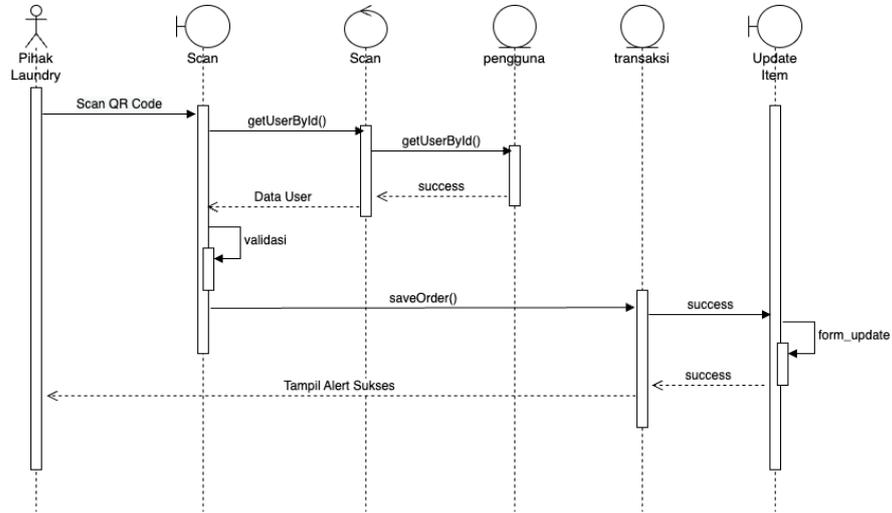


Gambar 3.21 *Sequence Diagram* Kelola Data Master

b. Scan QR Code

Pada sequence diagram 3.22, pihak laundry memindai kode QR dan mengirimkan hasilnya pada sistem. Setelah data ditemukan, sistem akan mengirimkan notifikasi konfirmasi pada pihak laundry.

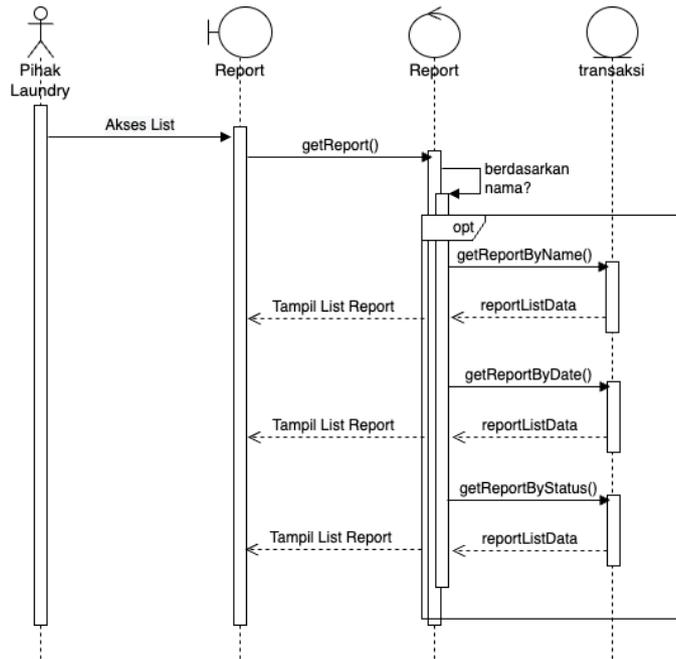
Jika ingin melanjutkan pemesanan, sistem akan menyimpan data pesanan ke dalam *database* dan kemudian mengarahkan tampilan ke formulir pembaruan.



Gambar 3.22 *Sequence Diagram* Scan QR Code

*c. Report*

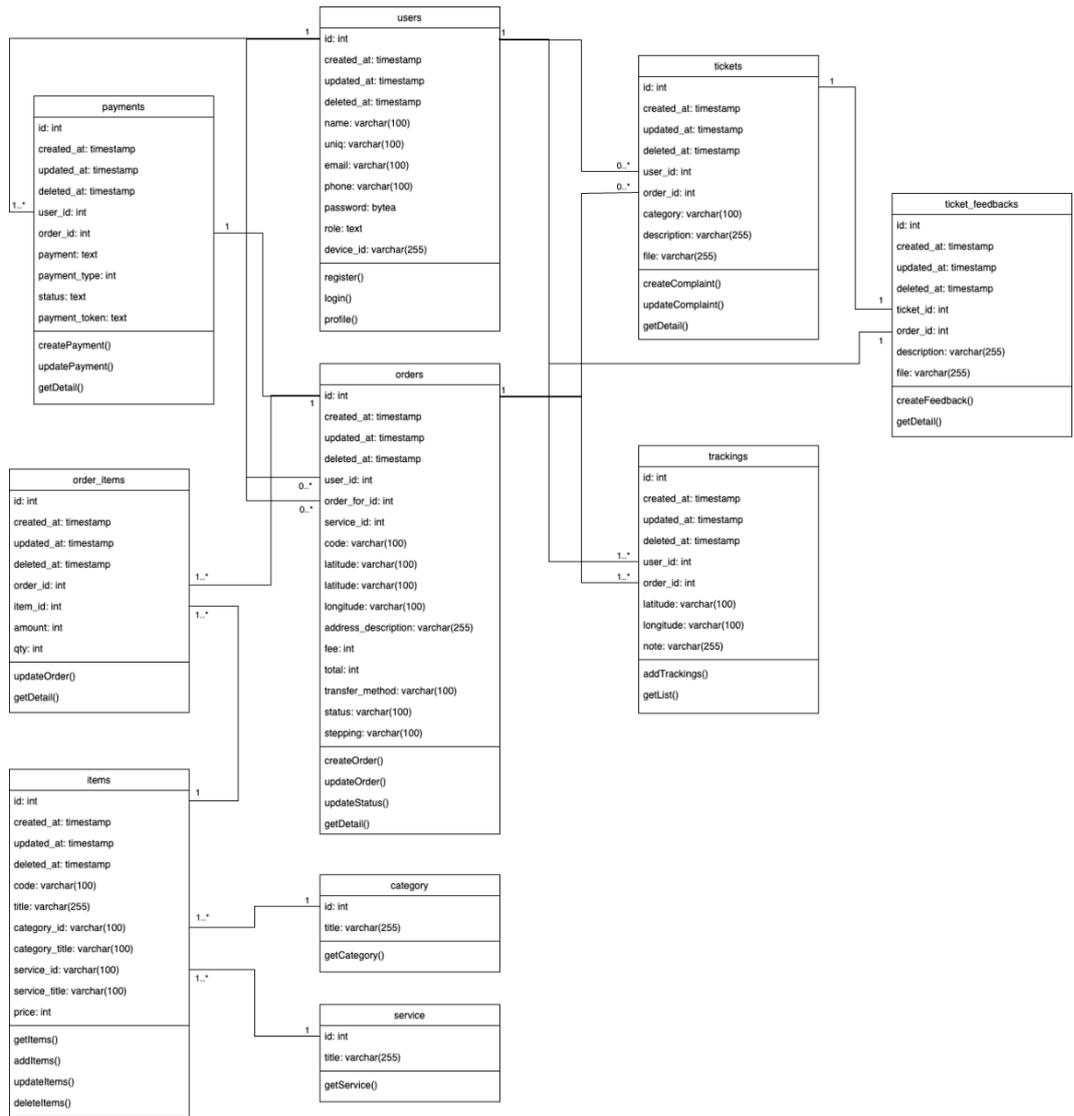
Pada *sequence diagram* 3.23, pihak laundry halaman *list* yang akan di proses sistem dan mengeluarkan data dari *database*. Data tersebut dapat di *filter* berdasarkan nama laporan, rentang tanggal transaksi, dan status pemesanan. Jika data berhasil di *filter*, maka sistem akan mengirimkan *list* terbaru untuk ditampilkan.



Gambar 3.23 *Sequence Diagram Report*

### 3.2.1.4 Class Diagram

Gambar 3.24 menunjukkan class diagram dari aplikasi laundry. Class diagram merupakan salah satu jenis UML yang menggambarkan kelas, atribut, metode, dan hubungan antar objek. Fungsinya adalah membantu peneliti dalam mengembangkan sistem aplikasi laundry karena memberikan gambaran umum skema aplikasi dan dapat menyampaikan kebutuhan dari suatu sistem.



Gambar 3.24 Class Diagram

Berikut penjelasan dari class diagram pada gambar diatas :

a. Users

Class users memiliki atribut id, created\_at, updated\_at, deleted\_at, name, uniq, email, phone, password, role, dan device\_id. Operasi pada class tersebut antara lain register, login, dan profile. Atribut yang bersifat mandatory antara lain id, created\_at, updated\_at, deleted\_at, name, uniq, email, phone, password, role.

#### b. Category

Class category memiliki atribut id dan name. Operasi pada class tersebut adalah getCategory. Semua atribut tersebut bersifat mandatory.

#### c. Service

Class service memiliki atribut id dan name. Operasi pada class tersebut adalah getService. Semua atribut tersebut bersifat mandatory.

#### d. Items

Class items memiliki atribut id, created\_at, updated\_at, deleted\_at, code, title, category\_id, category\_title, service\_id, service\_title, dan price. Operasi pada class tersebut antara lain getItem, addItem, updateItem, dan deleteItem. Class ini memiliki relasi dengan class service dan class category. Semua atribut tersebut bersifat mandatory.

#### e. Orders

Class orders memiliki atribut id, created\_at, updated\_at, deleted\_at, user\_id, order\_for\_id, service\_id, code, latitude, longitude, address\_description, fee, total, transfer\_method, status, dan stepping. Operasi pada class tersebut antara lain createOrder, updateOrder, updateStatus, dan getDetail. Class ini memiliki relasi dengan class users. Atribut yang bukan mandatory adalah address\_description, selain itu merupakan atribut yang mandatory.

#### f. Order Items

Class order items memiliki atribut id, created\_at, updated\_at, deleted\_at, order\_id, item\_id, amount, dan qty. Operasi pada class tersebut antara lain updateOrder, dan getDetail. Class ini memiliki relasi dengan class orders dan class items. Semua atribut tersebut bersifat mandatory.

#### g. Trackings

Class trackings memiliki atribut id, created\_at, updated\_at, deleted\_at, user\_id, order\_id, latitude, longitude, dan note. Operasi pada class tersebut antara lain addTracking, dan getList. Class ini memiliki relasi dengan class orders dan class users. Atribut yang bukan mandatory adalah note, selain itu merupakan atribut yang mandatory.

#### h. Payments

Class payments memiliki atribut id, created\_at, updated\_at, deleted\_at, user\_id, order\_id, payment, payment\_type, status, dan payment\_token. Operasi pada class tersebut antara lain createPayment, updatePayment, dan getDetail. Class ini memiliki relasi dengan class orders dan class users. Semua atribut tersebut bersifat mandatory.

#### i. Tickets

Class tickets memiliki atribut id, created\_at, updated\_at, deleted\_at, user\_id, order\_id, category, description, dan file. Operasi pada class tersebut antara lain createComplaint, updateComplaint, dan getDetail. Class ini memiliki relasi dengan class orders dan class users. Atribut yang bukan mandatory adalah file, selain itu merupakan atribut yang mandatory.

#### j. Ticket Feedbacks

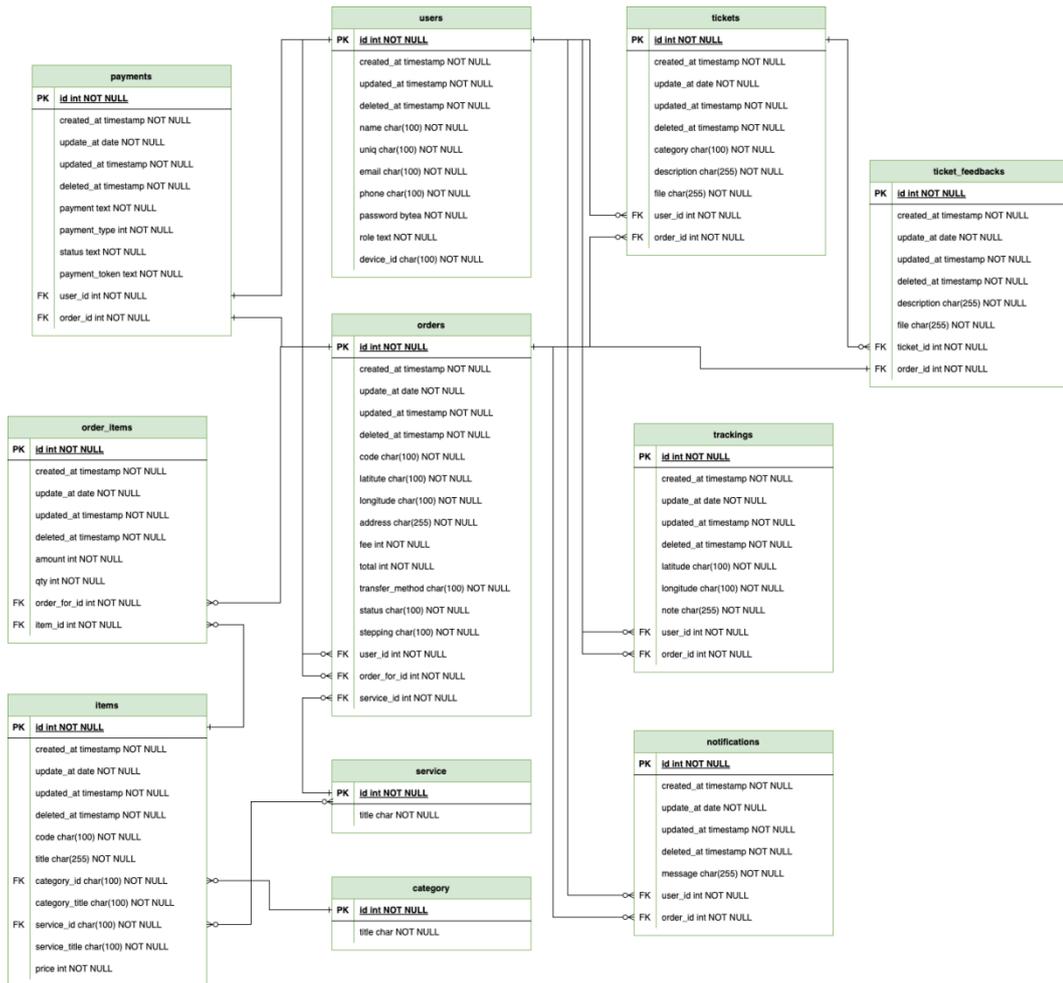
Class ticket feedbacks memiliki atribut id, created\_at, updated\_at, deleted\_at, ticket\_id, order\_id, description, dan file. Operasi pada class tersebut antara lain createFeedback, dan getDetail. Class ini memiliki relasi dengan class orders dan class tickets. Atribut yang bukan mandatory adalah file, selain itu merupakan atribut yang mandatory.

### 3.2.1.5 Entity Relational Diagram (ERD)

Pada gambar 3.25, terdapat 11 tabel dimana tabel master berjumlah 4 dan tabel transaksi berjumlah 7. Berikut penjelasan dari relasi yang ada :

- a. Tabel items berelasi dengan tabel services mempunyai kardinalitas many to many karena satu service di punyai banyak item begitu sebaliknya.
- b. Tabel items memiliki hubungan dengan tabel category dengan kardinalitas one to many karena satu kategori (kiloan dan satuan) mempunyai banyak item barang.
- c. Tabel users dengan tabel orders mempunyai kardinalitas one to many karena satu pelanggan dapat memesan lebih dari satu jasa laundry.
- d. Tabel orders dengan tabel order\_items mempunyai kardinalitas one to many karena pada satu order mempunyai detail items yang beragam.
- e. Tabel orders dengan tabel services mempunyai kardinalitas one to many karena satu service bisa di pesan berulang kali.

- f. Tabel trackings dengan tabel orders mempunyai kardinalitas one to many karena satu order dapat mempunyai banyak location tracking yang di inputkan.
- g. Tabel trackings berelasi dengan tabel users yang mempunyai kardinalitas one to many karena satu users yang memesan jasa laundry mempunyai banyak location tracking.
- h. Tabel payments dengan tabel orders mempunyai kardinalitas one to one karena satu order hanya dapat dilakukan satu pembayaran.
- i. Tabel payments dengan tabel users mempunyai kardinalitas one to one karena satu payment dipunyai satu order dengan spesifik satu user.
- j. Tabel notification dengan tabel orders mempunyai kardinalitas one to many karena banyak notifikasi dipunyai satu order dengan spesifik satu user.



Gambar 3.25 ERD

### 3.2.1.5.1 Struktur Tabel

Untuk menjelaskan secara detail setiap atribut yang ada di setiap entitas dibuatlah struktur tabel. Berikut penjelasan dari tabel-tabel yang ada :

#### a. Tabel users

Tabel users digunakan untuk menyimpan informasi pengguna aplikasi yaitu pelanggan dan pihak laundry.

Tabel 3.4 Tabel Users

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik pengguna yang terdaftar di aplikasi. Selain itu, id digunakan untuk menghubungkan ke tabel transaksi.
2	created_at	timestamp		Tanggal saat pengguna melakukan registrasi.
3	updated_at	timestamp		Tanggal perubahan data users.
4	deteled_at	timestamp		Tanggal saat data users dihapus.
5	name	varchar(100)		Nama lengkap yang dimasukkan saat registrasi.
6	uniq	varchar(100)		Kode unik dari users.
7	email	varchar(100)		Email dari pengguna yang dapat digunakan untuk masuk ke aplikasi.
8	phone	varchar(100)		Nomor telepon dari pengguna yang dapat digunakan untuk masuk ke aplikasi.
9	password	bytea		Password yang di masukkan pengguna sebagai kode keamanan akun.
10	role	text		Tipe pengguna saat masuk aplikasi.
11	device_id	varchar(255)		ID device untuk push notifikasi onesignal.

b. Tabel items

Tabel items untuk menyimpan data jenis, kategori, nama item, dan harga per itemnya.

Tabel 3.5 Tabel Items

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik sebagai identitas pada tabel item. Selain itu, id digunakan untuk menghubungkan ke tabel transaksi.
2	created_at	timestamp		Tanggal dan waktu penambahan item.
3	updated_at	timestamp		Tanggal dan waktu perubahan data item.
4	deteled_at	timestamp		Tanggal dan waktu saat data item dihapus.
5	code	varchar(100)		Kode unik dari items..
6	title	varchar(255)		Nama dari items.
7	category_id	varchar(100)	Foreign	ID dari kategori kiloan atau satuan.
8	category_title	varchar(100)		Nama dari kategori kiloan atau satuan.
9	service_id	varchar(100)	Foreign	ID dari jenis cuci laundry.
10	service_name	varchar(100)		Nama dari jenis cuci laundry.
11	price	int8		Harga item sesuai dengan kategori dan jenis cuci.

c. Tabel orders

Tabel orders untuk menyimpan data transaksi pemesanan.

Tabel 3.6 Tabel Orders

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik sebagai identitas pada tabel order.
2	created_at	timestamp		Tanggal dan waktu penambahan order.

3	updated_at	timestamp		Tanggal dan waktu perubahan data order.
4	deteled_at	timestamp		Tanggal dan waktu saat data order dihapus.
5	user_id	int8	Foreign	ID dari tabel users sebagai identitas pihak laundry.
6	order_for_id	int8	Foreign	ID dari tabel users sebagai identitas pemesan.
7	service_id	int8	Foreign	ID dari data services.
8	code	varchar(100)		Kode unik tabel orders.
9	latitude	varchar(100)		Latitude lokasi dari users yang melakukan pemesanan.
10	longitude	varchar(100)		Longitudelokasi dari users yang melakukan pemesanan.
11	address_description	varchar(255)		Alamat sesuai longitude dan latitude.
12	fee	int8		Biaya pengantaran untuk layanan antar jemput.
13	total	int8		Total harga pemesanan kalkulasi dari tabel order_items.
14	transfer_method	varchar(100)		Jenis layanan antar jemput atau ambil di tempat.
15	status	varchar(100)		Status pemesanan laundry.
16	stepping	varchar(100)		Status pengerjaan dari pemesanan laundry.

d. Tabel order\_items

Tabel order\_items untuk detail item dari tabel order..

Tabel 3.7 Tabel Order\_items

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik sebagai identitas pada tabel orders. Selain itu, id digunakan untuk menghubungkan ke tabel transaksi.
2	created_at	timestamp		Tanggal dan waktu penambahan order.
3	updated_at	timestamp		Tanggal dan waktu perubahan data order.
4	deteled_at	timestamp		Tanggal dan waktu saat data order dihapus.
5	order_id	int8	Foreign	ID dari tabel orders.
6	item_id	int8	Foreign	ID dari tabel items..
7	amount	int8		Total harga dari kalkulasi kuantitas dikali harga item.
8	qty	int8		Kuantitas pakaian dalam kiloan maupun satuan.

e. Tabel trackings

Tabel trackings untuk menyimpan perubahan lokasi dari petugas pengantaran laundry yang dapat dilihat pelanggan.

Tabel 3.8 Tabel Trackings

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik sebagai identitas pada tabel trackings. Selain itu, id digunakan untuk menghubungkan ke tabel transaksi.

2	created_at	timestamp		Tanggal dan waktu penambahan data.
3	updated_at	timestamp		Tanggal dan waktu perubahan data.
4	deteled_at	timestamp		Tanggal dan waktu saat data dihapus.
5	user_id	int8	Foreign	ID dari tabel users.
6	order_id	int8	Foreign	ID dari tabel orders.
7	latitude	varchar(100)		Latitude lokasi petugas laundry.
8	longitude	varchar(100)		Longitude lokasi petugas laundry.
9	note	varchar(255)		Deskripsi lokasi petugas.

f. Tabel payments

Tabel payments berfungsi untuk menyimpan informasi pembayaran pelanggan.

Tabel 3.9 Tabel Payments

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik sebagai identitas pada tabel payments. Selain itu, id digunakan untuk menghubungkan ke tabel transaksi.
2	created_at	timestamp		Tanggal penambahan data.
3	updated_at	timestamp		Tanggal perubahan data.
4	deteled_at	timestamp		Tanggal saat data dihapus.
5	user_id	int8	Foreign	ID dari tabel users.
6	order_id	int8	Foreign	ID dari tabel orders.
7	payment	text		Total dari pembayaran.
8	payment_type	int8		Tipe pembayaran tunai dan non tunai.
9	status	text		Status pembayaran transaksi.

10	payment_token	text		Token untuk pembayaran melalui payment gateway.
----	---------------	------	--	---

g. Tabel tickets

Tabel tickets untuk menyimpan data pengaduan pelanggan terhadap kesalahan pihak laundry.

Tabel 3.10 Tabel Tickets

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik sebagai identitas pada tabel tickets. Selain itu, id digunakan untuk menghubungkan ke tabel transaksi.
2	created_at	timestamp		Tanggal penambahan data.
3	updated_at	timestamp		Tanggal perubahan data.
4	deteled_at	timestamp		Tanggal saat data dihapus.
5	user_id	int8	Foreign	ID dari tabel users.
6	order_id	int8	Foreign	ID dari tabel orders.
7	category	varchar(100)		Kategori jenis pengaduan
8	description	varchar(255)		Deskripsi pengaduaan pelanggan.
9	file	varchar(255)		File gambar dengan base64.

h. Tabel ticket\_feedbacks

Tabel ticket\_feedbacks untuk menyimpan data feedback dari pihak laundry terhadap pengaduan pelanggan.

Tabel 3.11 Tabel Ticket\_feedbacks

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik sebagai identitas pada tabel ticket_feedbacks. Selain itu, id digunakan untuk menghubungkan ke tabel tickets.
2	created_at	timestamp		Tanggal penambahan data.
3	updated_at	timestamp		Tanggal perubahan data.
4	deteled_at	timestamp		Tanggal saat data dihapus.
5	ticket_id	int8	Foreign	ID dari tabel tickets.
6	order_id	int8	Foreign	ID dari tabel orders.
7	description	varchar(255)		Deskripsi feedback laundry.
8	file	varchar(255)		File gambar dengan base64.

i. Tabel notifications

Tabel notifications untuk menyimpan data notifikasi yang dikirim setiap ada transaksi maupun saat adanya perubahan status.

Tabel 3.12 Tabel Notifications

No	Field	Type	Key	Keterangan
1	id	int8	Primary	Id unik sebagai identitas pada tabel ticket_feedbacks. Selain itu, id digunakan untuk menghubungkan ke tabel tickets.
2	created_at	timestamp		Tanggal penambahan data.
3	updated_at	timestamp		Tanggal perubahan data.
4	deteled_at	timestamp		Tanggal saat data dihapus.
5	user_id	int8	Foreign	ID dari tabel users.

6	order_id	int8	Foreign	ID dari tabel orders.
7	message	varchar(255)		Deskripsi notifikasi.

### 3.2.1.6 User Interface / Mock-up Aplikasi

Perancangan UI dilakukan agar rancangan dari *activity* diagram tertuangkan ke dalam desain sesuai *input* dan *output*-nya. Desain UI berupa teks, gambar, layouting, dan sebagainya yang di design menggunakan *software* khusus *wireframing*. Desain ini terbagi dalam dua tipe untuk pelanggan dan pihak laundry.

#### 3.2.1.6.1 Wireframe Pelanggan

##### a. Halaman Pendaftaran

Halaman pendaftaran menampilkan form pendaftaran yang harus diisi untuk menggunakan aplikasi ini. Form pendaftaran terdiri dari input nama, email, nomer telepon, *password* dan konfirmasi *password*.

9:41

FIDI Laundry

Masukkan nama lengkap

Masukkan email

Masukkan nomer telepon

Masukkan password

Masukkan konfirmasi password

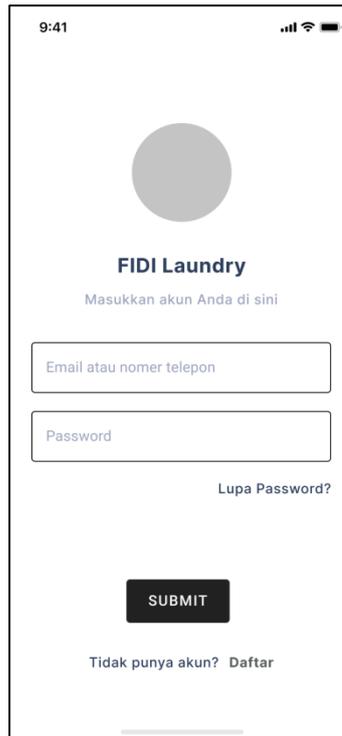
REGISTER

Sudah punya akun? Masuk

Gambar 3.26 Halaman Pendaftaran

b. Halaman *Login*

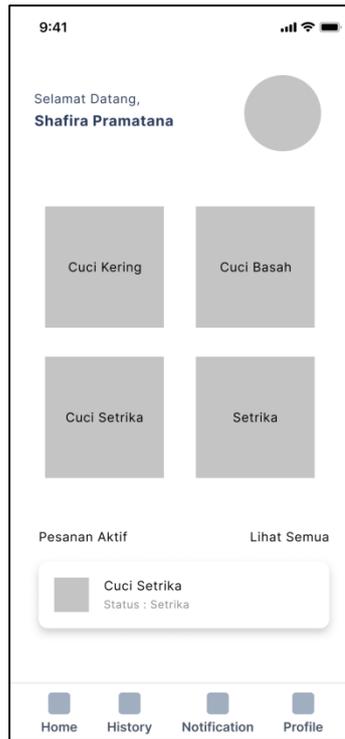
Halaman *login* terdiri dari memasukkan email dan *password*, klik *login* untuk masuk ke dalam aplikasi. Terdapat label “Lupa password?” yang berfungsi mengarahkan ke halaman lupa *password*.



Gambar 3.27 Halaman *Login*

c. Halaman *Home*

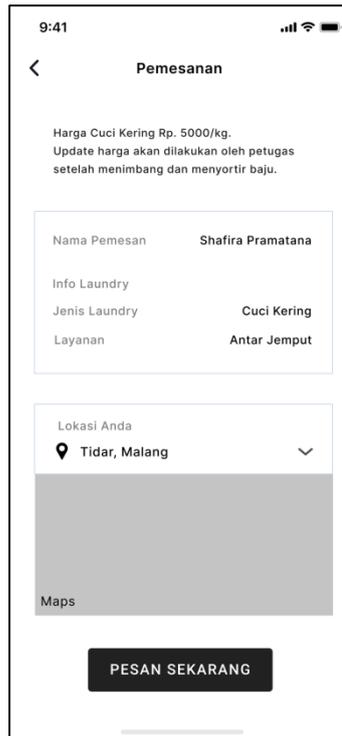
Halaman *home* berisi informasi akun pengguna, menu pemesanan laundry, dan pesanan yang aktif. Untuk memesan jasa laundry melalui aplikasi dapat memilih jenis layanan terlebih dahulu, kemudian aplikasi akan dibawa ke halaman pemesanan.



Gambar 3.28 Halaman *Home*

d. Halaman Pemesanan

Halaman pemesanan memuat informasi data pengguna, pemesanan laundry yang telah dipilih sebelumnya, dan lokasi dari pengguna yang akan memudahkan petugas penjemputan.



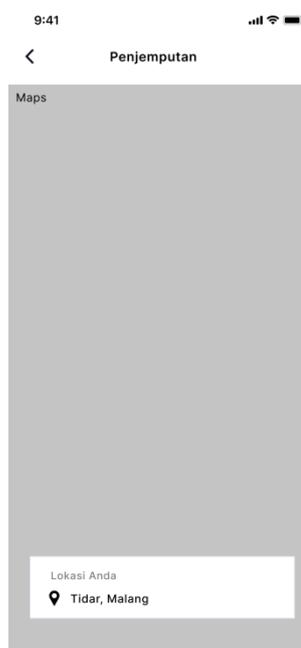
Gambar 3.29 Halaman Pemesanan

Setelah klik “Pesan Sekarang” akan tampil konfirmasi apakah pelanggan ingin melanjutkan pemesanan atau tidak.



Gambar 3.30 *Pop-up* Konfirmasi Pemesanan

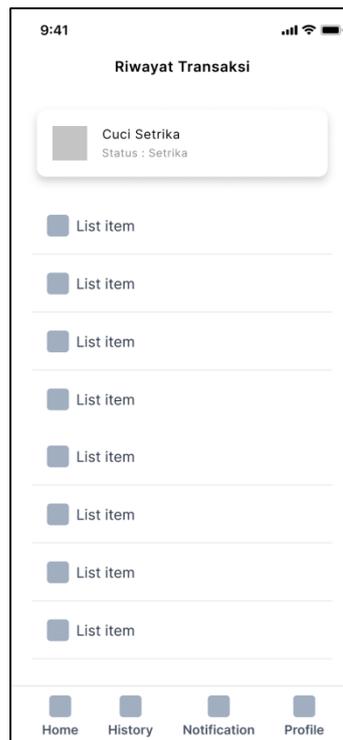
Jika pelanggan melanjutkan pemesanan, maka aplikasi akan diarahkan ke halaman pelacakan petugas yang berisi marker dari pelanggan dan petugas.



Gambar 3.31 Halaman *Tracking* Penjemputan

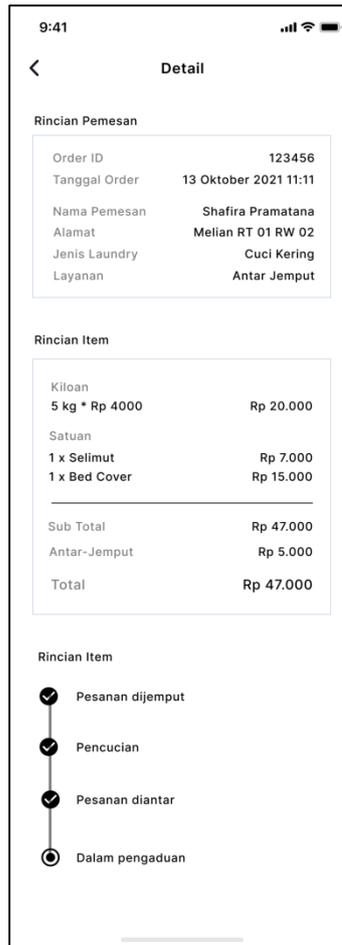
e. Halaman *History*

Halaman *history* memuat data transaksi pelanggan, tampilan *card* per *item*-nya di bedakan berdasarkan status pesanan yaitu *in-progress* atau telah selesai.



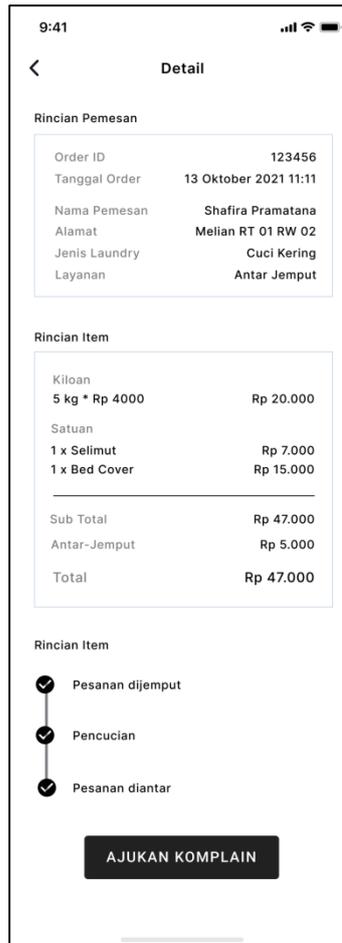
Gambar 3.32 Halaman *History*

Untuk melihat detail transaksi, pelanggan dapat mengklik salah satu data. Pada halaman ini memuat informasi transaksi berupa rincian pemesanan, rincian item laundry, maupun status dari laundry. Gambar 3.34 merupakan halaman detail transaksi yang transaksinya belum selesai.



Gambar 3.33 Halaman Detail *History*

Gambar 3.35 merupakan halaman detail transaksi pesanan telah selesai. Pada halaman ini, pelanggan dapat mengajukan komplain jika dalam batas waktu yang telah ditentukan.



Gambar 3.34 Halaman Detail *History*

Aplikasi akan di arahkan ke halaman form komplain yang terdiri dari pilihan kategori komplain, keterangan, dan bukti pendukung berupa foto.

9:41

< Pengaduan

Kategori Pengaduan  
Dropdown closed

Masukkan keterangan

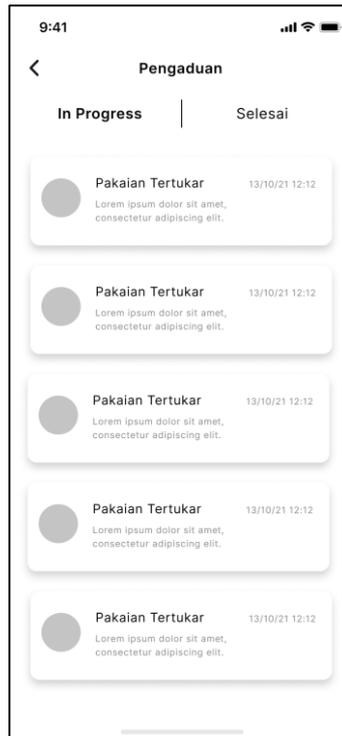
Bukti Pendukung PILIH

AJUKAN KOMPLAIN

Gambar 3.35 Halaman Form Komplain

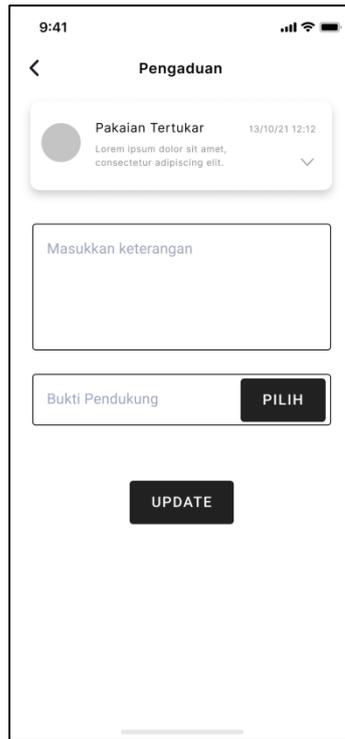
f. Halaman Komplain/Pengaduan

Halaman komplain memuat list komplain dari pelanggan yang tersimpan di database, halaman ini dapat diakses melalui tab *profile*.



Gambar 3.36 Halaman List Komplain

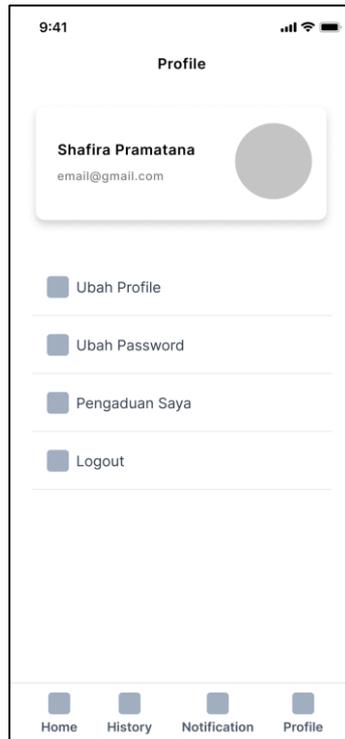
Gambar 3.38 adalah halaman detail komplain yang dapat di akses dengan mengklik salah satu list komplain. Berisi informasi komplain dan form untuk berkomunikasi dengan pihak laundry.



Gambar 3.37 Halaman Form Komplain

g. Halaman *Profile*

Halaman *profile* memuat informasi pengguna aplikasi dan beberapa menu yang berkaitan dengan kelola akun.



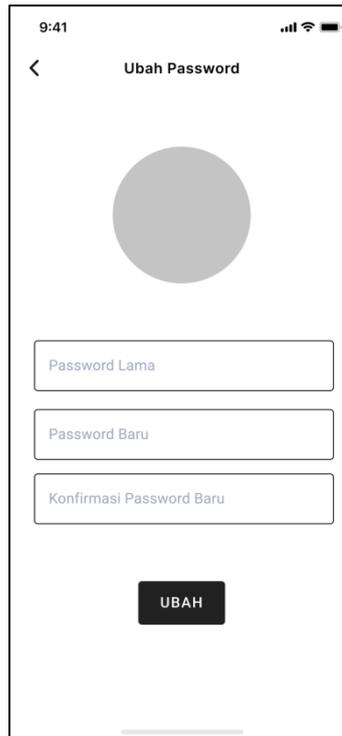
Gambar 3.38 Halaman *Profile*

Gambar 3.40 menampilkan halaman untuk mengubah profil terdiri dari input nama, alamat, email, telepon, dan gender.



Gambar 3.39 Halaman *Ubah Profile*

Gambar 3.41 adalah halaman ubah password untuk mengubah password dari akun pengguna.

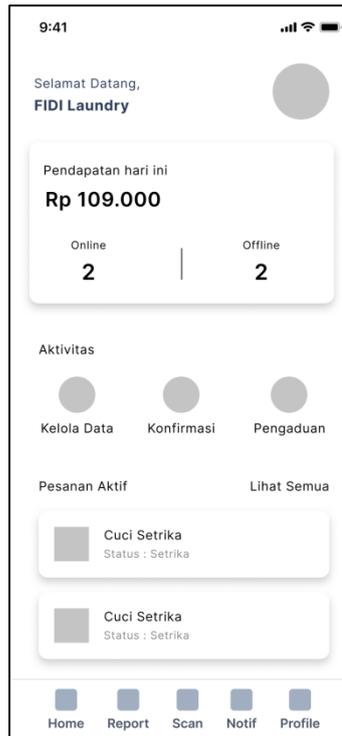


Gambar 3.40 Halaman Ubah *Password*

### 3.2.1.6.2 Wireframe Pihak Laundry

#### a. Halaman *Home*

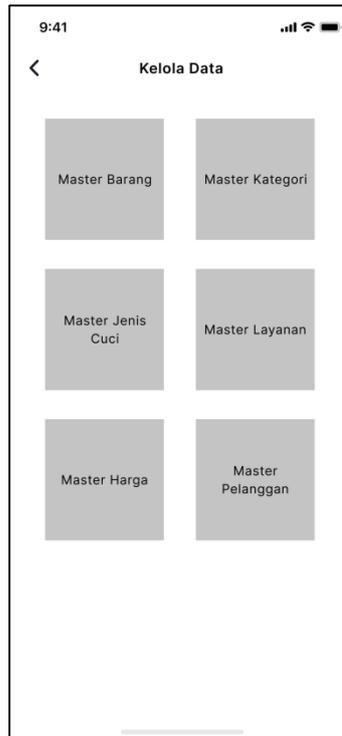
Halaman *home* memuat informasi pendapatan dan jumlah transaksi per hari, menu khusus pihak laundry, serta pesanan yang aktif.



Gambar 3.41 Halaman *Home* Laundry

b. Halaman Kelola Data

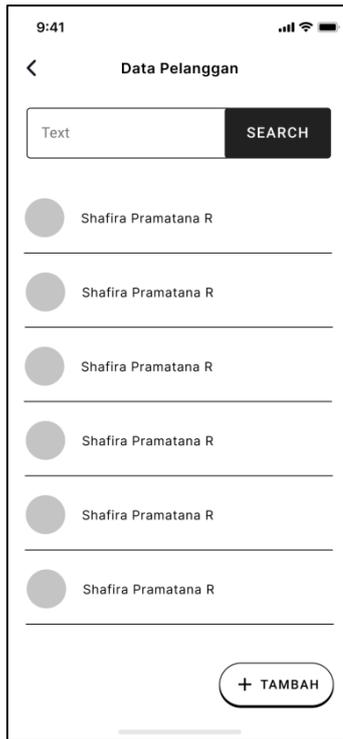
Halaman kelola data adalah halaman yang memuat menu tabel master yang dapat dilakukan CRUD oleh pihak laundry.



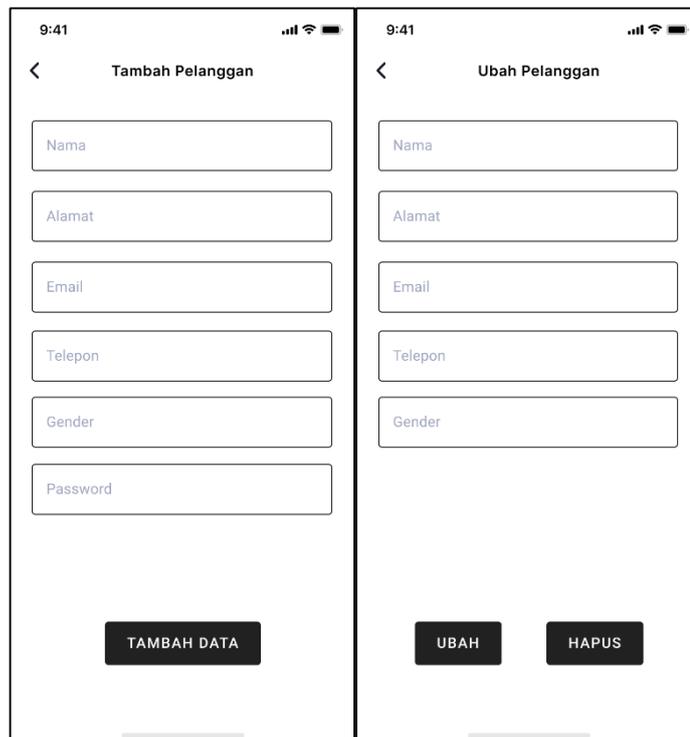
Gambar 3.42 Halaman Kelola Data Laundry

#### 1. Halaman Master Pelanggan

Gambar ini adalah halaman list pelanggan yang terdapat fitur pencarian, ubah data, dan hapus data.



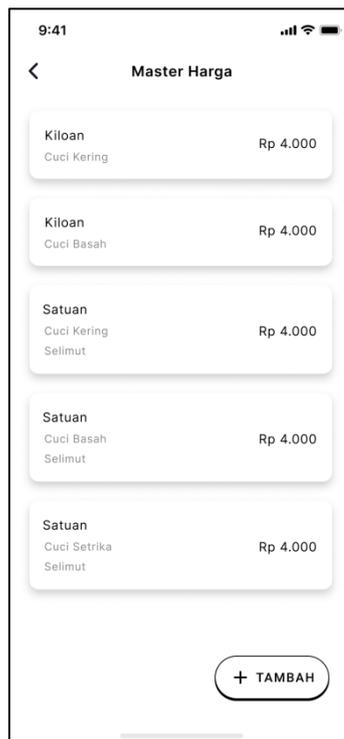
Gambar 3.43 Halaman List Pelanggan Laundry



Gambar 3.44 Halaman Tambah & Ubah Pelanggan

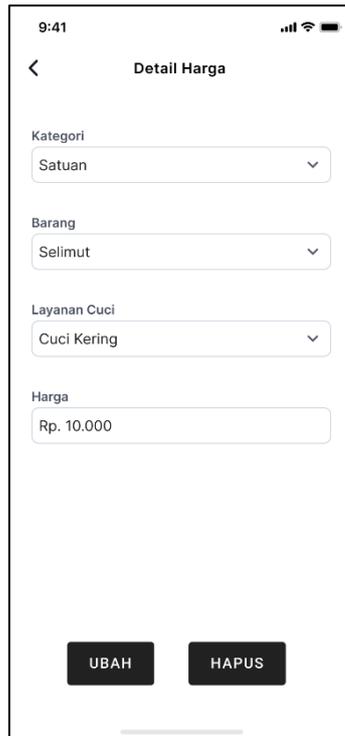
## 2. Halaman Master Items

Halaman master items memuat informasi harga dari setiap barang, kategori, dan jenis cuci karena setiap barang mempunyai harga yang berbeda untuk setiap kategori dan jenis cucinya.



Gambar 3.45 Halaman List Master Items

Gambar 3.47 adalah halaman detail harga yang dapat diubah dan dihapus oleh pihak laundry.



Gambar 3.46 Halaman Detail Harga

Gambar 3.48 adalah halaman tambah harga dimana harga mengikuti jenis-jenis yang dipilih.

9:41

< Tambah Harga

Kategori  
Satuan

Barang  
Selimut

Layanan Cuci  
Cuci Kering

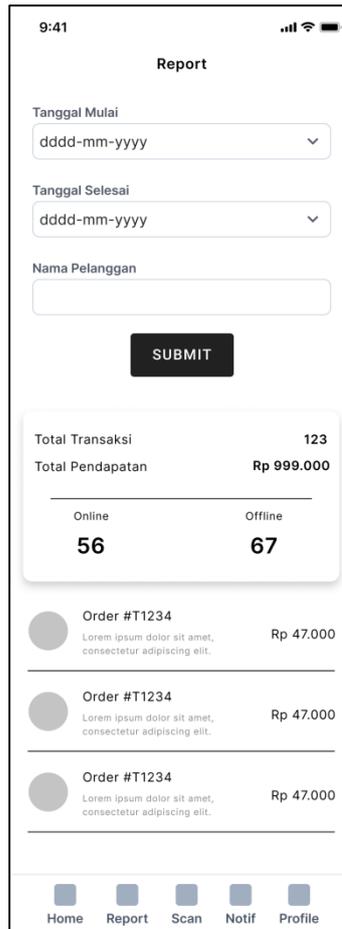
Harga  
Rp. 0

TAMBAH

Gambar 3.47 Halaman Tambah Harga

c. Halaman *Report*

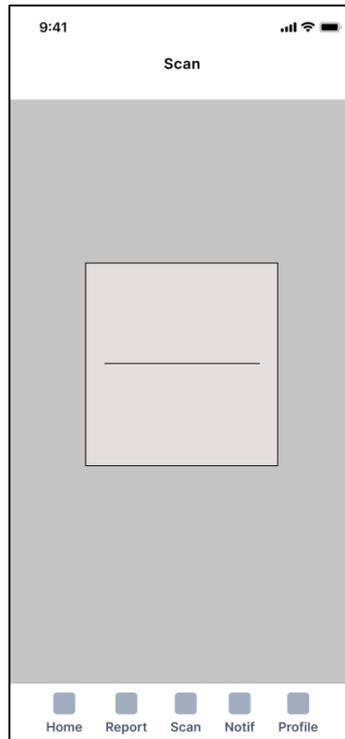
Halaman *report* memuat informasi semua transaksi yang dapat di *filter* berdasarkan nama, bulan, dan tahun.



Gambar 3.48 Halaman *Report* Laundry

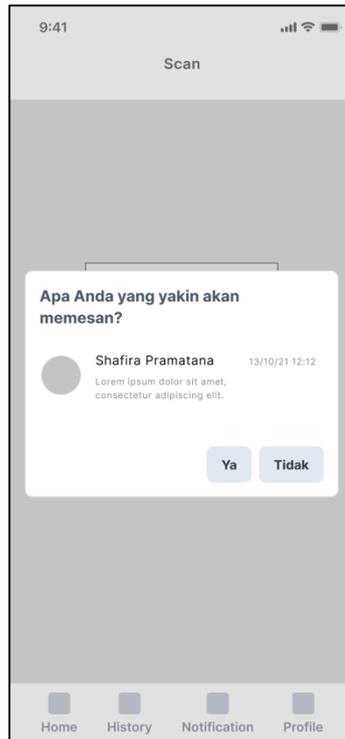
d. Halaman *Scan*

Pada halaman *scan*, pihak laundry dapat mencatat transaksi laundry yang terjadi secara *offline* agar data tetap *ter-record*.



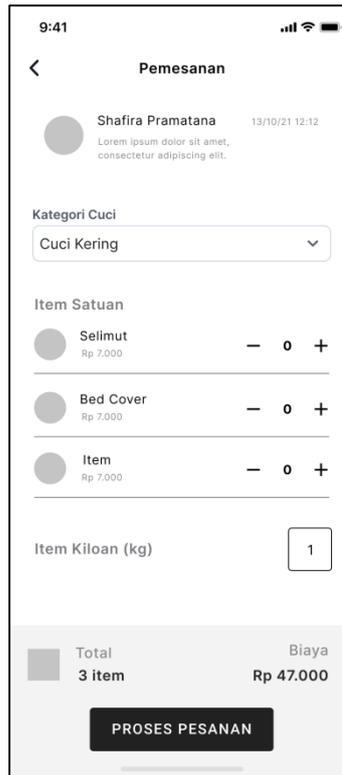
Gambar 3.49 Halaman *Scan QR Code*

Sistemnya akan mencari data yang sesuai dengan *QR code* pelanggan dan akan menampilkan pop-up konfirmasi untuk melanjutkan transaksi.



Gambar 3.50 Halaman *Pop-up* Konfirmasi *Scan*

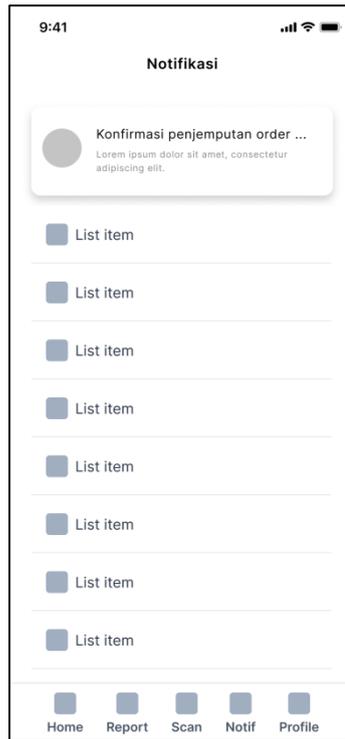
Setelah itu akan tampil halaman pemesanan dengan memasukkan kategori cuci dan pengecekan *item* barang yang termasuk ke dalam kategori satuan. Sistem akan menghitung harga sesuai dengan masukkannya.



Gambar 3.51 Halaman Update Pemesanan Laundry

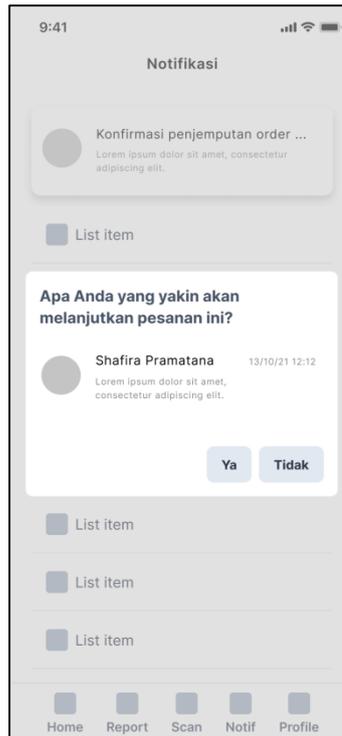
e. Halaman Notifikasi

Halaman notifikasi memuat informasi mengenai pesanan pelanggan yang melalui aplikasi.



Gambar 3.52 Halaman Notifikasi

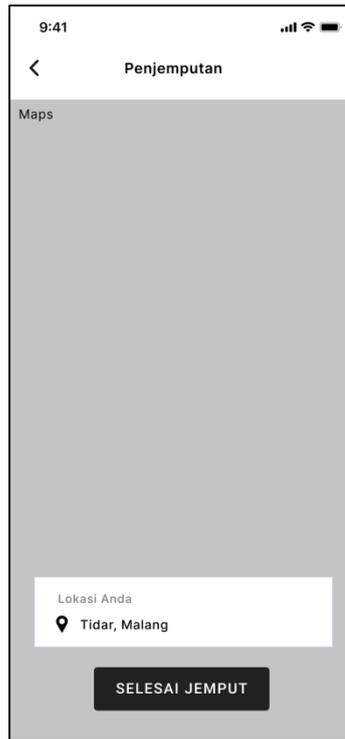
Jika notifikasi tersebut dibuka maka akan tampil pop-up konfirmasi apakah laundry menerima pesanan tersebut. (Gambar 3.54)



Gambar 3.53 Halaman *Pop-up* Konfirmasi Pemesanan

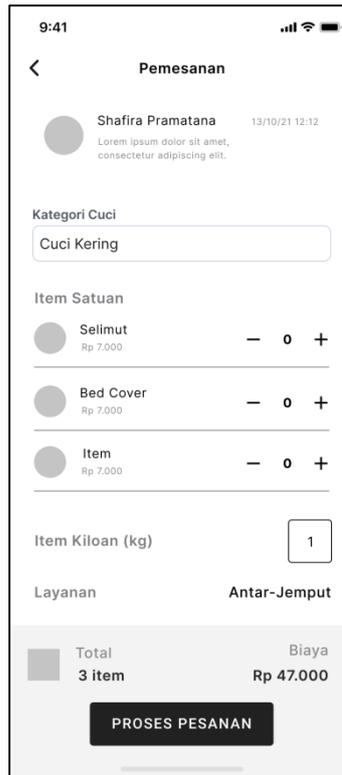
f. Halaman Maps Penjemputan

Setelah pesanan diterima maka aplikasi akan diarahkan ke halaman penjemputan. Jika petugas telah selesai menjemput laundry, maka untuk meneruskan ke halaman selanjutnya dengan mengklik *button* “Selesai Jemput”.



Gambar 3.54 Halaman *Tracking* Penjemputan

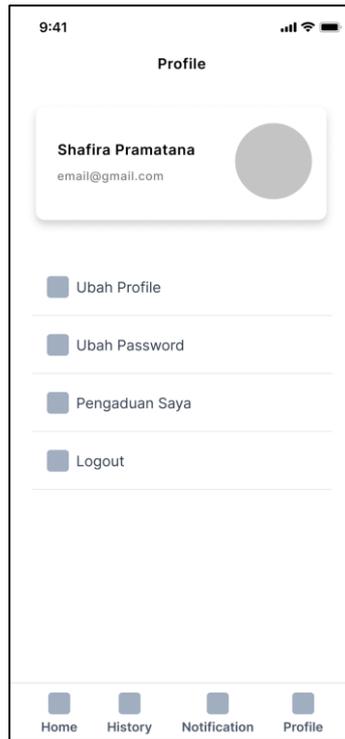
Gambar 3.56 adalah halaman *update* pesanan sama seperti step awal jika menerima pesanan secara *offline*.



Gambar 3.55 Halaman *Update* Transaksi

g. Halaman *Profile*

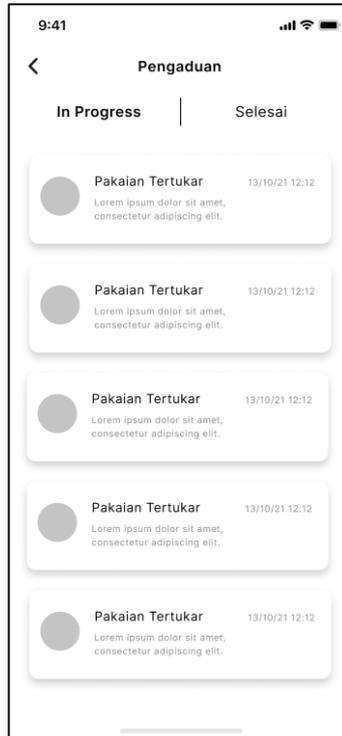
Halaman *profile* memuat informasi pengguna aplikasi dan beberapa menu yang berkaitan dengan kelola akun. Terdapat menu ubah *profile* dan ubah *password* sama seperti aplikasi pelanggan.



Gambar 3.56 Halaman *Profile* Laundry

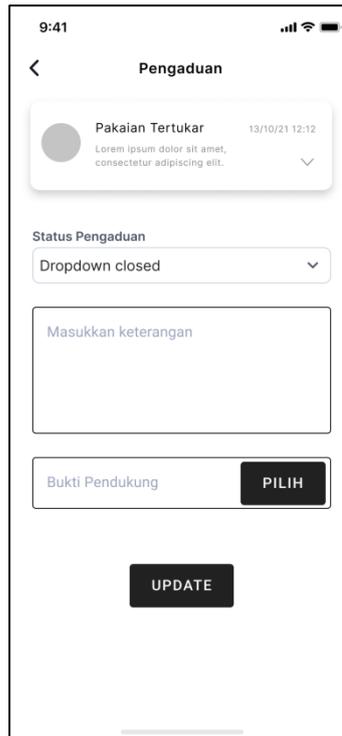
h. Halaman Komplain/Pengaduan

Halaman komplain memuat informasi komplain dari pelanggan yang dibedakan statusnya menjadi *in-progress* dan selesai. Halaman ini dapat diakses melalui *home* pihak laundry.



Gambar 3.57 Halaman *List* Komplain Laundry

Gambar 3.59 merupakan halaman detail komplain yang dapat di akses dengan mengklik salah satu list komplain. Berisi informasi komplain dan form untuk berkomunikasi dengan pihak laundry.



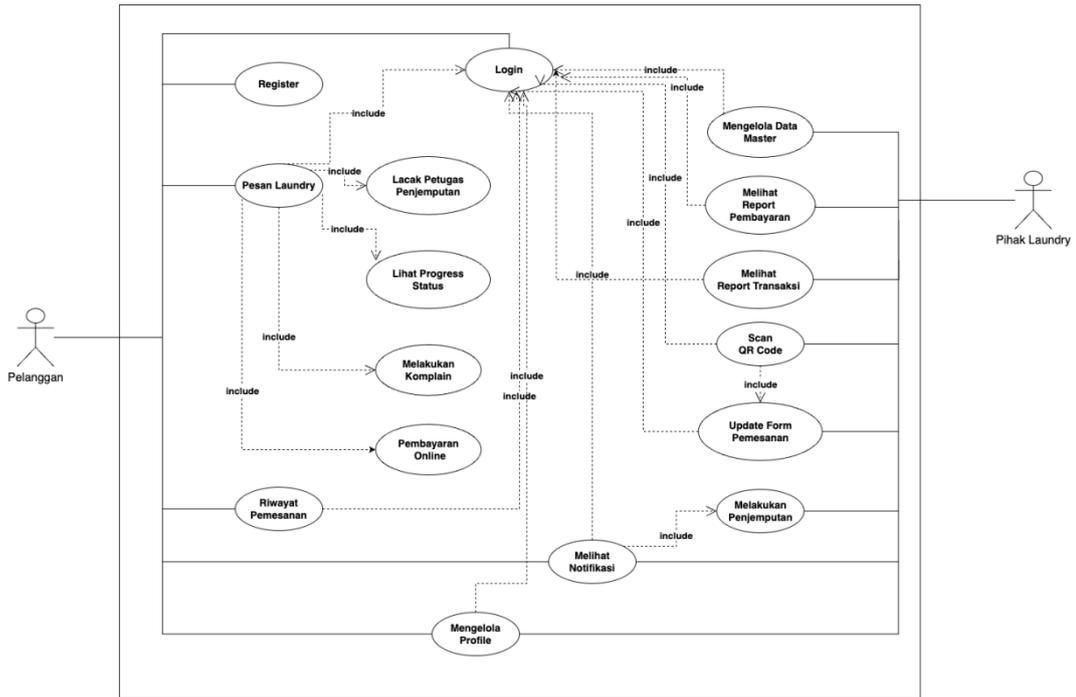
Gambar 3.58 Halaman Form Komplain Laundry

### 3.2.2 Iterasi Kedua

Pada iterasi kedua di tahap elaboration ini, merupakan hasil dari peneliti menyerahkan hasil rancangan sebelumnya ke pengguna. Peneliti menambahkan halaman report pembayaran sebagai masukkan dari pengguna, berikut rancangan yang akan di tambahkan ke sistem :

#### 3.2.2.1 Use Case Diagram

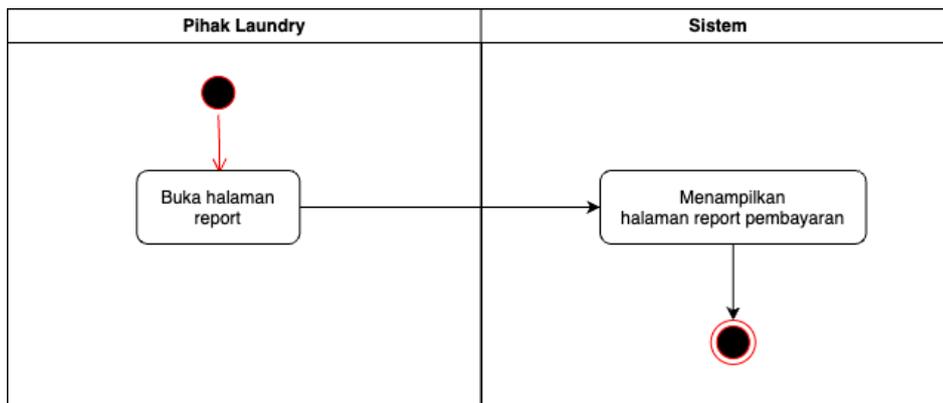
Peneliti menambahkan satu fitur baru yaitu report pembayaran pada role Pihak Laundry, dimana berfungsi untuk melihat status pembayaran pelanggan terkini.



Gambar 3.59 Use Case Diagram 2

### 3.2.2.2 Activity Diagram

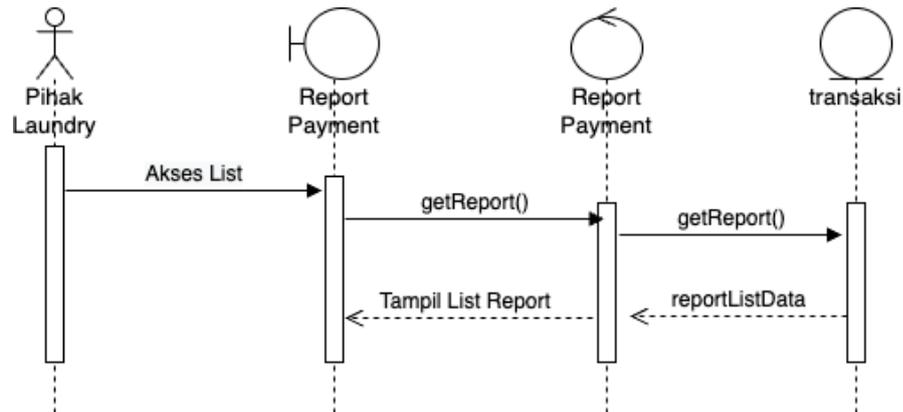
Pada gambar ini, pihak laundry dapat mengakses menu report pembayaran lalu sistem akan menampilkan list pesanan berdasarkan status dan tipe pembayarannya.



Gambar 3.60 Activity Diagram Report Pembayaran

### 3.2.2.3 Sequence Diagram

Pada sequence diagram 3.62, pihak laundry mengakses halaman *list* yang akan di proses sistem. Sistem akan meminta data pada database sehingga data tersebut diteruskan ke tampilan pihak laundry.

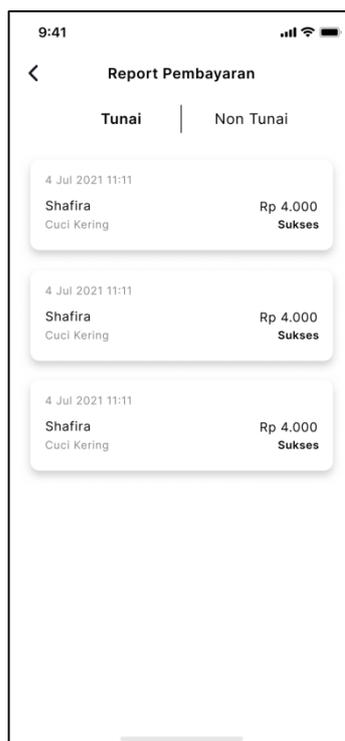


Gambar 3.61 Sequence Diagram Report Pembayaran

### 3.2.2.4 User Interface / Mock-up Aplikasi

#### a. Halaman *Report* Pembayaran

Halaman *report* pembayaran ditampilkan dalam *list* yang berisi data nama pelanggan, tanggal transaksi, jenis cuci, total pembayaran, metode dan status pembayaran dari pesanan tersebut.



Gambar 3.62 Halaman Report Pembayaran

### 3.3 Rancangan Pengujian

Pada rancangan pengujian ini, peneliti menerapkan metode *black box testing* sebagai pengujian di akhir pembuatan aplikasi. *Black box testing* adalah jenis pengujian yang fokus pada memeriksa hasil dari *input* dan *output* tanpa mengetahui kode dari aplikasi yang dibuat. Metode tersebut berfokus pada fungsionalitas dari fitur-fitur aplikasi sehingga mendapatkan hasil dari perspektif pengguna.

*Black box testing* mempunyai kelebihan dan kekurangan dalam menguji aplikasi. Salah satu kelebihannya adalah penguji dapat langsung menguji aplikasi

dan dapat mendeteksi kesalahan serta inkonsistensi sistem di awal pengujian. Sedangkan kekurangannya adalah penguji tidak dapat menguji aplikasi sepenuhnya karena tidak dapat menguji kode yang digunakan.