

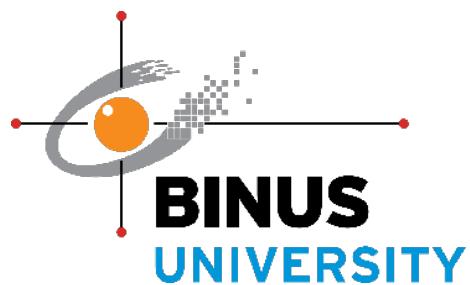
**Pengembangan Aplikasi Self-Service Berbasis Mobile Application**

**Mobile Application Based Self-Service Application Development**

**SKRIPSI**

2201835883      Fahrul Rozi Arsyandra      Computer Science    081270617007

2201836936      Annisa Octaviana      Computer Science    087782555016  
                        Nurshanty



**Computer Science Study Program  
School of Computer Science  
Universitas Bina Nusantara  
Jakarta  
2022**

**Pengembangan Aplikasi Self-Service Berbasis Mobile Application**

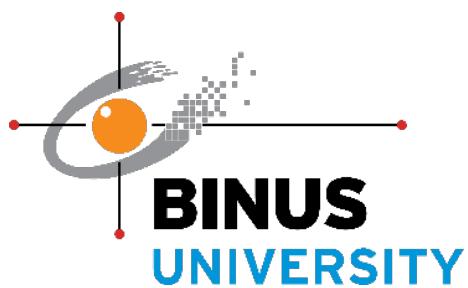
**Mobile Application Based Self-Service Application Development**

**SKRIPSI**

**Diajukan sebagai salah satu syarat  
Untuk gelar kesarjanaan pada  
Program Studi Teknik Informatika  
Jenjang Pendidikan Strata-1**

2201835883      Fahrul Rozi Arsyandra      Computer Science      081270617007

2201836936      Annisa Octaviana      Computer Science      087782555016  
                        Nurshanty



**Computer Science Study Program  
School of Computer Science  
Universitas Bina Nusantara  
Jakarta  
2022**

**Pernyataan Kesiapan Skripsi untuk Ujian Pendadaran**

**Pernyataan Penyusunan Skripsi**

**Kami,**

**Fahrul Rozi Arsyandra**

**Annisa Octaviana Nurshanty,**

dengan ini menyatakan bahwa Skripsi yang berjudul:

**Pengembangan Aplikasi Self-Service Berbasis Mobile Application**

**Mobile Application Based Self-Service Application Development**

adalah benar hasil karya kami dan belum pernah diajukan sebagai karya ilmiah, sebagian atau seluruhnya, atas nama kami atau pihak lain

**Fahrul Rozi Arsyandra**  
**2201835883**

**Annisa Octaviana**  
**Nurshanty**  
**2201836936**

**Disetujui oleh Pembimbing**  
**Saya setuju Skripsi tersebut layak diajukan untuk Ujian Pendadaran**

**Diana, S.Kom., M.T.I.**  
**D4458**

16 Juli 2022

## **PERNYATAAN**

Dengan ini saya,

Nama : Fahrul Rozi Arsyandra  
NIM : 2201835883  
Judul Skripsi : Pengembangan Aplikasi Self-Service Berbasis Mobile Application

Memberikan kepada Universitas Bina Nusantara *hak non-eksklusif* untuk menyimpan, memperbanyak, dan menyebarluaskan Skripsi karya saya, secara keseluruhan atau hanya sebagian atau hanya ringkasannya saja, dalam bentuk format tercetak dan atau elektronik.

Menyatakan bahwa saya, akan mempertahankan **hak eksklusif** saya, untuk menggunakan seluruh atau sebagian isi Skripsi saya, guna pengembangan karya di masa depan, misalnya bentuk artikel, buku, perangkat lunak, ataupun sistem informasi.

Jakarta, 12 Juli 2022

Hormat Saya,



**Fahrul Rozi Arsyandra**  
2201835883

Diketahui Oleh,

**Diana, S.Kom., M.T.I.**  
D4458

## **PERNYATAAN**

Dengan ini saya,

Nama : Annisa Octaviana Nurshanty  
NIM : 2201836936  
Judul Skripsi : Pengembangan Aplikasi Self-Service Berbasis Mobile Application

Memberikan kepada Universitas Bina Nusantara *hak non-eksklusif* untuk menyimpan, memperbanyak, dan menyebarluaskan Skripsi karya saya, secara keseluruhan atau hanya sebagian atau hanya ringkasannya saja, dalam bentuk format tercetak dan atau elektronik.

Menyatakan bahwa saya, akan mempertahankan **hak eksklusif** saya, untuk menggunakan seluruh atau sebagian isi Skripsi saya, guna pengembangan karya di masa depan, misalnya bentuk artikel, buku, perangkat lunak, ataupun sistem informasi.

Jakarta, 12 Juli 2022

Hormat Saya,

Diketahui Oleh,



Annisa Octaviana Nurshanty  
2201836936

Diana, S.Kom., M.T.I.  
D4458

## UNIVERSITAS BINA NUSANTARA

---

Computer Science Program  
Computer Science Study Program  
School of Computer Science  
Skripsi Sarjana Teknik Informatika

### Pengembangan Aplikasi Self-Service Berbasis Mobile Application

**Fahrul Rozi Arsyandra 2201835883**

**Annisa Octaviana Nurshanty 2201836936**

### ABSTRACT

*The aim of this research is to create an application that applies self-service technology to facilitate users in conducting transactions independently and has a reporting feature that can record all transactions and display them in the form of sales reports. This application is based on the Android operating system. The application development process using the SCRUM agile method starts from feature research and validation to prospective users, elaboration of application requirements, development, testing, and evaluation processes. The evaluation was carried out with two approaches, namely a questionnaire and a comparison of similar applications. It was concluded that this self-service application had met the requirements and obtained various good suggestions for future application development.*

**Keywords:** Mobile Application, Self-service, Scrum Method, UMKM

### ABSTRAK

Penelitian ini bertujuan untuk membuat aplikasi yang menerapkan *self-service technology* untuk memudahkan pengguna dalam melakukan transaksi secara mandiri dan memiliki fitur *reporting* yang dapat mencatat semua transaksi yang terjadi dan menampilkannya dalam bentuk laporan penjualan. Aplikasi ini dibuat dengan basis sistem operasi Android. Proses pengembangan aplikasi menggunakan metode *agile* yaitu *SCRUM* dimulai dari riset fitur dan validasi kepada calon pengguna, penjabaran requirement aplikasi, proses development, testing, dan evaluasi. Evaluasi dilakukan dengan dua pendekatan yaitu kuesioner dan perbandingan dengan aplikasi sejenis. Disimpulkan bahwa aplikasi self service ini sudah memenuhi requirement, dan memperoleh berbagai saran baik untuk pengembangan aplikasi di masa depan.

**Kata Kunci:** Aplikasi, Self-service, Metode Scrum, UMKM

**Keterangan:**

Computer Science Study Program : Computer Science Program

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan karunia-nya, sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul **“Pengembangan Aplikasi Self-Service Berbasis Mobile Application”**. Tujuan penyusunan skripsi ini untuk memenuhi syarat-syarat guna mencapai gelar Strata-1 (S1) Jurusan Teknik Informatika Universitas Bina Nusantara.

Tentunya penulis menyadari penulisan skripsi ini tidak lepas dari dukungan dan bantuan yang telah diberikan dari berbagai pihak. Sehingga pada kesempatan ini dengan segala kerendahan hati dan rasa hormat, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya bagi semua pihak yang telah memberikan moril maupun materil baik secara langsung maupun tidak langsung dalam penyusunan skripsi ini, terutama kepada:

1. Bapak Prof. Dr. Ir. Harjanto Prabowo, M. M. selaku Rektor Universitas Bina Nusantara.
2. Bapak Dr. Fredy Purnomo, S.Kom., M.Kom. selaku Dean of School of Computer Science Universitas Bina Nusantara.
3. Ibu Irene Aninda Iswanto, S. Kom., M.Sc.Eng. selaku Head of Computer Science Program Universitas Bina Nusantara
4. Ibu Diana, S.Kom., M.TI selaku Dosen Pembimbing yang telah berkenan memberikan ilmu dan pengetahuannya dan dengan sabar memerikan solusi pada setiap permasalahan dalam penyusunan skripsi ini
5. Seluruh dosen School of Computer Science yang telah memberikan ilmu yang bermanfaat selama penulis menjalani perkuliahan
6. Orang tua, saudara, kerabat dan teman-teman yang senantiasa memberikan doa dan dukungan selama penyusunan skripsi ini
7. Semua pihak yang tidak dapat penulis tuliskan satu persatu yang telah memberikan dukungan

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan segala bentuk kritik dan saran yang membangun dari berbagai pihak untuk pengembangan aplikasi kedepannya. Akhir kata, semoga skripsi

ini bermanfaat bagi para pembaca dan semua pihak khususnya dalam bidang ilmu Teknologi Informasi.

Jakarta, 3 Juni 2022

(Penulis)

## DAFTAR ISI

<b>PERNYATAAN .....</b>	<b>iv</b>
<b>PERNYATAAN .....</b>	<b>v</b>
<b>KATA PENGANTAR .....</b>	<b>vii</b>
<b>DAFTAR ISI .....</b>	<b>ix</b>
<b>DAFTAR GAMBAR .....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xv</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>xvi</b>
<b>BAB 1. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Ruang Lingkup.....	2
1.4 Tujuan dan Manfaat .....	2
1.5 Metode Penelitian.....	3
1.5.1 Metode Pengumpulan Data .....	3
1.5.2 Metode Perencanaan dan Pengembangan Aplikasi.....	3
1.6 Sistematika Penulisan.....	5
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1 Teori yang berkaitan dengan Mobile Application Development.....	7
2.1.1 Sistem Informasi .....	7
2.1.2 Mobile Application .....	7
2.1.3 Scrum .....	8
2.1.4 Project Management App Trello.....	18
2.1.5 Figma.....	18
2.1.6 Github.....	18
2.1.7 Gantt Chart .....	19
2.2 Teori Terkait Desain Model Aplikasi .....	20
2.2.1 UML (Unified Modelling Language).....	20
2.2.2 Data Dictionary .....	30
2.3 Teori yang Berkaitan dengan Pengembangan Mobile Application .....	31
2.3.1 Dart.....	31
2.3.2 Flutter .....	32
2.3.3 HTTP Request Methods.....	32
2.3.4 Firebase .....	33
2.3.5 Laravel.....	33
2.3.6 Database Management System (DBMS) .....	33
2.3.7 Black Box Testing.....	37
2.4 Teori berkaitan dengan evaluasi.....	37
2.4.1 Eight Golden Rules of Interface Design .....	37
2.4.2 Lima Faktor manusia terukur .....	39

<b>BAB 3. METODE PENELITIAN .....</b>	<b>41</b>
3.1 Kerangka Pikir .....	41
3.1.1 Membuat Product Backlog.....	41
3.1.2 Membuat Sprint Planing .....	42
3.1.3 Daily Scrum .....	48
3.1.4 Sprint Review dan Sprint Retrospective .....	59
3.1.5 Pembagian Tugas .....	61
3.1.6 Gantt Chart.....	61
3.2 Analisis.....	62
3.2.1 Analisis Aplikasi Sejenis.....	62
3.2.2 Analisis Kebutuhan User.....	65
3.2.3 Usulan Pemecahan Masalah.....	77
3.3 Perancangan Aplikasi.....	78
3.3.1 Software Design Documentation .....	78
3.3.2 Perancangan Sistem .....	81
3.3.3 Perancangan Database.....	115
3.3.4 Perancangan <i>User Interface (UI)</i> .....	124
<b>BAB 4. HASIL DAN PEMBAHASAN .....</b>	<b>142</b>
4.1 Spesifikasi <i>System</i> .....	142
4.1.1 Spesifikasi Perangkat Keras.....	142
4.1.2 Spesifikasi Perangkat Lunak .....	142
4.2 Prosedur Penggunaan Aplikasi .....	142
4.2.1 Prosedur Untuk Customer.....	143
4.2.2 Prosedur Untuk Merchant .....	155
4.2.3 Prosedur Untuk Admin .....	167
4.3 Evaluasi .....	169
4.3.1 Evaluasi User Interface .....	169
4.3.2 Evaluasi Sistem .....	178
4.3.3 Evaluasi User aplikasi “ComanderApp” Berdasarkan Kuesioner .....	181
4.3.4 Evaluasi Aplikasi sejenis “ComanderApp” .....	187
<b>BAB 5. SIMPULAN DAN SARAN .....</b>	<b>189</b>
5.1 Simpulan .....	189
5.2 Saran.....	189
<b>REFERENSI .....</b>	<b>190</b>
<b>LAMPIRAN .....</b>	<b>195</b>
<b>RIWAYAT HIDUP.....</b>	<b>209</b>

## DAFTAR GAMBAR

Gambar 2.1 Contoh Gantt Chart.....	20
Gambar 2.2 Contoh Usecase Description .....	22
Gambar 2.3 Contoh Class.....	23
Gambar 2.4 Contoh Association .....	23
Gambar 2.5 Contoh Sequence Diagram .....	24
Gambar 2.6 Actor pada Sqequence Diagram.....	25
Gambar 2.7 Object.....	25
Gambar 2.8 Lifeline.....	25
Gambar 2.9 Excecution Occurrence .....	26
Gambar 2.10 Message .....	26
Gambar 2.11 Contoh Activity Diagram .....	27
Gambar 2.12 Activity dan Action .....	28
Gambar 2.13 Object node.....	28
Gambar 2.14 Control Flow.....	28
Gambar 2.15 Object Flow .....	28
Gambar 2.16 Initial Node.....	29
Gambar 2.17 Final-activity Node .....	29
Gambar 2.18 Decision Node .....	29
Gambar 2.19 Merge Node .....	29
Gambar 2.20 Fork Node.....	30
Gambar 2.21 Join Node .....	30
Gambar 2.22 Swimlane.....	30
Gambar 2.23 Contoh Data Dictionary.....	31
Gambar 2.24 Contoh Entity .....	35
Gambar 2.25 Contoh Relationship Type .....	36
Gambar 3.1 Gantt Chart .....	62
Gambar 3.2 Aplikasi Bakmi GM .....	63
Gambar 3.3 Aplikasi Kopi Kenangan .....	64
Gambar 3.4 Diagram Pertanyaan 1.....	67
Gambar 3.5 Diagram Pertanyaan 2.....	68
Gambar 3.6 Diagram Pertanyaan 3.....	68
Gambar 3.7 Diagram Pertanyaan 4.....	69
Gambar 3.8 Diagram Pertanyaan 5.....	69
Gambar 3.9 Diagram Pertanyaam 6 .....	70
Gambar 3.10 Diagram Pertanyaan 7.....	70
Gambar 3.11 Diagram Pertanyaan 8.....	71

Gambar 3.12 Diagram Pertanyaan 9.....	71
Gambar 3.13 Diagram Pertanyaan 10.....	72
Gambar 3.14 Diagram Pertanyaan 11.....	73
Gambar 3.15 Diagram Pertanyaan 12.....	73
Gambar 3.16 Use Case Diagram .....	81
Gambar 3.17 Activity Diagram melakukan sign up.....	97
Gambar 3.18 Activity Diagram Melakukan Login.....	98
Gambar 3.19 Activity Diagram Melihat Status Pesanan .....	99
Gambar 3.20 Activity Diagram Melakukan Isi Ulang Saldo.....	99
Gambar 3.21 Activity Diagram Memilih Merchant.....	100
Gambar 3.22 Activity Diagram Memesan Barang.....	100
Gambar 3.23 Activity Diagram Melakukan Logout .....	101
Gambar 3.24 Memperbarui Data Barang .....	102
Gambar 3.25 Activity Diagram Melakukan Update Profil.....	103
Gambar 3.26 Activity Diagram Memproses Pesanan .....	104
Gambar 3.27 Activity Diagram Melihat Data Penjualan .....	104
Gambar 3.28 Activity Diagram Input Barang Penjualan .....	105
Gambar 3.29 Class Diagram Aplikasi ComanderApp .....	106
Gambar 3.30 Sequence Diagram Sign Up .....	107
Gambar 3.31 Sequence Diagram Login.....	108
Gambar 3.32 Sequence Diagram Memilih Merchant .....	108
Gambar 3.33 Sequence Diagram Memesan Barang .....	109
Gambar 3.34 Sequence Diagram Melakukan Pembayaran .....	109
Gambar 3.35 Sequence Diagram Melihat Status Pemesanan .....	110
Gambar 3.36 Sequence Diagram Melakukan Isi Ulang Saldo .....	110
Gambar 3.37 Sequence Diagram Memperbarui Data Barang .....	111
Gambar 3.38 Sequence Diagram Input Data Barang Penjualan .....	111
Gambar 3.39 Sequence Diagram Mengubah Profil .....	112
Gambar 3.40 Sequence Diagram Memproses Pesanan.....	112
Gambar 3.41 Sequence Diagram Melihat Data Penjualan.....	113
Gambar 3.42 Sequence Diagram Hapus Pembeli atau Merchant .....	113
Gambar 3.43 Sequence Diagram Logout .....	114
Gambar 3.44 Sequence Diagram Hapus Produk.....	114
Gambar 3.45 ERD Aplikasi Comander App .....	115
Gambar 3.46 Tabel User .....	116
Gambar 3.47 Tabel Account .....	117
Gambar 3.48 Tabel Product.....	118
Gambar 3.49 Tabel ProductType .....	119

Gambar 3.50 Tabel Transaction.....	120
Gambar 3.51 Tabel TransactionDetail .....	121
Gambar 3.52 Tabel Merchant.....	122
Gambar 3.53 User Interface Register Customer.....	124
Gambar 3.54 User Interface Login .....	125
Gambar 3.55 User Interface Homepage .....	126
Gambar 3.56 User Interface halaman Scan Barcode .....	127
Gambar 3.57 User Interface halaman Transaksi.....	128
Gambar 3.58 User Interface Halaman Menu.....	129
Gambar 3.59 User Interface Halaman Cart.....	130
Gambar 3.60 User Interface Halaman Topup .....	131
Gambar 3.61 User Interface Halaman Register Merchant .....	133
Gambar 3.62 User Interface Halaman Login.....	134
Gambar 3.63 User Interface Halaman Home.....	135
Gambar 3.64 User Interface Halaman Product.....	136
Gambar 3.65 User Interface Form Add Product .....	137
Gambar 3.66 User Interface Form Edit Product .....	138
Gambar 3.67 User Interface Halaman Transaksi .....	139
Gambar 3.68 User Interface Halaman Detail Transaksi .....	140
Gambar 3.69 User Interface Halaman Profil.....	141
Gambar 4.1 Screenshot Halaman Login.....	143
Gambar 4.2 Screenshot Halaman Sign Up .....	144
Gambar 4.3 Screenshoot Halaman Home Customer.....	145
Gambar 4.4 Screenshoot Halaman Scan Barcode.....	146
Gambar 4.5 Screenshoot Halaman Daftar Menu Merchant.....	147
Gambar 4.6 Screenshoot Bottom Sheet Quantity Produk.....	148
Gambar 4.7 Screenshoot Halaman Cart .....	149
Gambar 4.8 Screenshoot Bottom Sheet Payment Method .....	150
Gambar 4.9 Screenshoot Halaman Top up Saldo .....	151
Gambar 4.10 Bottom Sheet Validasi Top up Saldo .....	152
Gambar 4.11 Screenshoot Halaman Profile Customer .....	153
Gambar 4.12 Screenshoot Halaman Edit Profile Customer.....	154
Gambar 4.13 Screenshoot Halaman Login.....	155
Gambar 4.14 Screenshoot Halaman Register .....	156
Gambar 4.15 Screenshoot Halaman Merchant Dashboard .....	157
Gambar 4.16 Screenshoot Halaman Summary.....	158
Gambar 4.17 Screenshoot Halaman Transaksi Merchant .....	159
Gambar 4.18 Screenshoot Halaman Detail transaksi .....	160

Gambar 4.19 Screenshot Bottom Sheet Update Status Transaksi.....	161
Gambar 4.20 Screenshot Halaman Product Merchant.....	162
Gambar 4.21 Screenshot Halaman Add Product Merchant .....	163
Gambar 4.22 Screenshot Halaman Edit Produk Merchant.....	164
Gambar 4.23 Screenshot Halaman Profile Merchant.....	165
Gambar 4.24 Screenshot Halaman Edit Profile Merchant.....	166
Gambar 4.25 Screenshot Halaman Login Admin .....	167
Gambar 4.26 Screenshot Halaman Home.....	167
Gambar 4.27 Screenshot Halaman List User .....	168
Gambar 4.28. Screenshot modal delete .....	168
Gambar 4.29. Contoh penerapan Strive for Consistency .....	169
Gambar 4.30. Contoh penerapan Strive for Consistency .....	170
Gambar 4.31. Contoh penerapan informative feedback.....	171
Gambar 4.32. Contoh pesan pop up.....	172
Gambar 4.33. Contoh penerapan prevent error.....	173
Gambar 4.34. Contoh penerapan Strive for Consistency .....	174
Gambar 4.35. Contoh keep user control .....	175
Gambar 4.36. Contoh penerapan short term memory.....	176
Gambar 4.37 Diagram Pertanyaan 1.....	182
Gambar 4.38 Diagram Pertanyaan 2.....	183
Gambar 4.39 Diagram Pertanyaan 3.....	183
Gambar 4.40 Diagram Pertanyaan 4.....	184
Gambar 4.41 Diagram Pertanyaan 5 .....	184
Gambar 4.42 Diagram Pertanyaan 6.....	185
Gambar 4.43 Diagram Pertanyaan 7.....	185
Gambar 4.44 Diagram Pertanyaan 8.....	186
Gambar 4.45 Diagram Pertanyaan 9.....	186

## DAFTAR TABEL

Tabel 3.1 Product Backlog .....	41
Tabel 3.2 Sprint Planning .....	43
Tabel 3.3 Daily Scrum .....	48
Tabel 3.4 Sprint Review dan Sprint Retrospective.....	60
Tabel 3.5 Perbandingan Aplikasi Sejenis.....	65
Tabel 3.6 Pertanyaan dan Alasan Pertanyaan Kuesioner .....	65
Tabel 3.7 Pertanyaan dan Alasan Pertanyaan Wawancara .....	74
Table 3.8 Use Case Description Login .....	82
Table 3.9 Use Case Description Keluar .....	83
Table 3.10 Use Case Description Pendaftaran Pengguna .....	83
Table 3.11 Use Case Daftar sebagai merchant .....	84
Table 3.12 Use Case Tambah Barang.....	86
Tabel 3.13 Use Case Description Update Barang Merchant.....	86
Tabel 3.14 Use Case Description Update Profil .....	87
Table 3.15 Use Case Description Merchant Memperoses Pesanan .....	89
Table 3.16 Use Case Description Melihat Riwayat Transaksi.....	91
Table 3.17 Use Case Description Melihat Data Penjualan.....	92
Table 3.18 Use Case Description Pembeli memesan pesanan .....	93
Table 3.19 Use Case Description Rating .....	94
Table 3.20 Use Case Description Tinjau Merchant .....	95
Tabel 3.21 Data Dictionary User .....	116
Tabel 3.22 Data Dictionary Account .....	118
Tabel 3.23 Data Dictionary Product.....	119
Tabel 3.24 Data Dictionary ProductType .....	120
Tabel 3.25 Data Dictionary Transaction.....	120
Tabel 3.26 Data Dictionary TransactionDetail .....	121
Tabel 3.27 Data Dictionary Merchant.....	123
Table 4.1 Testing Aplikasi ‘ComanderApp’ .....	178
Tabel 4.2 Pertanyaan dan Alasan Pertanyaan Kuesioner .....	181
Tabel 4.3 Evaluasi Aplikasi Sejenis .....	187

## **DAFTAR LAMPIRAN**

Lampiran 1 Kuisioner analisis aplikasi.....	195
Lampiran 2 Kuisioner Evaluasi Aplikasi .....	201
Lampiran 3 Surat Survei.....	206
Lampiran 4 Surat Pernyataan Telah Melakukan Survei .....	207

## BAB

1.

### PENDAHULUAN

#### 1.1 Latar Belakang

Berbagai inovasi digital pada berbagai bidang membuktikan bahwa masyarakat juga turut andil dalam perkembangan zaman yang semakin *modern*. Dengan munculnya uang elektronik akan memungkinkan masyarakat untuk melakukan transaksi finansial tanpa menggunakan uang tunai. Transaksi non tunai juga didukung dengan berubahnya pola hidup masyarakat modern saat ini Penggunaan uang elektronik sebagai alternatif alat pembayaran non-cash didukung dengan gaya hidup masyarakat yang semakin mobile, gaya hidup mobile transaction.

UMKM di Indonesia telah menjadi pilar terpenting bagi ekosistem ekonomi. Diketahui bahwa 99% pelaku usaha di Indonesia adalah sektor UMKM. Peran UMKM ini telah berkontribusi 60% terhadap produk domestik bruto nasional dan 97% terhadap penyerapan tenaga kerja yang terdampak pandemi (Arianto, B. 2020). Namun Pandemi Covid-19 telah membuat menurunnya daya beli masyarakat. Dikarenakan publik telah mengurangi interaksi diluar ruangan untuk menekan persebaran pandemi. Dengan demikian, banyak konsumen yang kemudian menjaga jarak dan mengalihkan pembelian secara digital. Sedangkan dari UMKM yang ada saat ini ternyata baru 16 persen yang telah masuk dalam ekosistem ekonomi digital. Dalam konteks Indonesia, sektor UMKM merupakan salah satu pilar utama dari fundamental ekonomi Indonesia.

Memberikan inovasi dan system layanan serba cepat dan mandiri yang sesuai dengan kebutuhan merupakan keinginan yang selalu menjadi tuntutan pelanggan sehingga menjadi salah satu kunci sukses bersaing dibanyak bisnis. *Self Service Technology* adalah contoh dari pelayanan konsumen yang dapat dioptimalkan sehingga memberikan pelanggan lebih banyak kontrol di tangan mereka sendiri disbanding sebelumnya. Dimasa pandemic ini, dimana kontak fisik antara manusia sangat dibatasi, self service merupakan salah satu pilihan para UMKM.

Oleh karena itu, penulis akan mengembangkan sebuah aplikasi yang dapat menerapkan *Self Service Technology*. Pengguna dapat menggunakan aplikasi tersebut untuk melihat menu dan dapat langsung menggunakan transaksi pemesanan. Apabila

berhasil dapat langsung mengambil di toko. Dari segi penjual, aplikasi ini dapat mencatat semua transaksi dan dapat memberikan hasil analisis penjualanan.

### 1.2 Rumusan Masalah

1. Bagaimana aplikasi dapat meminimalisir kerumunan demi mengoptimalkan social dan physical distancing dimasa pandemic?
2. Bagaimana membuat sebuah aplikasi yang dapat membantu pengguna melakukan transaksi secara mandiri?
3. Bagaimana aplikasi dapat membantu UMKM tersebut dalam menyelesaikan masalah *reporting*?

### 1.3 Ruang Lingkup

1. Aplikasi ini akan di buat berbasis *mobile application dan web application*
2. Aplikasi ini akan di buat menggunakan android dan Bahasa pemrograman PHP dengan framework Laravel dan Bootstrap dan bahasa pemrograman dart dengan framework Flutter.
3. Software yang digunakan dalam pembuatan aplikasi ini adalah Visual Studio Code dan database MySql.
4. User akan dibagi menjadi 2 yaitu *customer* dan *merchant*

### 1.4 Tujuan dan Manfaat

#### Tujuan

1. Mengembangkan aplikasi yang dapat membantu mengurangi kerumunan yang disebabkan oleh antrian.
2. Membuat aplikasi yang dapat membantu pengguna untuk membuat pesanan secara mandiri tanpa harus mengantri.
3. Membuat aplikasi yang dapat membantu UMKM dalam melakukan pembukuan.

#### Manfaat

1. Aplikasi dapat mengurangi kerumunan pada saat mengantri
2. Aplikasi menjadi sarana untuk pengguna dalam membeli barang secara mandiri
3. UMKM dapat menganalisis hasil transaksi dan penjualan

## 1.5 Metode Penelitian

### 1.5.1 Metode Pengumpulan Data

#### 1. Studi Pustaka

Metode ini merupakan metode pengumpulan data dengan cara mempelajari dan membaca literatur-literatur yang berkaitan dengan topik skripsi ini. Sumber literatur yang bisa dijadikan sebagai landasan teori untuk pengembangan aplikasi seperti karya ilmiah, jurnal, e-book dll.

#### 2. Observasi

Metode ini merupakan metode pengumpulan data dengan cara melakukan pengamatan secara langsung untuk mendapatkan informasi yang dibutuhkan. Observasi ini akan dilakukan dengan cara mengamati aplikasi yang memiliki fitur serupa. Penulis akan melakukan observasi beberapa aplikasi seperti Kopi Kenangan, Bakmi GM, dan aplikasi self service pada restoran McDonald.

#### 3. Kuesioner

Metode ini dilakukan dengan melakukan penyebaran Kuesioner yang berisi pertanyaan atau pernyataan tertulis kepada sejumlah responden terkait untuk mendapatkan sejumlah data dan informasi yang relevan dengan pengembangan aplikasi.

## 4. Wawancara

Metode ini dilakukan untuk mendapatkan informasi yang tepat dari narasumber dalam bentuk komunikasi antara dua orang atau lebih. Wawancara akan dilakukan kepada calon pengguna aplikasi untuk memperoleh informasi yang berkaitan dengan permasalahan yang dihadapi calon pengguna.

### 1.5.2 Metode Perencanaan dan Pengembangan Aplikasi

Metode dan perancangan yang digunakan dalam pengembangan aplikasi ini adalah metode *Agile Scrum*. Menurut Faisal Hayat dalam jurnalnya

yang berjudul “The Influence of Agile Methodology (Scrum) on Software Project Management” *Scrum* digunakan untuk mengelola proses pengembangan dengan menerapkan konsep kemampuan beradaptasi, kemahiran dan kelenturan dari hipotesis pengendalian proses organisasi. Scrum berkonsentrasi pada kerja tim dan kualitas produk di serbaguna lingkungan. Artinya dengan menggunakan metode scrum memberikan cara pengembangan software secara bertahap dan berkala. Dengan metode ini pengerjaan aplikasi yang sebelumnya kompleks menjadi lebih mudah karena pengerjaan aplikasi dibagi menjadi bagian-bagian kecil yang melewati beberapa proses.

#### 1. *Product Backlog*

Tahapan scrum dimulai dari membuat produk backlog yaitu tahapan untuk mengumpulkan fungsionalitas yang diinginkan dari sebuah produk yang berpotensi untuk dikembangkan lebih lanjut. Setelah tujuan diketahui selanjutnya adalah membaginya menjadi bagian-bagian kecil untuk menentukan skala prioritas dan menghasilkan todo list untuk dikerjakan. Setelah to-do list dibuat tahapan selanjutnya adalah mem-breakdown kebutuhan apa saja yang diperlukan dan membuat sprint planning.

#### 2. *Sprint Planing*

Pada tahap *sprint planning*, projek teratas dalam *product backlog* akan disusun kembali menjadi *sprint backlog* dan selanjutnya menentukan bagaimana cara untuk menyelesaikan apa yang ada dalam *sprint backlog*.

#### 3. *Daily Scrum*

Evaluasi dilakukan pada tahap ini dimana setiap anggota masing-masing menyampaikan *update* tiap perkerjaan yang dilakukan setiap hari juga menyampaikan kendala apa saja yang dihadapi selama mengerjakan tugas nya tersebut. *Daily scrum* akan dilakukan via rapat secara daring menggunakan *google meet* atau via chat menggunakan grup whatsapp.

#### 4. *Sprint Review*

Pada tahap *sprint review* para anggota akan mendemonstrasikan apa saja yang sudah diselesaikan dalam periode satu *sprint* atau setiap satu *sprint* selesai.

## 5. *Sprint Retrospective*

*Sprint Restrospective* dilakukan setiap *sprint* berakhir dimana semua anggota bisa menyampaikan pendapat dan evaluasi mengenai kinerja selama menjalani *scrum*.

## 1.6 Sistematika Penulisan

Sistematika penulisan dalam pembuatan skripsi ini dibagi menjadi 5 bagian yaitu:

- **BAB 1: PENDAHULUAN**

Pada bagian ini menjelaskan mengenai latar belakang penulisan, rumusan masalah, ruang lingkup, tujuan dan manfaat, metode penelitian, dan sistematika penulisan yang digunakan.

- **BAB 2: TINJAUAN REFERENSI**

Pada bagian ini berisi teori-teori dan konsep yang digunakan dalam pembuatan skripsi dan aplikasi ini. Teori-teori tersebut didapatkan dari hasil studi pustaka, buku jurnal, e-book dan sumber lain yang dapat mendukung pembuatan skripsi ini.

- **BAB 3: METODE PENELITIAN**

Pada bagian ini menjelaskan metode pelaksanaan, analisis, dan perancangan sistem yang digunakan dalam pembuatan aplikasi.

- **BAB 4: HASIL DAN PEMBAHASAN**

Pada bagian ini menjelaskan hasil, pembahasan dan evaluasi dalam pembuatan aplikasi.

- **BAB 5: SIMPULAN DAN SARAN**

Pada bagian ini merupakan bagian terakhir dalam penulisan skripsi ini, bagian ini berisi kesimpulan dan saran yang diperoleh setelah pembuatan aplikasi ini agar aplikasi ini lebih baik lagi kedepannya.



## **BAB 2.**

### **TINJAUAN PUSTAKA**

#### 2.1 Teori yang berkaitan dengan Mobile Application Development

##### 2.1.1 Sistem Informasi

Menurut Tata Sutabri dalam bukunya yang berjudul “Analisis Sistem Informasi” mengemukakan bahwa Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variable yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu.

Definisi sistem Menurut Dr. Ir. Harijono Djojodihardjo (1984: 78) dalam Jeperson Hutahaean (2015: 2) “Suatu sistem adalah sekumpulan objek yang mencakup hubungan fungsional antar tiap-tiap objek dan hubungan antara ciri tiap objek, dan yang secara keseluruhan merupakan suatu kesatuan secara fungsional.”

Menurut Gordon B. Davis (2015: 9) mengemukakan bahwa informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang.

Menurut Jeperson hutahaean (2015: 9) “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Berdasarkan definisi yang telah dinayatakan oleh para ahli diatas, yang menimpulkan bahwa sistem informasi merupakan kumpulan dari berbagai unsur atau komponen yang saling berinteraksi, terorganisasi dan terhubung satu sama lain yang memiliki tujuan untuk mengumpulkan, mengolah, dan menganalisa suatu sumber untuk dijadikan suatu bentuk yang dapat bermanfaat untuk penerima.

##### 2.1.2 Mobile Application

*Mobile Application* dapat ditujukan untuk aplikasi perangkat lunak yang memungkinkan pengguna untuk melakukan tugas yang dapat diinstal dan

dijalankan pada platform pengembangan tertentu contohnya App Store dan Google Play. (F. Lin & J. Zhao, 2022)

#### 2.1.2.1 Android

Javapoint.com yang merupakan situs resmi yang membahas materi-materi terkait bahasa pemrograman menjelaskan bahwa android merupakan seperangkat perangkat lunak dan sistem operasi berbasis linux yang didesain untuk perangkat seluler. Android dikembangkan pertama kali oleh Android, Inc., sebelum diakusisi oleh Google pada tahun 2005. Android merupakan sistem operasi yang open source atau sumber yang terbuka. Android memiliki komunitas pengembang aplikasi yang cukup besar. Komunitas tersebut memberikan berbagai macam peningkatan fungsionalitas pada untuk perangkat. Faktor-faktor tersebut yang membuat android menjadi salah satu sistem operasi *smartphone* yang paling banyak digunakan.

#### 2.1.3 Scrum

*Scrum* adalah kerangka kerja sederhana yang membantu orang secara individu, tim, dan organisasi untuk menghasilkan nilai melalui solusi adaptif untuk masalah yang kompleks. *Scrum* tidak dibangun dengan memberikan instruksi secara terperinci dan hanya mendefinisikan bagian-bagian yang diperlukan untuk menerapkan teori *scrum*, dan berdasarkan kecerdasan kolektif orang-orang yang menggunakannya. *Scrum* menunjukkan keefektifan relatif dari manajemen, lingkungan, dan teknik kerja saat ini, sehingga perbaikan dapat dilakukan (Schwaber & Sutherland, 2020, p. 3).

*Scrum* didasarkan pada *empirism* dan *lean thinking* yaitu pengalaman, membuat keputusan berdasarkan apa yang diamati, dan fokus pada hal-hal penting. *Scrum* juga akan berlangsung dengan baik apabila menerapkan pilar-pilar *Scrum*, yaitu sebagai berikut (Schwaber & Sutherland, 2020, pp. 3-4):

- Transparansi

Proses pekerjaan dan hasil dari sebuah pekerjaan harus dapat dilihat oleh semua pihak, yaitu orang yang mengerjakan dan orang yang akan menerima hasil pekerjaan. Transparansi

memungkinkan inspeksi. Inspeksi tanpa transparansi akan menyesatkan dan boros.

- **Inspeksi**

Selama kegiatan *Scrum* berlangsung, harus terdapat inspeksi berkala untuk mendeteksi masalah yang tidak diinginkan. Inspeksi memungkinkan adaptasi. Inspeksi tanpa adaptasi dianggap sia-sia.

- **Adaptasi**

Merupakan sebuah proses penyesuaian proses kerja atau hasil kerja bila terdapat aspek proses yang menyimpang di luar batas yang dapat diterima atau jika produk yang dihasilkan tidak dapat diterima, proses yang diterapkan harus disesuaikan. Penyesuaian harus dilakukan sesegera mungkin untuk meminimalkan penyimpangan lebih lanjut dan harus dilakukan oleh tim *scrum* agar tidak terjadi penyimpangan yang terlalu jauh.

#### 2.1.3.1 Scrum Values

Keberhasilan penggunaan *Scrum* tergantung pada kemampuan seseorang yang menjadi lebih mahir dalam menjalankan lima nilai *Scrum*, yaitu *Commitment, Focus, Openness, Respect, and Courage*. Nilai-nilai ini memberikan arahan kepada tim *scrum* terkait dengan pekerjaan, tindakan, dan perilaku mereka. Keputusan yang dibuat, langkah-langkah yang diambil, dan cara penggunaan *Scrum* harus memperkuat nilai-nilai *scrum* dan bukan mengurangi atau melemahkan. Anggota Tim *Scrum* belajar dan mengeksplorasi nilai-nilai saat mereka menggunakan *scrum event* dan *scrum artifacts* (Schwaber & Sutherland, 2020, p. 4).

### 2.1.3.2 Scrum Team

Berdasarkan teori *Scrum Team* oleh Ken Schwaber & Jeff Sutherland Tim *Scrum* bertanggung jawab atas semua aktivitas terkait produk mulai dari kolaborasi pemangku kepentingan, verifikasi, pemeliharaan, operasi, eksperimen, penelitian dan pengembangan, dan lainnya yang mungkin diperlukan. Mereka terstruktur dan diberdayakan oleh organisasi untuk mengelola kerja sendiri. Bekerja di *Sprint* dengan kecepatan yang berkelanjutan meningkatkan fokus dan konsistensi Tim *Scrum*. Seluruh Tim *Scrum* bertanggung jawab untuk menciptakan Increment yang berharga dan berguna setiap *Sprint*. *Scrum* mendefinisikan tiga akuntabilitas khusus dalam Tim *Scrum*: *Developer, Product Owner, dan Scrum master*.

#### 2.1.3.2.1 Developer

*Developer* adalah orang-orang di Tim *Scrum* yang berkomitmen untuk menciptakan aspek apa pun yang dapat digunakan dan ditingkatkan setiap *Sprint*. Keterampilan khusus yang dibutuhkan oleh *developer* seringkali luas dan akan bervariasi dengan domain pekerjaan. Namun, *developer* selalu bertanggung jawab untuk:

- Membuat rencana untuk *Sprint, Sprint Backlog*
- Menanamkan kualitas dengan berpegang pada *Definition of Done*
- Menyesuaikan rencana mereka setiap hari menuju *Sprint Goal*
- Memegang tanggung jawab satu sama lain sebagai profesional.

#### 2.1.3.2.2 Product Owner

*Product Owner* merupakan satu orang yang memiliki tanggung jawab untuk memaksimalkan nilai produk yang dihasilkan dari kinerja *Scrum Team*. *Product Owner* juga bertanggung jawab atas pengelolaan *Product Backlog* yang efektif, yang meliputi:

- Mengembangkan dan mengomunikasikan *product goal* secara eksplisit

- Membuat dan mengomunikasikan item *product backlog* dengan jelas
- Membuat prioritas *product backlog*; dan,
- Memastikan bahwa *product backlog* transparan, terlihat dan dipahami.

*Product Owner* dapat mewakili kebutuhan banyak *stakeholders* dalam *Product Backlog* oleh karena itu jika terjadi perubahan pada *Product Backlog* dapat mendiskusikan dan mencoba meyakinkan *Product Owner*.

#### 2.1.3.2.3 Scrum Master

*Scrum Master* bertanggung jawab untuk membangun *Scrum* sebagaimana didefinisikan dalam Panduan *Scrum*. Mereka melakukan ini dengan membantu semua orang memahami teori dan praktik *Scrum*, baik di dalam Tim *Scrum* maupun di organisasi.

*Scrum Master* bertanggung jawab atas efektivitas Tim *Scrum*. Mereka melakukan ini dengan mengaktifkan Tim *Scrum* untuk meningkatkan praktiknya, dalam kerangka kerja *Scrum*. *Scrum Masters* adalah pemimpin sejati yang melayani Tim *Scrum* dan organisasi yang lebih besar. *Scrum Master* melayani Tim *Scrum* dalam beberapa cara, termasuk:

- Melatih anggota tim dalam manajemen diri dan lintas fungsi
- Membantu Tim *Scrum* untuk fokus menciptakan Increment bernilai tinggi yang memenuhi *Definition of Done*
- Menyebabkan hilangnya hambatan bagi kemajuan Tim *Scrum*; dan,
- Memastikan bahwa semua acara *Scrum* berlangsung dan bersifat positif, produktif, dan disimpan dalam kotak waktu.

#### 2.1.3.3 Scrum Event

*Sprint* adalah wadah untuk semua acara lainnya. Setiap *Scrum event* merupakan kesempatan untuk memeriksa dan mengadaptasi *Scrum artifact*. *Scrum event* secara khusus dirancang untuk

memungkinkan transparansi yang diperlukan. Kegagalan untuk mengoperasikan *scrum event* akan mengakibatkan hilangnya kesempatan untuk melakukan *inspect* dan *adapt*. Scrum event digunakan untuk menciptakan keteraturan dan meminimalkan kebutuhan yang tidak ditentukan dalam *Scrum*. Secara optimal, semua *scrum event* diadakan pada waktu dan tempat yang sama untuk mengurangi kompleksitas (Schwaber & Sutherland, 2020, p. 7).

*Scrum event* mendefinisikan yang terjadi di dalam setiap *sprint* yaitu, *sprint*, *sprint planning*, *daily scrum*, *sprint review*, and *sprint retrospective*.

#### 2.1.3.3.1 Sprint

*Sprint* merupakan inti dari *Scrum*. *Sprint* berdurasi tetap yaitu, satu bulan atau kurang untuk menciptakan konsistensi. *Sprint* baru segera dimulai setelah *Sprint* sebelumnya selesai. Selama *Sprint* dijalankan, akan ada beberapa rangkaian kegiatan didalamnya, mulai dari *Sprint Planning*, *Daily Scrum*, *Sprint Review*, hingga *Sprint Retrospective*. Selama *Sprint* berlangsung tidak ada perubahan yang dilakukan yang dapat membahayakan *Sprint Goal*, tidak menurunkan kualitas, menyempurnakan *Product Backlog* sesuai kebutuhan dan berdiskusi ulang dengan *Product Owner* seiring semakin banyak yang dipelajari.

Setiap *Sprint* dapat dianggap sebagai proyek pendek, yang memiliki tujuan *Sprint*, atau disebut *Sprint Goal*, yang tidak terlalu panjang. Jika *Sprint* dilakukan sebagai proyek panjang, *Sprint Goal* mungkin menjadi tidak valid atau using dan dapat meningkatkan kompleksitas dan resiko. *Sprint* yang lebih pendek dapat digunakan untuk menghasilkan lebih banyak pembelajaran. Jika, *Sprint Goal* sudah tidak valid, *Sprint* dapat dibatalkan oleh *Product Owner* (Schwaber & Sutherland, 2020, pp. 7-8).

### 2.1.3.3.2 Sprint Planning

*Sprint* dimulai dengan *Sprint Planning*, yaitu merencanakan pekerjaan yang akan dilakukan selama *Sprint*. Rencana yang dihasilkan ini dibuat secara kolaboratif oleh seluruh *Scrum Team*. *Sprint Planning* akan membahas penggerjaan *Product Backlog* berdasarkan tingkat kepentingannya yang ditentukan oleh *Product Owner* (Schwaber & Sutherland, 2020, p. 8).

*Sprint planning* membahas topik-topik berikut:

- Topik Satu: Mengapa *Sprint* ini penting?

*Product Owner* mengusulkan bagaimana produk dapat meningkatkan nilai dan kegunaannya dalam *Sprint* saat ini.

*Sprint Goal* harus diselesaikan sebelum *Sprint Planning* berakhir.

- Topik Dua: Apa yang Dapat Dilakukan *Sprint* ini?

Melalui diskusi dengan *Product Owner*, *Developer* memilih item dari *Product Backlog* untuk dimasukan dalam *Sprint* saat ini. *Scrum team* dapat memperbaiki item-item ini selama proses yang meningkatkan pemahaman dan kepercayaan diri. Memilih berapa banyak yang bisa diselesaikan dalam *Sprint* mungkin sulit. Namun, semakin *Developer* mengetahui tentang kinerja mereka sebelumnya, kapasitas mereka yang akan datang, dan *Definition of Done*, semakin yakin mereka akan *Sprint Forecast* mereka.

- Topik Tiga: Bagaimana pekerjaan yang dipilih akan diselesaikan?

Untuk setiap item *Product Backlog* yang dipilih, *Developer* merencanakan pekerjaan yang diperlukan untuk membuat *Increment* yang memenuhi *Definition of Done*. Hal ini dilakukan dengan cara menguraikan item *Product Backlog* menjadi pekerjaan-pekerjaan yang lebih kecil untuk satu hari atau kurang.

*Sprint Goal*, item *Product Backlog* yang dipilih untuk *Sprint*, ditambah rencana penggerjaannya disebut sebagai *Sprint Backlog*.

*Sprint Planning* juga menjadi tempat untuk menentukan batas waktu menjalankan *Sprint* hingga *Product Backlog Item* dalam *Sprint* tersebut dapat diselesaikan. Selanjutnya, *Sprint* akan dipantau oleh *Scrum Master* agar sesuai dengan apa yang telah dibuat dalam *Sprint Planning*.

#### **2.1.3.3.3 Daily Scrum**

Tujuan dari *Daily Scrum* adalah untuk memeriksa kemajuan menuju *Sprint Goal* dan mengadaptasi *Sprint Backlog* seperlunya dan menyesuaikan rencana pekerjaan yang akan datang. *Daily Scrum* dilakukan selama 15 menit untuk *Developer scrum team* setiap hari selama *sprint* berlangsung untuk mengurangi kompleksitas. *Product Owner* dan *Scrum Master* aktif mengerjakan item di *Sprint Backlog* dengan berpartisipasi sebagai *Developers*. *Daily Scrum* meningkatkan komunikasi, mengidentifikasi hambatan, mendorong pengambilan keputusan yang cepat, dan mengurangi pertemuan-pertemuan yang tidak diperlukan (Schwaber & Sutherland, 2020, p. 9).

#### **2.1.3.3.4 Sprint Review**

Tujuan dari *Sprint Review* adalah untuk memeriksa hasil *Sprint* dan menentukan *adaptation* selanjutnya. Tim *Scrum* mempresentasikan hasil kerja mereka kepada *stakeholders* dan membahas kemajuannya menuju *Product Goal*. Selama *scrum event*, *scrum team* dan *stakeholders* meninjau apa yang telah dicapai selama *Sprint*. Berdasarkan informasi ini, peserta *Sprint* berdiskusi tentang apa yang harus dilakukan selanjutnya. *Product Backlog* juga dapat disesuaikan untuk memenuhi peluang baru. *Sprint Review* merupakan *sprint event* kedua terakhir yang dibatasi hingga maksimum empat jam untuk satu bulan *Sprint* (Schwaber & Sutherland, 2020, p. 9).

#### 2.1.3.3.5 Sprint Retrospective

*Sprint Retrospective* adalah waktu untuk merencanakan cara-cara untuk meningkatkan kualitas dan efektivitas. *Scrum team* memeriksa bagaimana *Sprint* terakhir berjalan sehubungan dengan individu, interaksi, proses, alat, dan *Definition of done*. *Scrum team* membahas apa yang berjalan dengan baik selama *Sprint*, masalah apa yang dihadapi, dan bagaimana masalah itu diselesaikan atau bagaimana masalah itu akan diselesaikan kedepannya. *Scrum team* mengidentifikasi perubahan yang paling membantu untuk meningkatkan efektivitasnya. Yang paling berdampak perbaikan ditangani sesegera mungkin dan dapat ditambahkan ke *Sprint Backlog* untuk *Sprint* berikutnya. *Sprint Retrospective* dilakukan setelah *Sprint Review* dan sebelum melakukan *Sprint Planning* untuk *Sprint berikutnya* (Schwaber & Sutherland, 2020, p. 10).

#### 2.1.3.4 Scrum Artifacts

*Scrum Artifacts* mewakili hasil pekerjaan atau nilai produk. *Scrum Artifacts* dirancang untuk memaksimalkan transparansi informasi agar setiap orang yang memeriksanya memiliki dasar yang sama untuk adaptasi. Setiap artefak berisi komitmen untuk memastikannya memberikan informasi yang meningkatkan transparansi dan fokus terhadap kemajuan yang dapat diukur (Schwaber & Sutherland, 2020, pp. 10-12):

- Untuk *Product Backlog* itu adalah *Product Goal*.
- Untuk *Sprint Backlog* adalah *Sprint Goal*.
- Untuk *Increment* adalah *Definition of Done*.

Komitmen ini ada untuk memperkuat empirisme dan nilai-nilai *Scrum* untuk *Scrum team* dan *stakeholders*.

#### 2.1.3.4.1 Product Backlog

*Product Backlog* adalah daftar urutan yang muncul dari apa yang dibutuhkan untuk meningkatkan produk dan merupakan satu sumber pekerjaan yang dilakukan oleh *Scrum team*. *Item Product*

*Backlog* yang dapat dikerjakan oleh *Scrum Team* dalam satu *Sprint* dianggap sudah siap seleksi dalam acara *Sprint Planning*. *Product Backlog* akan dipecah dan didefinisikan lebih lanjut menjadi *item* yang lebih kecil dan lebih tepat untuk menambahkan detail, seperti deskripsi, pesanan, dan ukuran. *Developer* yang akan melakukan pekerjaan bertanggung jawab atas ukuran. *Product Owner* dapat mempengaruhi *Developer* dengan membantu mereka memahami dan bertukar pikiran.

- **Product Goal**

*Product Goal* merupakan gambaran produk di masa depan yang dapat berfungsi sebagai target untuk *Scrum team* untuk merencanakannya. *Product Goal* terdapat dalam *Product Backlog*. Sisa dari *Product Backlog* muncul untuk menentukan "apa" yang akan memenuhi *Product Goal*. Produk adalah sarana untuk memberikan nilai yang memiliki batas yang jelas, pemangku kepentingan yang dikenal, terdefinisi dengan baik pengguna atau pelanggan. Produk dapat berupa layanan, produk fisik, atau sesuatu yang lebih abstrak. *Product Goal* adalah tujuan jangka panjang dari *Scrum Team* dan harus memenuhi atau meninggalkan satu tujuan sebelum mengambil yang berikutnya.

#### **2.1.3.4.2 Sprint Backlog**

*Sprint Backlog* terdiri dari *Sprint Goal*, kumpulan *item* *Product Backlog* yang dipilih untuk *Sprint*, serta rencana yang dapat ditindaklanjuti untuk memberikan *Increment*. *Sprint Backlog* merupakan rencana yang dibuat oleh *Developer* dan untuk *Developer* yang mengambarkan pekerjaan yang sangat terlihat dan secara *real-time* yang akan dicapai oleh *Developer* selama *Sprint* untuk mencapai *Sprint Goal*. *Sprint Backlog* diperbarui sepanjang *Sprint* berlangsung dan berisi cukup detail sehingga dapat melihat kemajuan saat melakukan *Daily Scrum*.

- **Sprint Goal**

*Sprint Goal* adalah satu-satunya tujuan *Sprint*. Meskipun *Sprint Goal* adalah komitmen dari *Developer*, *Sprint Goal* memberikan fleksibilitas dalam hal pekerjaan yang tepat yang diperlukan untuk mencapainya. *Sprint Goal* juga menciptakan koherensi dan fokus, mendorong *Scrum team* untuk bekerja bersama. *Sprint Goal* dibuat selama *event Sprint Planning* dan kemudian ditambahkan ke *Sprint Backlog*. *Developer* mengingat *Sprint goal* selama bekerja dalam *Sprint*. Jika pekerjaan itu ternyata berbeda dari yang mereka harapkan, mereka berkolaborasi dengan *Product Owner* untuk menegosiasikan ruang lingkup *Sprint Backlog* dalam *Sprint* tanpa mempengaruhi *Sprint Goal*.

#### 2.1.3.4.3 Increment

*Increment* adalah batu loncatan menuju *Product Goal*. Setiap *Increment* adalah tambahan untuk semua *Increment* sebelumnya dan akan diverifikasi secara menyeluruh untuk memastikan bahwa semua *Increment* bekerja sama. Untuk memberikan nilai, *Increment* harus dapat digunakan. Beberapa *Increment* dapat dibuat dalam *Sprint*. Jumlah *Increment* disajikan di *Sprint Review* sehingga mendukung empirisme. Sebuah *Increment* dapat diberikan kepada para *stakeholder* sebelum akhir *Sprint*. *Sprint Review* tidak boleh dianggap sebagai gerbang untuk melepaskan nilai. Suatu pekerjaan tidak dapat dianggap sebagai bagian dari *Increment* kecuali memenuhi *Definition of Done*.

- **Definition of Done**

*Definition of Done* adalah deskripsi formal tentang status *Increment* ketika memenuhi kualitas tindakan yang diperlukan untuk produk. Saat *item Product Backlog* memenuhi *Definition of Done*, *Increment* lahir. *Definition of Done* menciptakan transparansi dengan memberikan pemahaman bersama kepada semua orang tentang pekerjaan apa yang telah selesai sebagai bagian dari *Increment*. Item *Product Backlog*

yang tidak memenuhi *Definition of Done* tidak bisa dirilis atau bahkan dipresentasikan di *Sprint Review* dan dikembalikan ke *Produk Backlog* untuk menjadi pertimbangan di masa mendatang. Jika *Definition of Done* untuk *Increment* merupakan bagian dari standar organisasi sehingga semua *Scrum Team* harus mengikutinya. *Developer* diharuskan untuk menyesuaikan diri dengan *Definition of Done*. *Scrum Team* yang bekerja sama pada suatu produk harus saling mendefinisikan dan mematuhi *Definition of Done* yang sama.

#### 2.1.4 Project Management App Trello

Project management app menyediakan fitur untuk membantu untuk mengelola semua jenis proyek, alur kerja, atau pelacakan tugas dan membagi proyek menjadi beberapa tahapan utama seperti yang ada di dalam *Scrum* (Trello, 2022)

#### 2.1.5 Figma

Figma merupakan sebuah *tools* untuk design berupa website yang terhubung dengan *cloud*. Figma berbasis *vector* sehingga cocok digunakan untuk mendesign UI *mobile application* maupun *web application*. Kelebihan figma adalah dengan berbasis *cloud* sehingga memungkinkan penggunaanya dalam berkolaborasi secara *realtime* dalam satu *platform* tanpa harus menggabungkannya secara manual (Figma, 2022).

#### 2.1.6 Github

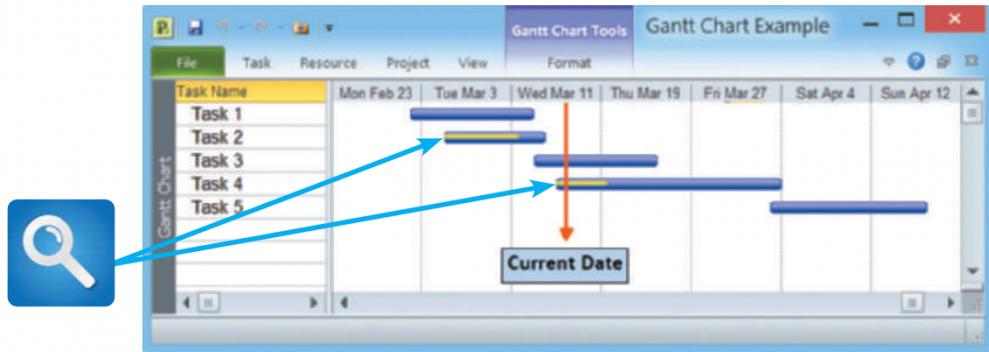
Github merupakan sebuah platform jaringan sosial yang dirancang khusus bagi developer untuk manajemen proyek dengan sistem versioning code yang menerapkan Git. Git merupakan sebuah software VCS (Version Controlled System) yang dapat mengontrol pembagian versi aplikasi yang digunakan untuk menyimpan update versi software tanpa menghapus atau menimbulkan sistem sebelumnya (Github, 2022).

Berikut perintah dalam github yang dapat digunakan:

- *git init* : merupakan perintah yang digunakan untuk membuat repository baru
- *git branch* : perintah yang digunakan untuk menampilkan semua branch yang ada pada repository
- *git clone* : perintah yang digunakan untuk melakukan checkout dari repository dan untuk membuat salinan repository pada perangkat local
- *git config* : perintah yang digunakan untuk mengatur konfigurasi tertentu
- *git status* : perintah yang digunakan untuk menampilkan daftar file yang berubah dan file yang ingin ditambahkan atau di-commit.
- *git checkout* : perintah yang digunakan untuk membuat branch dan berpindah di dalamnya
- *git pull* : perintah yang digunakan untuk menggabungkan semua perubahan yang ada di remote repository ke direktori local
- *git add* : perintah yang digunakan untuk menambahkan file kedalam index
- *git commit* : perintah yang digunakan untuk melakukan commit pada perubahan
- *git push* : perintah yang digunakan untuk mengirim perubahan ke master branch
- *git merge* : perintah yang digunakan untuk menggabungkan sebuah branch ke branch aktif

#### 2.1.7 Gantt Chart

Gantt chart adalah bagan batang horizontal yang mewakili serangkaian tugas. Posisi bar menunjukkan waktu mulai dan berakhir yang direncanakan dari setiap tugas, dan panjang bar menunjukkan durasinya. Gantt chart juga dapat menyederhanakan proyek kompleks dengan menggabungkan beberapa kegiatan ke dalam kelompok tugas yang berisi dari sub tugas. Hal ini memungkinkan proyek yang kompleks untuk dilihat sebagai satu set modul terintegrasi (Tilley & Rosenblatt, 2016, p. 73). Berikut contoh penggambaran gantt chart:



Gambar 2.1 Contoh Gantt Chart

(Tilley & Rosenblatt, 2016, p. 73).

## 2.2 Teori Terkait Desain Model Aplikasi

### 2.2.1 UML (Unified Modelling Language)

A. Dennis (2015: 34) menjelaskan bahwa *Unified Modelling language* atau biasa disingkat UML merupakan sebuah teknik standar dalam membuat diagram. Tujuan utama dari UML untuk menyediakan daftar istilah-istilah umum untuk ketentuan yang berorientasi objek dan sebuah teknik dalam membuat diagram yang cukup luas untuk memodelkan segala bentuk pemgembangan sistem dari analisis hingga implementasi. Terdapat beberapa jenis model pada UML untuk memodelkan suatu sistem dalam pengembangan sistem yang akan dijelaskan lebih lanjut di bawah ini.

#### 2.2.1.1 Use Case Diagram

Use case diagram merupakan sebuah model diagram untuk menggambarkan proses bisnis atau sistem. Use case dapat mendokumentasikan sistem saat ini atau sistem yang baru atau sedang dikembangkan. (Dennis & Wixom & Tegarden, 2015:120).

#### 2.2.1.2 Use case Description

Use-case Description mendokumentasikan lebih dalam untuk setiap individu use case. Use-case description berisi semua informasi yang diperlukan untuk mendokumentasikan fungsionalitas dari proses bisnis. (Dennis &

Wixom & Tegarden, 2015:140). Use-case description terdiri dari beberapa elemen, yaitu:

a. *Overview Information*

*Overview Information* mengidentifikasi use-case dan menjelaskan latar belakang tentang use-case tersebut. Pada elemen ini terdapat beberapa komponen yaitu:

- Nama use-case, nama use-case harus mendeskripsikan kegiatan usecase
- Kode use-case, untuk mengidentifikasi use-case
- Jenis use-case, apakah tipenya gambaran atau detail dan penting atau tidak
- aktor utama, aktor yang akan menggunakan use-case
- tujuan use-case, untuk memberikan
- tingkat kepentingan, apakah tingkat use-case tersebut penting atau tidak
- pemicu atau *trigger*, hal yang memicu use-case tersebut dijalankan
- stakeholder, aktor yang berpengaruh dalam use-case

b. *Relationship*

*Relationship* menjelaskan hubungan use-case tersebut dengan use-case lainnya. Terdapat 4 dasar hubungan antar usecase yaitu, *association*, *extended*, *included*, *generalization*.

c. *Flow of Event*

*Flow of Event* menjelaskan alur proses dari use-case dari awal use-case dimulai hingga akhir. Ada 3 kategori *event* yang dapat di dokumentasikan, yaitu:

- *Normal flow*, yang merupakan flow utama dari use-case yang dijelaskan secara simple
- *Sub-flow*, apabila normal flow memiliki beberapa flow kompleks maka dapat dimasukkan kedalam sub flow.

- *Alternative* atau *exceptional* flow, merupakan flow yang harus didokumentasikan meskipun bukan merupakan kriteria utama

Use Case Name:	Make Old Patient Appt	ID:	2	Importance Level:	Low								
Primary Actor:	Old Patient	Use Case Type:	Detail, Essential										
Stakeholders and Interests: Old Patient – wants to make, change, or cancel an appointment Doctor – wants to ensure patient's needs are met in a timely manner													
Brief Description: This use case describes how we make an appointment as well as changing or canceling an appointment for a previously seen patient.													
Trigger: Patient calls and asks for a new appointment or asks to cancel or change an existing appointment													
Type:	External												
Relationships: Association: Old Patient Include: Extend: Update Patient Information Generalization: Manage Appointments													
Normal Flow of Events: 1. The Patient contacts the office regarding an appointment. 2. The Patient provides the Receptionist with his or her name and address. 3. If the Patient's information has changed Execute the Update Patient Information use case. 4. If the Patient's payment arrangements has changed Execute the Make Payments Arrangements use case. 5. The Receptionist asks Patient if he or she would like to make a new appointment, cancel an existing appointment, or change an existing appointment. If the patient wants to make a new appointment, the S-1: new appointment subflow is performed. If the patient wants to cancel an existing appointment, the S-2: cancel appointment subflow is performed. If the patient wants to change an existing appointment, the S-3: change appointment subflow is performed. 6. The Receptionist provides the results of the transaction to the Patient.													
SubFlows: S-1: New Appointment 1. The Receptionist asks the Patient for possible appointment times. 2. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment. S-2: Cancel Appointment 1. The Receptionist asks the Patient for the old appointment time. 2. The Receptionist finds the current appointment in the appointment file and cancels it. S-3: Change Appointment 1. The Receptionist performs the S-2: cancel appointment subflow. 2. The Receptionist performs the S-1: new appointment subflow.													
Alternate/Exceptional Flows: S-1, 2a1: The Receptionist proposes some alternative appointment times based on what is available in the appointment schedule. S-1, 2a2: The Patient chooses one of the proposed times or decides not to make an appointment.													

Gambar 2.2 Contoh Usecase Description

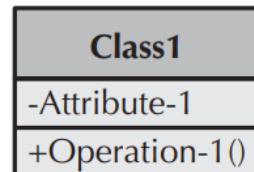
(source: Dennis & Wixom & Tegarden, 2015:168)

### 2.2.1.3 Class Diagram

Class diagram merupakan model statis yang menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam waktu ke waktu (Dennis &

Wixom & Tegarden, 2015:176). Terdapat beberapa komponen dalam class diagram, yaitu:

a. *Class*



Gambar 2.3 Contoh Class

(source: Dennis & Wixom & Tegarden, 2015:178)

Class merepresentasikan sebuah objek yang mana dibutuhkan oleh sistem untuk memperoleh atau menyimpan data. Class memiliki sebuah nama

b. *Attribute*

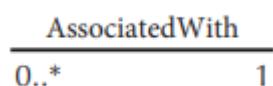
Attribute merupakan sebuah properti yang dimiliki oleh kelas yang menggambarkan kelas tersebut. Pada **Gambar 2.3** attribut terletak di bagian tengah.

c. *Operation*

Operation merepresentasikan sebuah fungsi yang dapat dijalankan oleh class tersebut. Pada **Gambar 2.3** operation terletak di bagian paling bawah.

d. *Association*

Association merupakan hubungan antara beberapa class atau class itu sendiri.



Gambar 2.4 Contoh Association

(source: Dennis & Wixom & Tegarden, 2015:178)

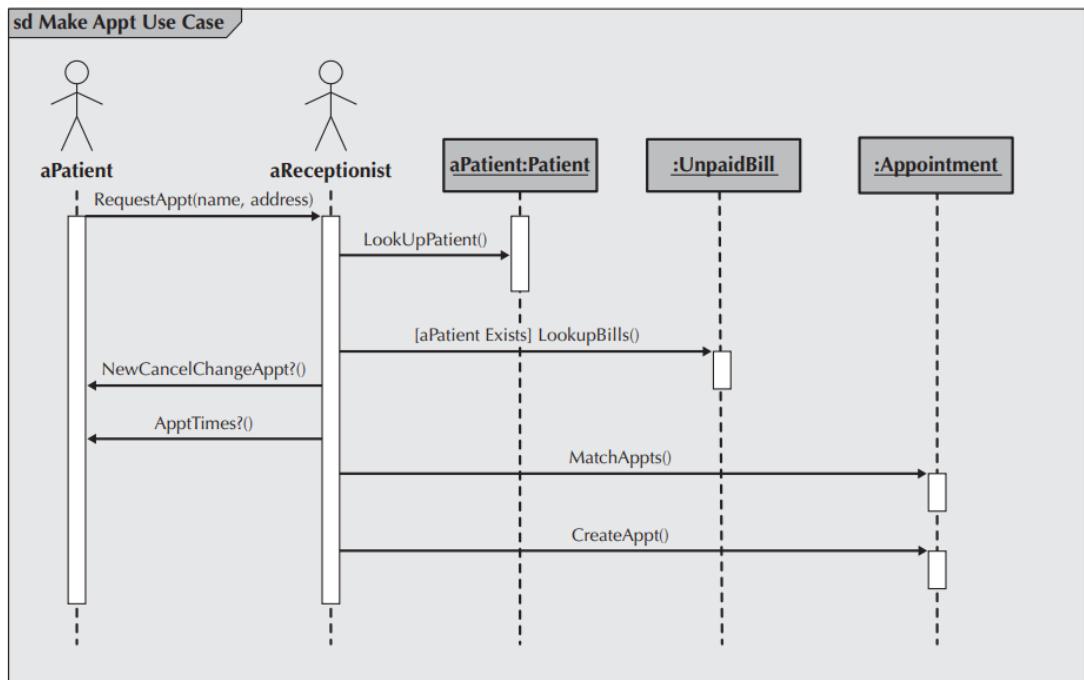
terdapat 2 bentuk pada association:

- Aggregation, merupakan bentuk hubungan antar kelas namun tidak saling ketergantungan
- Composition, merupakan bentuk hubungan antar kelas yang saling ketergantungan antar kelas

e. Generalization

Generalization menjelaskan sebuah kelas yang merupakan turunan dari induk kelas yang mewarisi attribute atau operartion dari induk kelas tersebut.

#### 2.2.1.4 Sequence Diagram

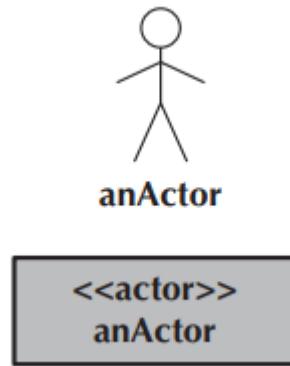


Gambar 2.5 Contoh Sequence Diagram

(sumber: Dennis & Wixom & Tegarden, 2015:205).

Sequence Diagram merupakan sebuah model dinamis yang menunjukkan alur pesan secara yang dikirim antar objek dalam sebuah interaksi yang telah ditentukan (Dennis & Wixom & Tegarden, 2015:176). Sequence diagram memiliki beberapa komponen diantar lain sebagai berikut:

a. Actor



Gambar 2.6 Actor pada Squeuence Diagram

(sumber: Dennis & Wixom & Tegarden, 2015:205).

Actor merupakan sebuah komponen yang menggambarkan user yang berinteraksi dengan sistem

b. Object



Gambar 2.7 Object

(sumber: Dennis & Wixom & Tegarden, 2015:205).

Objek merupakan komponen yang bertugas mengirim atau menerima pesan

c. Lifeline



Gambar 2.8 Lifeline

(sumber: Dennis & Wixom & Tegarden, 2015:205).

Lifeline merupakan sebuah komponen untuk menggambarkan jalur interaksi objek.

d. Execution Occurrence

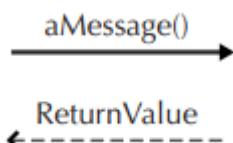


Gambar 2.9 Excecution Occurrence

(sumber: Dennis & Wixom & Tegarden, 2015:205).

Execution Occurrence merupakan komponen yang menggambarkan bahwa sebuah objek dalam keadaan aktif atau sedang menerima atau mengirim pesan pada lifeline.

e. Message

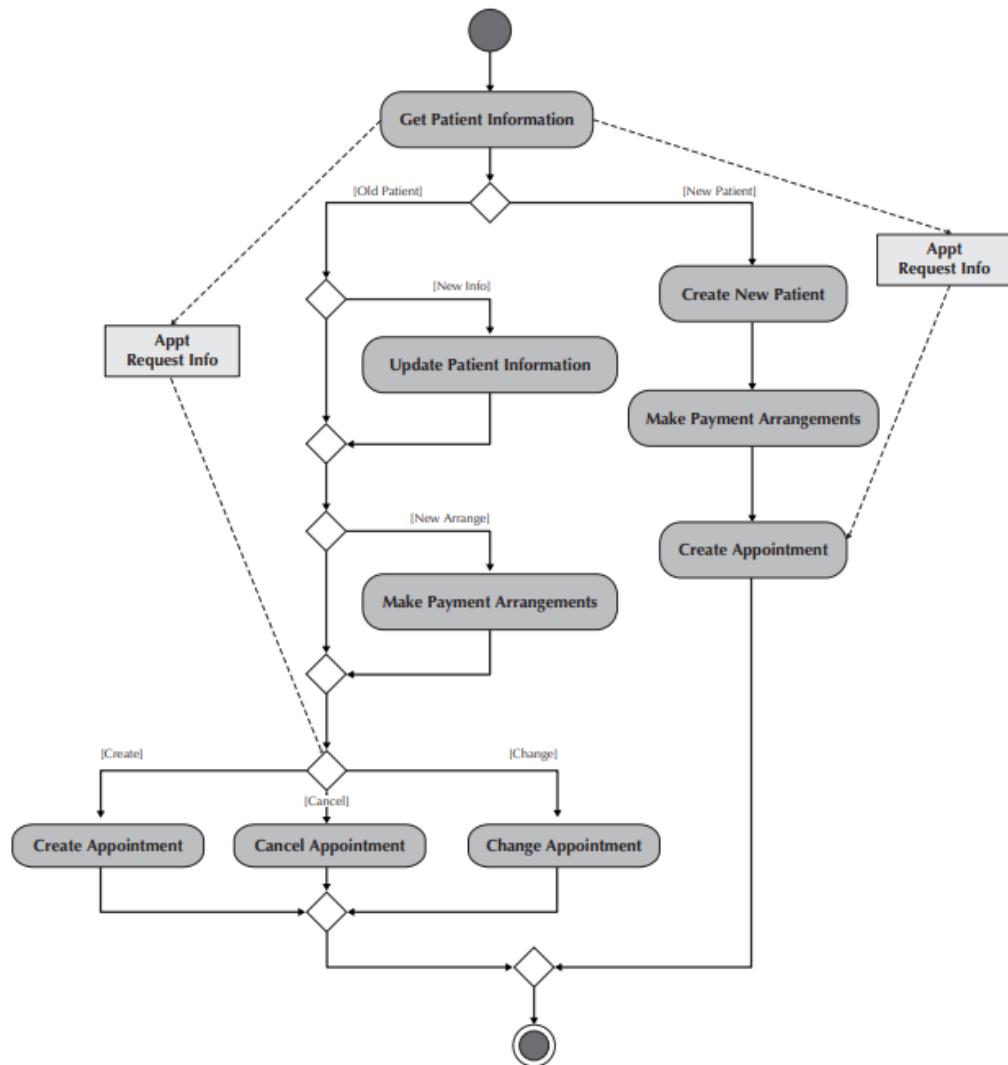


Gambar 2.10 Message

(sumber: Dennis & Wixom & Tegarden, 2015:205).

Message merupakan pesan yang disampaikan dari object satu ke lainnya.

### 2.2.1.5 Activity Diagram



Gambar 2.11 Contoh Activity Diagram

(sumber: Dennis & Wixom & Tegarden, 2015:133).

Activity diagram merupakan model yang menjelaskan proses-proses yang terjadi pada sebuah sistem. Activity diagram biasa digunakan untuk menggambarkan perilaku sebuah proses yang tidak bergantung pada objek. Terdapat beberapa komponen pada activity diagram, yaitu:

a. Action dan Activity



Gambar 2.12 Activity dan Action

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Action merupakan sebuah perilaku sederhana yang tidak dapat diurai, sedangkan activity kumpulan dari beberapa action.

b. Object nodes



Gambar 2.13 Object node

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Object nodes mewakili aliran informasi dari sebuah activity ke activity lainnya.

c. Control flow



Gambar 2.14 Control Flow

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Control flow merupakan komponen yang memodelkan jalur eksekusi sebuah action atau activity.

d. Object flow



Gambar 2.15 Object Flow

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Object flow memodelkan aliran object pada action atau activity

e. Initial node



Gambar 2.16 Initial Node

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Initial node merupakan komponen yang merepresentasikan awal dari action atau activity.

f. Final-activity node

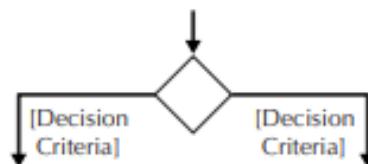


Gambar 2.17 Final-activity Node

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Final-activity node merupakan komponen yang menggambarkan akhir sebuah acitivity atau action.

g. Decision Node

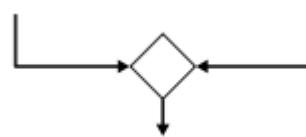


Gambar 2.18 Decision Node

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Decision node merupakan komponen yang merepresentasikan kondisi pengujian untuk memastikan bahwa aliran hanya berjalan pada satu aliran.

h. Merge Node

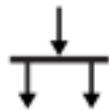


Gambar 2.19 Merge Node

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Merge node berfungsi untuk menyatukan aliran yang terjadi pada decision Node.

i. Fork Node

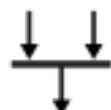


Gambar 2.20 Fork Node

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Fork node merupakan komponen yang memisahkan perilaku menjadi beberapa aliran secara parallel.

j. Join Node

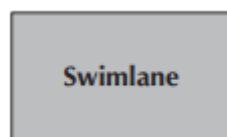


Gambar 2.21 Join Node

(sumber: Dennis & Wixom & Tegarden, 2015:132)

Join node merupakan komponen yang menyatukan aliran yang terjadi pada fork node.

k. Swimlane



Gambar 2.22 Swimlane

(sumber: Dennis & Wixom & Tegarden, 2015:132).

Swimlane berfungsi untuk memecahkan activity diagram menjadi baris dan kolom untuk menetapkan activity atau action kepada object atau individu untuk menjalankan activity atau action tersebut

## 2.2.2 Data Dictionary

Data dictionary adalah tempat penyimpanan informasi yang menggambarkan data dalam database. Banyaknya informasi dan cara informasi

digunakan bervariasi dengan DBMS (Connoly & Begg, 2015, p. 98). Biasanya, katalog sistem menyimpan:

- nama, jenis, dan ukuran item data;
- nama hubungan;
- batasan integritas pada data;
- nama pengguna yang berwenang yang memiliki akses ke data;
- item data yang dapat diakses oleh setiap pengguna dan jenis akses yang diizinkan; misalnya, menyisipkan, memperbarui, menghapus, atau membaca akses;
- skema eksternal, konseptual, dan internal dan pemetaan antar skema,
- statistik penggunaan, seperti frekuensi transaksi dan penghitungan nomor
- dari akses yang dibuat ke objek dalam database.

<i>Entity name</i>	<i>Attributes</i>	<i>Description</i>	<i>Data Type &amp; Length</i>	<i>Nulls</i>	<i>Multi-valued</i>	...
<b>Staff</b>	<b>staffNo</b> <b>name</b> <b>fName</b> <b>IName</b> <b>position</b> <b>sex</b> <b>DOB</b>	Uniquely identifies a member of staff  First name of staff Last name of staff Job title of member of staff Gender of member of staff Date of birth of member of staff	5 variable characters  15 variable characters 15 variable characters 10 variable characters 1 character (M or F) Date	No No No No Yes Yes	No No No No No No	
<b>PropertyForRent</b>	<b>propertyNo</b>	Uniquely identifies a property for rent	5 variable characters	No	No	

Gambar 2.23 Contoh Data Dictionary

(Connoly & Begg, 2015, p. 516)

## 2.3 Teori yang Berkaitan dengan Pengembangan Mobile Application

### 2.3.1 Dart

Dart adalah Bahasa pemograman yang dikembangkan oleh Google, yang ditujukan untuk mengembangkan aplikasi multi-plataform secara cepat. Tujuan utamanya adalah untuk mempertahankan sifat dinamis Java Script, tetapi memiliki kinerja yang lebih baik dan dapat diperluas ke sarana untuk proyek-proyek yang lebih besar dan juga dapat dikompilasi silang ke dalam bentuk javascript (A. M. Hasan, 2020). Oleh karena itu, dart dapat kompilasi code melalui beberapa target seperti web, seluler, ataupun desktop. Dart

merupakan bahasa yang object-oriented, class-based, dan garbage-collected dengan sintaks C.

### 2.3.2 Flutter

Flutter.dev yang merupakan situs resmi dari framework Flutter menjelaskan bahwa Flutter merupakan sebuah *open-source* untuk *User Interface SDK (Software Development Kit)* yang dikembangkan oleh Google. Flutter sendiri dapat mengembangkan aplikasi lintas platform banyak dengan menggunakan kode tunggal. Flutter menggunakan dart sebagai bahasa pemogramannya (Flutter, 2022).

### 2.3.3 HTTP Request Methods

*Hypertext Transfer Protocol* atau biasa disebut HTTP merupakan suatu protokol transfer data. Protokol HTTP dirancang untuk melakukan permintaan dan tanggapan sehingga memungkinkan terjadinya komunikasi antara *client* dan *web server* (Prajapati, P., Patel, N., & Shah, P., 2019). Data tersebut dapat berupa file, dokumen, gambar atau yang lainnya. Proses tersebut berlangsung antara web server dengan perangkat klien atau bisa juga sebaliknya. Klien dapat melakukan permintaan atau *request* dengan mengakses URL (Uniform Resource Locator) yang menjadi sumber. Ada beberapa *request* yang sering dilakukan oleh perangkat klien, yaitu:

a. GET

Sesuai dengan Namanya , metode GET berfungsi hanya untuk menerima data dari sumber.

b. POST

metode POST berfungsi untuk menampung hasil inputan yang telah dilakukan oleh klien.

c. PUT

Metode PUT merupakan request dari klien untuk memodifikasi data yang ada pada sumber yang telah disimpan dengan data yang terlampir oleh klien.

d. DELETE

Metode DELETE berfungsi untuk menghapus sumber yang diminta oleh klien

#### 2.3.4 Firebase

Firebase merupakan sebuah layanan yang dikembangkan oleh Google untuk membudahkan developer dalam mengembangkan suatu aplikasi. menurut L. Moroney (2015: 1) dalam bukunya yang berjudul “Definitive Guide to Firebase” menjelaskan bahwa firebase dikembangkan dengan tujuan untuk menyediakan alat dan infrastuktur yang dibutuhkan untuk mengembangkan sebuah aplikasi, dan meningkatkan bisnis. Firebase dibangun diatas 3 pilar, kembangkan, tingkatkan, dan dapatkan.

#### 2.3.5 Laravel

Bagwan M. K. I. (2019) dalam jurnalnya yang berjudul “A Modern Review on Laravel-PHP Framework” menjelaskan bahwa laravel merupakan framework web berbasis PHP untuk membangun sebuah aplikasi web kelas atas menggunakan sintaks yang signifikan dan anggun. Laravel adalah salah satu Framework PHP yang menggunakan konsep *Model-View-Controller* (MVC) yaitu metode aplikasi dengan memisahkan data dari tampilan berdasarkan komponen- komponen aplikasi, seperti manipulasi data, controller, dan user interface. Laravel berfokus pada bagian end-user, yaitu pada kejelasan dan kesederhanaannya baik dalam penulisan maupun tampilan, serta dapat menghasilkan fungsionalitas aplikasi yang bekerja sebagaimana mestinya.

#### 2.3.6 Database Management System (DBMS)

Menurut Rahmi Fitri dalam buku nya yang berjudul “pemrograman basis data menggunakan MySQL” menjelaskan bahwa *Database Management System* atau biasa disingkat DBMS merupakan perangkat lunak yang dirancang untuk dapat melakukan pengaturan dan mengelola koleksi data dalam jumlah besar dan dapat memanipulasi data secara mudah.

Menurut James A.Hall, DBMS adalah sebuah sistem perangkat lunak khusus yang di program untuk mengetahui elemen data mana yang bisa diakses oleh pemakai.

Tujuan utama dari DBMS adalah untuk menyediakan cara untuk menyimpan dan mengambil informasi database yang nyaman dan efisien. DBMS memiliki keuntungan seperti:

- Mengontrol Redundansi: Redundansi data mengacu pada duplikasi data
- Peningkatan Berbagi Data: DBMS memungkinkan pengguna untuk berbagi data di sejumlah program aplikasi.
- Integritas Data: Integritas berarti bahwa data dalam database akurat.
- Keamanan : Memiliki otoritas penuh atas data operasional, memungkinkan DBA memastikan bahwa satu-satunya akses ke database adalah melalui saluran yang tepat.
- Konsistensi Data : Dengan menghilangkan redundansi data sangat mengurangi peluang terjadinya inkonsistensi.
- Akses Data yang Efisien : Data dikelola oleh DBMS dan semua akses ke data tersebut melalui DBMS yang menyediakan kunci untuk pemrosesan data yang efektif
- Penegakan Standar : Dengan data terpusat, DBA dapat menetapkan dan menegakkan standar data yang mungkin termasuk konvensi penamaan, standar kualitas data, dll.
- Kemandirian Data : Dalam sistem basis data, sistem manajemen basis data menyediakan antarmuka antara program aplikasi dan data. Ketika perubahan dibuat pada representasi data, meta data diperoleh DBMS diubah tetapi DBMS tetap menyediakan data ke program aplikasi di cara yang digunakan sebelumnya.
- Pengurangan Waktu Pengembangan dan Pemeliharaan Aplikasi : DBMS mendukung banyak fungsi penting yang umum untuk banyak aplikasi, mengakses data yang disimpan dalam DBMS, yang memfasilitasi pengembangan cepat dari aplikasi.

### 2.3.6.1 Mysql

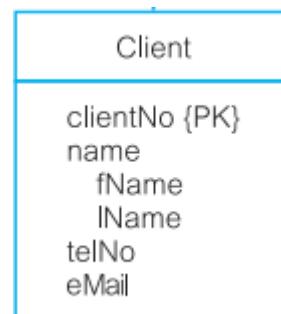
Menurut Rahmi Fitri (2020: 1) dalam buku nya yang berjudul “pemrograman basis data menggunakan MySQL” menjelaskan bahwa MySQL adalah DBMS yang *open Source* dengan dua bentuk lisensi yaitu *free software* (perangkat lunak bebas) dan shareware (perangkat lunak berpemilik yang penggunanya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) Sehingga Anda pakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada

### 2.3.6.2 Entity-Relationship Modeling

Entity-Relationship (ER) Modeling merupakan pendekatan top-down untuk desain database yang dimulai dengan mengidentifikasi data penting yang disebut entitas dan hubungan antar data yang harus diwakili dalam model (). Terdapat beberapa komponen yang dimiliki oleh ER Modeling, yaitu:

a. *Entity Type*

Entity type merupakan komponen yang merepresentasikan sekelompok objek dengan sifat yang sama (Connolly & Begg, 2015: 406). Objek tersebut dapat berupa objek fisik atau abstrak.

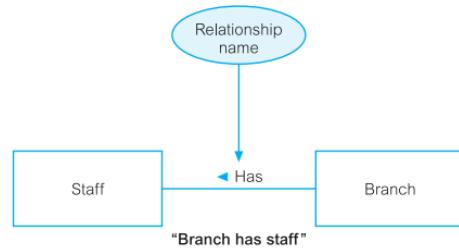


Gambar 2.24 Contoh Entity

(sumber: Connolly & Begg, 2015: 418).

b. Relationship Type

Relationship Type hubungan antara tiap-tiap entity. Setiap *relationship type* diberikan keterangan nama untuk menjelaskan hubungannya.



Gambar 2.25 Contoh Relationship Type

(sumber: Connolly & Begg, 2015: 410).

c. Attribute

Attribute merupakan propertiy yang dimiliki oleh entity. Contohnya terdapat pada Gambar 2.24. *name*, *telNo*, *eMail* merupakan attribute dari entitas Client

d. Cardinality

Cardinality menjelaskan jumlah maksimum kemungkinan terjadinya suatu hubungan untuk entitas yang berpartisipasi dalam suatu hubungan tertentu.

e. Multiplicity

Multiplicity merupakan jumlah kemungkinan terjadinya hubungan antar *entity type* pada kejadian tunggal melalui hubungan tertentu.

#### 2.3.6.3 Data-Definition Language (DDL)

Data-Definition Language digunakan untuk menentukan skema database dengan satu set definisi yang diungkapkan oleh bahasa khusus. DDL juga digunakan untuk menentukan properti tambahan dari data.

#### 2.3.6.4 Data-Manipulation Language (DML)

Data-Manipulation Language (DML) adalah bahasa yang memungkinkan pengguna untuk mengakses atau memanipulasi data seperti yang diatur oleh model data yang sesuai. Jenis-jenis akses tersebut adalah:

- Pengambilan informasi yang disimpan dalam database
- Penyisipan informasi baru ke dalam database
- Penghapusan informasi dari database
- Modifikasi informasi yang disimpan dalam database

#### 2.3.7 Black Box Testing

Menurut Yudie Irawan dalam jurnalnya yang berjudul “System Testing using Black Box Testing Equivalence Partitioning (Case Study at Garbage Bank Management Information System on Karya Sentosa” menjelaskan bahwa Black Box Testing atau bisa juga disebut Behavioral Testing, berfokus pada persyaratan fungsionalitas dari perangkat lunak. Artinya, Black Box Testing memungkinkan developer untuk memperoleh kumpulan input kondisi yang akan menjalankan semua persyaratan fungsionalitas dari perangkat lunak yang diuji. Hal tersebut untuk menemukan beberapa kategori error seperti fungsi yang salah, kesalahan interface, kesalahan dalam mengakses data, kesalahan perilaku pengguna, dan kesalahan inisialisasi atau terminasi.

### 2.4 Teori berkaitan dengan evaluasi

#### 2.4.1 Eight Golden Rules of Interface Design

Menurut Shneiderman (2016) dalam bukunya yang berjudul “Designing the User Interface: Strategies for Effective Human Computer Interaction”, terdapat delapan aturan emas yang berlaku di Sebagian besar sistem interaktif, yaitu:

1. *Strive for consistency*

Urutan tindakan yang konsisten diperlukan dalam situasi yang serupa. Terminologi identik harus digunakan dalam *prompt*, menu, layar bantuan dan memiliki desain warna, *font*, *layout* yang konsisten. Kecuali konfirmasi yang diperlukan untuk perintah yang dihapus atau pemberitahuan tidak ada kata sandi harus dapat dipahami dan dibatasi jumlahnya.

*2. Seek universal usability*

Mengenali kebutuhan yang beragam pengguna dan desain untuk plastisitas, memfasilitasi transformasi isi. Perbedaan pemula hingga ahli, rentang usia, kecacatan, variasi internasional, dan keragaman teknologi masing-masing memperkaya spektrum persyaratan yang memandu desain. Menambahkan fitur untuk pemula, seperti penjelasan, dan fitur untuk para ahli, seperti pintasan dan tempo yang lebih cepat, memperkaya desain antarmuka dan meningkatkan kualitas yang dirasakan.

*3. Offer informative feedback*

Untuk setiap tindakan pengguna, harus ada yang menjadi umpan balik dari antarmuka. Untuk tindakan sering dan kecil, responsnya bisa sederhana, sedangkan untuk tindakan yang jarang dan tindakan besar, responsnya harus lebih substansial. Presentasi visual dari objek yang menarik memberikan lingkungan yang nyaman untuk menunjukkan perubahan secara eksplisit.

*4. Design dialog to yield closure*

Urutan tindakan harus diatur ke dalam kelompok-kelompok dengan awal, tengah, dan akhir. Umpan balik yang informatif pada penyelesaian kelompok tindakan memberi pengguna kepuasan pencapaian, rasa lega, sinyal untuk membatalkan rencana darurat yang mereka pikirkan, dan indikator untuk mempersiapkan kelompok tindakan berikutnya.

*5. Prevent errors*

Sebisa mungkin, rancang antarmuka sedemikian rupa agar pengguna tidak dapat membuat kesalahan serius; misalnya, Contohnya memberi warna abu-abu pada menu yang tidak diperlukan atau pada menu yang dilarang, melarang huruf atau alfabet pada isian angka. Jika pengguna membuat kesalahan, antarmuka harus menawarkan instruksi sederhana, konstruktif, dan spesifik untuk pemulihannya. Kesalahan tindakan harus membiarkan status antarmuka tidak berubah, atau antarmuka harus memberikan instruksi tentang memulihkan keadaan.

*6. Permit easy reversal of action*

Sebisa mungkin tindakan harus dapat dibatalkan. Fitur ini mengurangi kecemasan, karena pengguna tahu bahwa kesalahan dapat dibatalkan, dan mendorong eksplorasi pilihan yang tidak biasa. Satuan reversibilitas mungkin menjadi satu tindakan, tugas entri data, atau sekelompok tindakan, seperti entri blok nama-alamat.

#### *7. Keep users in control*

Pengguna berpengalaman sangat menginginkan merasa bahwa mereka bertanggung jawab atas antarmuka dan bahwa antarmuka merespon tindakan mereka. Mereka tidak ingin kejutan atau perubahan perilaku, dan mereka terganggu oleh urutan entri data yang membosankan, kesulitan dalam memperoleh yang diperlukan informasi, dan ketidakmampuan untuk menghasilkan hasil yang diinginkan.

#### *8. Reduce short term memory load*

Kapasitas manusia yang terbatas untuk pemrosesan informasi dalam memori jangka pendek mengharuskan desainer menghindari antarmuka di mana pengguna harus mengingat informasi dari satu tampilan dan kemudian menggunakan informasi tersebut pada tampilan lain.

### 2.4.2 Lima Faktor manusia terukur

Menurut Shneiderman (2016) dalam bukunya yang berjudul “Designing the User Interface: Strategies for Effective Human Computer Interaction” perancangan antarmuka sebaiknya tidak lupa untuk mengikutsertakan evaluasi terhadap lima faktor terukur manusia sebagai berikut:

#### *1. Time to learn*

Berapa lama waktu yang dibutuhkan oleh pengguna untuk mempelajari fungsi-fungsi dalam sebuah aplikasi hingga akhirnya dapat digunakan dengan baik.

#### *2. Speed of performance*

Berapa lama waktu yang dibutuhkan suatu fungsi atau serangkaian tugas dalam aplikasi tersebut dilakukan.

3. *Rate of errors by users*

Berapa banyak dan jenis kesalahan apa yang dilakukan pengguna dalam melaksanakan serangkaian tugas.

4. *Retention over time*

Seberapa lama pengguna mempertahankan ingatan dan pengetahuannya setelah satu jam, sehari, atau seminggu.

5. *Subjective satisfaction*

Seberapa puas pengguna menggunakan berbagai aspek dari antarmuka.

## **BAB 3.**

### **METODE PENELITIAN**

#### **3.1 Kerangka Pikir**

##### **3.1.1 Membuat Product Backlog**

Pada tahap ini, kami akan mengumpulkan fungsionalitas yang diinginkan dari aplikasi yang berpotensi untuk dikembangkan lebih lanjut. Setelah tujuan diketahui kami akan membaginya menjadi bagian-bagian kecil untuk menentukan skala prioritas dan membuat to-do list untuk dikerjakan. Setelah to-do list dibuat tahapan selanjutnya adalah mem-breakdown kebutuhan apa saja yang diperlukan dan membuat sprint planning.

Tabel 3.1 Product Backlog

<b><i>Backlog</i></b>	<b><i>Estimation</i></b>	<b><i>Priority</i></b>
Sebagai user, saya dapat melakukan register agar terdaftar di aplikasi dan bisa menggunakan layanan aplikasi	4	High
Sebagai user, saya dapat mengakses akun aplikasi dengan melakukan login	3	High
Sebagai user, saya dapat mengakses fitur scan barcode pada main page	3	High
Sebagai user, saya dapat keluar dari akun aplikasi dengan melakukan logout	3	High
Sebagai user, saya dapat melihat daftar produk yang terdapat pada tiap merchant	4	High
<b>Sebagai user, saya dapat melihat daftar transaksi yang sedang berlangsung</b>	4	High

Sebagai user, saya dapat melihat status transaksi yang sedang berlangsung	3	High
Sebagai user, saya dapat menambahkan data produk baru	4	High
Sebagai user, saya dapat mengubah data produk	5	High
Sebagai user, saya dapat menghapus data produk	3	High
Sebagai user, saya dapat melihat reporting summary	4	High
Sebagai user, saya dapat melakukan top up virtual account	5	High
Sebagai user, saya dapat melakukan pembayaran melalui virtual account	4	High
Sebagai user, saya dapat membatalkan transaksi yang sedang berlangsung	3	Medium
Sebagai user, saya dapat melihat riwayat transaksi	3	Medium
Sebagai user, saya dapat melihat discount pada tiap merchant	3	Low
Sebagai user, saya dapat mengirim pesan kepada merchant	3	Low
Sebagai user, saya dapat mengubah profil	3	Low

### 3.1.2 Membuat Sprint Planing

Pada tahap ini, fitur dan fungsi teratas dalam *product backlog* akan disusun kembali menjadi *sprint backlog* berdasarkan prioritas dan selanjutnya menentukan bagaimana cara untuk menyelesaikan apa yang ada dalam *sprint backlog*.

*Sprint planning meeting* akan diadakan di setiap awal *sprint* dengan *development team* dan *product owner* untuk membuat kesepakatan tentang

pekerjaan yang harus dilakukan pada saat *sprint* berikutnya dan memutuskan fitur dan fungsi yang harus diimplementasikan. Fitur-fitur akan diimplementasikan dan diuji setiap hari melalui pertemuan yang disebut *daily scrum*. Berikut *sprint backlog* dari Comander App:

Tabel 3.2 Sprint Planning

Task Name	Type	Priority	Story Points
<b>Sprint 1 (1 Maret 2022 - 15 Maret 2022)</b>	<i>Sprint</i>	<i>High</i>	<b>21</b>
Membuat UI Homepage ComanderApp	<i>Sub-task</i>	<i>High</i>	3
Membuat UI Scan Barcode Page ComanderApp	<i>Sub-task</i>	<i>High</i>	2
Membuat UI Transaction Page ComanderApp	<i>Sub-task</i>	<i>High</i>	5
Membuat UI SignUp Page ComanderApp	<i>Sub-task</i>	<i>High</i>	2
Membuat UI Login Page ComanderApp	<i>Sub-task</i>	<i>High</i>	2
Membuat UI Product Page ComanderApp	<i>Sub-task</i>	<i>High</i>	5
<b>Sprint 2 (16 Maret 2022 – 1 April 2022)</b>	<i>Sprint</i>	<i>High</i>	<b>34</b>
Membuat Use Case ComanderApp	<i>Sub-task</i>	<i>High</i>	8
Membuat Use Case Description Login	<i>Sub-task</i>	<i>High</i>	5
Membuat Use Case Description Keluar	<i>Sub-task</i>	<i>High</i>	
Membuat Use Case Description Pendaftaran Pengguna	<i>Sub-task</i>	<i>High</i>	
Membuat Use Case Description Pendaftaran Merchant	<i>Sub-task</i>	<i>High</i>	

Membuat Use Case Description Menambahkan Barang	<i>Sub-task</i>	<i>High</i>	
Membuat Use Case Description Update Data Barang	<i>Sub-task</i>	<i>High</i>	
Membuat Use Case Description Memproses Pesanan	<i>Sub-task</i>	<i>High</i>	
Membuat Use Case Description Melihat Riwayat Transaksi	<i>Sub-task</i>	<i>High</i>	
Membuat Use Case Description Melihat Data Penjualan	<i>Sub-task</i>	<i>High</i>	
Membuat Use Case Description Pembeli Memesan Pesanan	<i>Sub-task</i>	<i>High</i>	
Membuat Use Case Description Rating	<i>Sub-task</i>	<i>Low</i>	
Membuat Use Case Description Tinjau Merchant	<i>Sub-task</i>	<i>Low</i>	
<b>Sprint 3 (2 April 2022 – 22 April 2022)</b>	<b><i>Sprint</i></b>	<b><i>High</i></b>	28
Membuat Activity Diagram Melakukan Sign Up	<i>Sub-task</i>	<i>High</i>	5
Membuat Activity Diagram Melakukan Login	<i>Sub-task</i>	<i>High</i>	
Membuat Activity Diagram Melihat Status Pesanan	<i>Sub-task</i>	<i>High</i>	
Membuat Activity Diagram Melakukan Isi Ulang Saldo	<i>Sub-task</i>	<i>High</i>	
Membuat Activity Diagram Memilih Merchant	<i>Sub-task</i>	<i>High</i>	
Membuat Activity Diagram Memesan Pesanan	<i>Sub-task</i>	<i>High</i>	
Membuat Activity Diagram Melakukan Logout	<i>Sub-task</i>	<i>High</i>	
Membuat Activity Diagram Memperbarui Data Barang	<i>Sub-task</i>	<i>High</i>	

Membuat Activity Diagram Memproses Pesanan	<i>Sub-task</i>	<i>High</i>	
Membuat Activity Diagram Melihat Data Penjualan	<i>Sub-task</i>	<i>High</i>	
Membuat Activity Diagram Menambahkan Barang	<i>Sub-task</i>	<i>High</i>	
<b>Sprint 4 (23 April 2022 – 16 Mei 2022)</b>	<b><i>Sprint</i></b>	<b><i>High</i></b>	34
Membuat Class Diagram <<view>> Register Page	<i>Sub-task</i>	<i>High</i>	8
Membuat Class Diagram <<view>> Homepage	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<view>> Login Page	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<view>> Profile Page	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<view>> ScanBarcode Page	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<view>> Product Page	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<view>> Transaction Page	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<controller>> UserController	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<controller>> CustomerController	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<controller>> MerchantController	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<model>> User	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<model>> Merchant	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<model>> Customer	<i>Sub-task</i>	<i>High</i>	

Membuat Class Diagram <<model>> Account	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<model>> Transaction	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<model>> TransactionDetail	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<model>> Product	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<model>> ProductType	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<controller>> AdminController	<i>Sub-task</i>	<i>Medium</i>	
Membuat Class Diagram <<view>> OrderDetail Page	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<view>> Management Page	<i>Sub-task</i>	<i>High</i>	
Membuat Class Diagram <<model>> Admin	<i>Sub-task</i>	<i>High</i>	
Coding - Backend			13
Membuat Entity Relationship Diagram	<i>Sub-task</i>	<i>High</i>	8
<b>Sprint 5 (17 Mei 2022 – 8 Juni 2022)</b>	<b>Sprint</b>	<b>High</b>	34
Membuat Sequence Diagram Sign Up	<i>Sub-task</i>	<i>High</i>	8
Membuat Sequence Diagram Login	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Memilih Merchant	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Memesan Barang	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Melakukan Pembayaran	<i>Sub-task</i>	<i>High</i>	

Membuat Sequence Diagram Melihat Status Pemesanan	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Melakukan Isi Ulang Saldo	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Memperbarui Data Barang	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Menambahkan Barang	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Melakukan Update Profile	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Memproses Pesanan	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Melihat Data Penjualan	<i>Sub-task</i>	<i>High</i>	
Membuat Sequence Diagram Block Pembeli/Merchant	<i>Sub-task</i>	<i>Medium</i>	
Membuat Sequence Diagram Logout	<i>Sub-task</i>	<i>High</i>	
Coding - Frontend dan Penyempurnaan	<i>Sub-task</i>	<i>High</i>	13
<b>Sprint 6 (9 Juni 2022 – 29 Juni 2022)</b>	<b><i>Sprint</i></b>	<b><i>High</i></b>	21
Membuat Data Dictionary User	<i>Sub-task</i>	<i>High</i>	2
Membuat Data Dictionary Product	<i>Sub-task</i>	<i>High</i>	
Membuat Data Dictionary ProductType	<i>Sub-task</i>	<i>High</i>	
Membuat Data Dictionary Account	<i>Sub-task</i>	<i>High</i>	
Membuat Data Dictionary Trasaction	<i>Sub-task</i>	<i>High</i>	
Membuat Data Dictionary TrasactionDetail	<i>Sub-task</i>	<i>High</i>	

### 3.1.3 Daily Scrum

*Daily scrum* meeting berdurasi 15 menit dan dilakukan untuk meningkatkan komunikasi, untuk menyinkronkan kegiatan, untuk memusatkan kembali fokus tim pada tujuan bersama dan untuk memecahkan masalah atau hambatan yang mungkin dihadapi tim. Evaluasi akan dilakukan pada tahap ini dimana setiap anggota masing-masing menyampaikan update tiap perkerjaan yang dilakukan setiap hari juga menyampaikan kendala apa saja yang dihadapi selama mengerjakan tugas nya tersebut. *Daily scrum* akan dilakukan via meeting secara online menggunakan google meet atau via chat menggunakan grup whatsapp.

Tabel 3.3 Daily Scrum

Date	Done	To Do	Issue
1 Maret 2022		<ul style="list-style-type: none"> <li>• Mengerjakan sprint planning pertama</li> <li>• Membuat user interface Homepage (Customer)</li> <li>• Membuat user interface Homepage (Merchant)</li> <li>• Membuat Product Page</li> <li>• Membuat ScanBarcode Page</li> </ul>	
4 Maret 2022	<ul style="list-style-type: none"> <li>• Mengerjakan sprint planning pertama</li> <li>• Membuat user interface Homepage (Customer)</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Product Page</li> <li>• Membuat ScanBarcode Page</li> <li>• Membuat Login Page</li> </ul>	

	<ul style="list-style-type: none"> <li>• Membuat user interface Homepage (Merchant)</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Register Page</li> <li>• Membuat Transaction Page</li> <li>• Membuat Menu Page</li> <li>• Membuat TopUp Page</li> </ul>	
5 Maret 2022	<ul style="list-style-type: none"> <li>• Membuat Product Page</li> <li>• Membuat ScanBarcode Page</li> <li>• Membuat Login Page</li> <li>• Membuat Register Page</li> <li>• Membuat Transaction Page</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Menu Page</li> <li>• Membuat TopUp Page</li> <li>• Membuat OrderDetail Page</li> <li>• Membuat Product Backlog</li> </ul>	Melengkapi user interface
9 Maret 2022	<ul style="list-style-type: none"> <li>• Membuat Menu Page</li> <li>• Membuat TopUp Page</li> <li>• Membuat OrderDetail Page</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Product Backlog</li> </ul>	Menentukan backlog tiap sprint
11 Maret 2022	<ul style="list-style-type: none"> <li>• Membuat Product Backlog</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint retrospective pertama</li> <li>• Mengikuti sprint review pertama</li> </ul>	
15 Maret 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint retrospective pertama</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning kedua</li> </ul>	

	<ul style="list-style-type: none"> <li>• Mengikuti sprint review pertama</li> </ul>		
16 Maret 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning kedua</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Usecase Diagram Customer</li> <li>• Membuat Usecase Diagram Merchant</li> <li>• Membuat Usecase Diagram Admin</li> <li>• Membuat Usecase Description Register</li> <li>• Membuat Usecase Diagram Login</li> </ul>	
18 Maret 2022	<ul style="list-style-type: none"> <li>• Membuat Usecase Diagram Customer</li> <li>• Membuat Usecase Diagram Merchant</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Usecase Diagram Admin</li> <li>• Membuat Usecase Description Register</li> <li>• Membuat Usecase Description Login</li> <li>• Membuat Usecase Description Logout</li> </ul>	
22 Maret 2022	<ul style="list-style-type: none"> <li>• Membuat Usecase Diagram Admin</li> <li>• Membuat Usecase Description Register</li> <li>• Membuat Usecase Description Login</li> <li>• Membuat Usecase Description Logout</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Usecase Description tambah barang</li> <li>• Membuat Usecase Description Update barang merchant</li> <li>• Membuat Usecase Description Update Profile</li> <li>• Membuat Usecase Description Memproses pesanan</li> </ul>	

24 Maret 2022	<ul style="list-style-type: none"> <li>• Membuat Usecase Description tambah barang</li> <li>• Membuat Usecase Description Update barang merchant</li> <li>• Membuat Usecase Description Update Profile</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Usecase Description tambah barang</li> <li>• Membuat Usecase Description melihat riwayat transaksi</li> <li>• Membuat Usecase Description melihat data penjualan</li> <li>• Membuat Usecase Description memesan pesanan</li> </ul>	
28 Maret 2022	<ul style="list-style-type: none"> <li>• Membuat Usecase Description Memproses pesanan</li> <li>• Membuat Usecase Description melihat riwayat transaksi</li> <li>• Membuat Usecase Description melihat data penjualan</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Usecase Description memesan pesanan</li> <li>• Membuat Usecase Description rating</li> <li>• Membuat Usecase Description Tinjau Merchant</li> </ul>	
30 Maret 2022	<ul style="list-style-type: none"> <li>• Membuat Usecase Description memesan pesanan</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint review kedua</li> <li>• Mengikuti sprint retrospective kedua</li> </ul>	

	<ul style="list-style-type: none"> <li>• Membuat Usecase Description rating</li> <li>• Membuat Usecase Description Tinjau Merchant</li> </ul>		
1 April 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint review kedua</li> <li>• Mengikuti sprint retrospective kedua</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning ketiga</li> </ul>	
2 April 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning ketiga</li> </ul>	<ul style="list-style-type: none"> <li>• Merancang kuesioner aplikasi untuk Customer</li> </ul>	
5 April 2022	<ul style="list-style-type: none"> <li>• Merancang kuesioner aplikasi untuk Customer</li> </ul>	<ul style="list-style-type: none"> <li>• Merancang kuesioner aplikasi untuk UMKM</li> </ul>	
8 April 2022	<ul style="list-style-type: none"> <li>• Merancang kuesioner aplikasi untuk UMKM</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat analisis aplikasi sejenis</li> <li>• Membuat Activity Diagram Sign Up</li> <li>• Membuat Activity Diagram Login</li> </ul>	
12 April 2022	<ul style="list-style-type: none"> <li>• Membuat analisis aplikasi sejenis</li> <li>• Membuat Activity Diagram Sign Up</li> <li>• Membuat Activity Diagram Login</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Activity Diagram melihat status pesanan</li> <li>• Membuat Activity Diagram isi ulang saldo</li> <li>• Membuat Activity Diagram memilih merchant</li> </ul>	

		<ul style="list-style-type: none"> <li>• Membuat Activity Diagram memesan pesanan</li> </ul>	
15 April 2022	<ul style="list-style-type: none"> <li>• Membuat Activity Diagram melihat status pesanan</li> <li>• Membuat Activity Diagram isi ulang saldo</li> <li>• Membuat Activity Diagram memilih merchant</li> <li>• Membuat Activity Diagram memesan pesanan</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Activity Diagram Logout</li> <li>• Membuat Activity Diagram memperbarui data barang</li> <li>• Membuat Activity Diagram melakukan update profile</li> <li>• Membuat Activity Diagram memproses pesanan</li> <li>• Membuat Activity Diagram melihat data penjualan</li> <li>• Membuat Activity Diagram input barang penjualan</li> </ul>	
18 April 2022	<ul style="list-style-type: none"> <li>• Membuat Activity Diagram Logout</li> <li>• Membuat Activity Diagram memperbarui data barang</li> <li>• Membuat Activity Diagram melakukan update profile</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint review ketiga</li> <li>• Mengikuti sprint retrospective ketiga</li> </ul>	

	<ul style="list-style-type: none"> <li>• Membuat Activity Diagram memproses pesanan</li> <li>• Membuat Activity Diagram melihat data penjualan</li> </ul>		
22 April 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint review ketiga</li> <li>• Mengikuti sprint retrospective ketiga</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning keempat</li> </ul>	
25 April 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning keempat</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat rekap hasil kuesioner</li> <li>• Membuat analisis kebutuhan user berdasarkan kuesioner Cuetomer</li> <li>• Membuat analisis kebutuhan user berdasarkan kuesioner UMKM</li> </ul>	
27 April 2022	<ul style="list-style-type: none"> <li>• Membuat rekap hasil kuesioner</li> <li>• Membuat analisis kebutuhan user berdasarkan kuesioner Cuetomer</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat analisis kebutuhan user berdasarkan kuesioner UMKM</li> <li>• Membuat analisis kebutuhan user berdasarkan observasi</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; RegisterPage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; HomePage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; LoginPage</li> </ul>	

29 April 2022	<ul style="list-style-type: none"> <li>• Membuat analisis kebutuhan user berdasarkan kuesioner UMKM</li> <li>• Membuat analisis kebutuhan user berdasarkan observasi</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; RegisterPage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; HomePage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; LoginPage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; ProfilePage</li> </ul>	
2 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; RegisterPage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; HomePage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; LoginPage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; ProfilePage</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; ScanBarcodePage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; ProductPage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; TransactionPage</li> <li>• Membuat Class Diagram &lt;&lt;controller&gt;&gt; UserController</li> <li>• Membuat Class Diagram &lt;&lt;controller&gt;&gt; CustomerController</li> <li>• Membuat Class Diagram &lt;&lt;controller&gt;&gt; MerchantController</li> <li>• Membuat Class Diagram &lt;&lt;controller&gt;&gt; Admin Controller</li> </ul>	
4 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; ScanBarcodePage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; ProductPage</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; user</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Merchant</li> </ul>	

	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; TransactionPage</li> <li>• Membuat Class Diagram &lt;&lt;controller&gt;&gt; UserController</li> <li>• Membuat Class Diagram &lt;&lt;controller&gt;&gt; CustomerController</li> <li>• Membuat Class Diagram &lt;&lt;controller&gt;&gt; MerchantController</li> <li>• Membuat Class Diagram &lt;&lt;controller&gt;&gt; Admin Controller</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Customer</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Account</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Transaction</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; TransactionDetail</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Product</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; ProductType</li> </ul>	
5 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; user</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Merchant</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Customer</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Account</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Transaction</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; TransactionDetail</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; OrderDetailPage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; ManagementPage</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Admin</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; Product</li> <li>• Membuat Class Diagram &lt;&lt;model&gt;&gt; ProductType</li> </ul>	
6 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; OrderDetailPage</li> <li>• Membuat Class Diagram &lt;&lt;view&gt;&gt; ManagementPage</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat ERD User</li> <li>• Membuat ERD Product</li> <li>• Membuat ERD ProductType</li> </ul>	

	<ul style="list-style-type: none"> <li>• Membuat Class Diagram   &lt;&lt;model&gt;&gt; Admin</li> <li>• Membuat Class Diagram   &lt;&lt;model&gt;&gt; Product</li> <li>• Membuat Class Diagram   &lt;&lt;model&gt;&gt; ProductType</li> </ul>		
10 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat ERD User</li> <li>• Membuat ERD Product</li> <li>• Membuat ERD ProductType</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat ERD Account</li> <li>• Membuat ERD Transaction</li> <li>• Membuat ERD TransactionDetail</li> </ul>	
13 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat ERD Account</li> <li>• Membuat ERD Transaction</li> <li>• Membuat ERD TransactionDetail</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint review keempat</li> <li>• Mengikuti sprint retrospective keempat</li> </ul>	
16 Mei 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint review keempat</li> <li>• Mengikuti sprint retrospective keempat</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning kelima</li> </ul>	
18 Mei 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning kelima</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram Sign Up</li> <li>• Membuat Sequence Diagram Login</li> <li>• Membuat Sequence Diagram Memilih Merchant</li> </ul>	
23 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram Sign Up</li> <li>• Membuat Sequence Diagram Login</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram memesan barang</li> </ul>	

	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram Memilih Merchant</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram melakukan pembayaran</li> <li>• Membuat Sequence Diagram melihat status pemesanan</li> <li>• Membuat Sequence Diagram melakukan pembayaran</li> </ul>	
25 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram memesan barang</li> <li>• Membuat Sequence Diagram melakukan pembayaran</li> <li>• Membuat Sequence Diagram melihat status pemesanan</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram melakukan pembayaran</li> <li>• Membuat Sequence Diagram melakukan isi ulang saldo</li> <li>• Membuat Sequence Diagram memperbarui data barang</li> <li>• Membuat Sequence Diagram menambahkan barang</li> </ul>	
28 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram melakukan pembayaran</li> <li>• Membuat Sequence Diagram melakukan isi ulang saldo</li> <li>• Membuat Sequence Diagram memperbarui data barang</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram menambahkan barang</li> <li>• Membuat Sequence Diagram Update Profile</li> <li>• Membuat Sequence Diagram memproses pesanan</li> <li>• Membuat Sequence Diagram melihat data penjualan</li> </ul>	
30 Mei 2022	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram menambahkan barang</li> <li>• Membuat Sequence Diagram Update Profile</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram melihat data penjualan</li> <li>• Membuat Sequence Diagram block pembeli/merchant</li> </ul>	

	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram memproses pesanan</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram Logout</li> </ul>	
4 Juni 2022	<ul style="list-style-type: none"> <li>• Membuat Sequence Diagram melihat data penjualan</li> <li>• Membuat Sequence Diagram block pembeli/merchant</li> <li>• Membuat Sequence Diagram Logout</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint review keenam</li> <li>• Mengikuti sprint retrospective keenam</li> </ul>	
7 Juni 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint review keenam</li> <li>• Mengikuti sprint retrospective keenam</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning ketujuh</li> </ul>	
9 Juni 2022	<ul style="list-style-type: none"> <li>• Mengikuti sprint planning ketujuh</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Data Dictionary User</li> <li>• Membuat Data Dictionary Product</li> <li>• Membuat Data Dictionary ProductType</li> </ul>	
10 Juni 2022	<ul style="list-style-type: none"> <li>• Membuat Data Dictionary User</li> <li>• Membuat Data Dictionary Product</li> <li>• Membuat Data Dictionary ProductType</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat Data Dictionary Acount</li> <li>• Membuat Data Dictionary Transaction</li> <li>• Membuat Data Dictionary TransactionDetail</li> </ul>	
12 Juni 2022	<ul style="list-style-type: none"> <li>• Membuat Data Dictionary Acount</li> <li>• Membuat Data Dictionary Transaction</li> <li>• Membuat Data Dictionary TransactionDetail</li> </ul>	<ul style="list-style-type: none"> <li>• Mengikuti sprint review ketujuh</li> <li>• Mengikuti sprint retrospective ketujuh</li> </ul>	

### 3.1.4 Sprint Review dan Sprint Retrospective

Pada tahap sprint review para anggota akan mendemonstrasikan apa saja yang sudah diselesaikan dalam periode satu sprint atau setiap satu sprint

selesai. Tujuan dari sprint review adalah untuk mengevaluasi fitur-fitur terbaru dan untuk mempertimbangkan rencana produk kedepannya.

Sprint Restrospective dilakukan setiap sprint berakhir dimana semua anggota bisa menyampaikan pendapat dan evaluasi mengenai kinerja selama menjalani scrum.

Tabel 3.4 Sprint Review dan Sprint Retrospective

<b><i>Sprint</i></b>	<b><i>Sprint Review</i></b>	<b><i>Sprint Retrospective</i></b>
Sprint 1 (1 Maret 2022 - 15 Maret 2022)	<ul style="list-style-type: none"> <li>- Membuat <i>User Interface</i> ComanderApp</li> </ul>	
Sprint 2 (16 Maret 2022 - 1 April 2022)	<ul style="list-style-type: none"> <li>- Membuat <i>Usecase ComanderApp</i></li> <li>- Membuat <i>Usecase Description ComanderApp</i></li> </ul>	
Sprint 3 (2 April 2022 – 22 April 2022)	<ul style="list-style-type: none"> <li>- Membuat <i>Activity Diagram ComanderApp</i></li> </ul>	
Sprint 4 (23 April 2022 – 16 Mei 2022)	<ul style="list-style-type: none"> <li>- Membuat <i>Class Diagram ComanderApp</i></li> <li>- Membuat <i>Entity Relationship Diagram ComanderApp</i></li> <li>- Coding - Backend</li> </ul>	Menyelesaikan error yang dapat menghambat pengembangan aplikasi kedepannya.
Sprint 5 (17 Mei 2022 - 8 Juni 2022)	<ul style="list-style-type: none"> <li>- Membuat <i>Sequence Diagram ComanderApp</i></li> </ul>	Menyesuaikan Sequence Diagram dengan Class Diagram yang sudah

	<ul style="list-style-type: none"> <li>– Coding Backend</li> <li>– Coding - Frontend</li> </ul>	dibuat pada sprint sebelumnya.
Sprint 6 ( 9 Juni 2022 - 29 Juni 2022)	<ul style="list-style-type: none"> <li>– Membuat Data Dictionary ComanderApp.</li> <li>– Coding Frontend dan Penyempurnaan</li> </ul>	Melakukan update progres 2x setiap minggunya sampai waktu deadline yang ditentukan

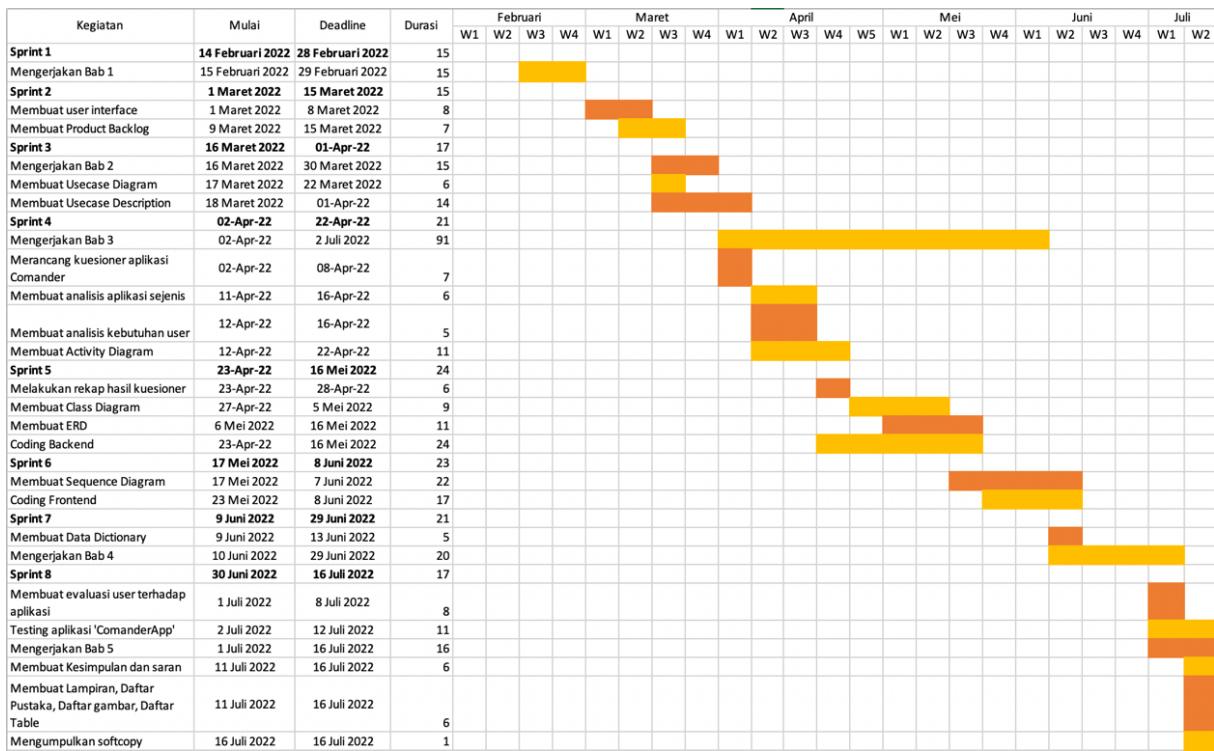
### 3.1.5 Pembagian Tugas

Berdasarkan teori Scrum oleh Ken Schwaber & Jeff Sutherland dalam Scrum team harus terdiri dari minimal Product Owner, Scrum Master dan Developer. Dalam pengerjaan skripsi ini, berikut pembagian role dalam Scrum Team:

- Fahrul Rozi Arsyandra (Product Owner, Scrum master, Developer)
- Annisa Octaviana Nurshanty (Product Owner, Developer)

### 3.1.6 Gantt Chart

Gantt chart berfungsi sebagai informasi mengenai tugas tim, jadwal, dan batas waktu pengembangan aplikasi.



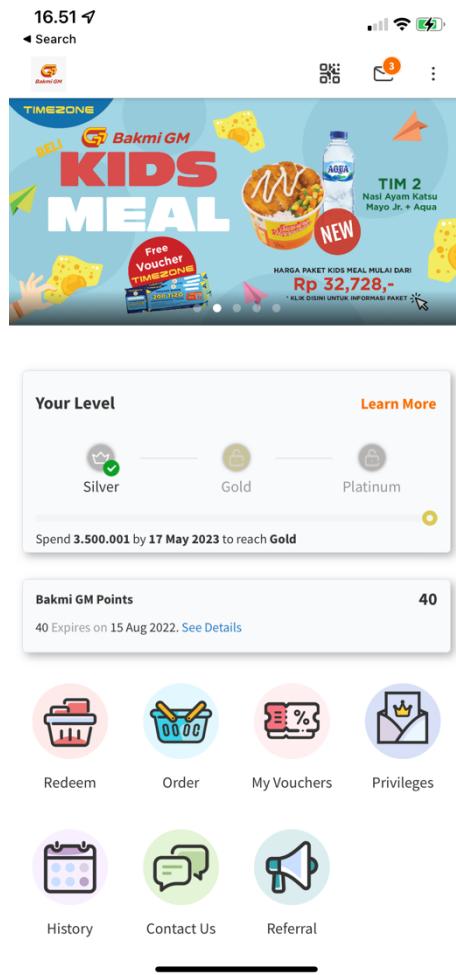
Gambar 3.1 Gantt Chart

## 3.2 Analisis

### 3.2.1 Analisis Aplikasi Sejenis

Untuk membandingkan Comander App dengan aplikasi mobile sejenis, kami melakukan penelitian dengan mencoba menggunakan aplikasi sejenis. Dimulai dari alur penggunaan aplikasi (pendaftaran, alur transaksi dan sejenisnya) sampai dengan fitur-fitur unggulan yang masing-masing situs aplikasi miliki. Kami melakukan penelitian di 3 aplikasi serupa, yaitu

### 3.2.1.1 Bakmi GM



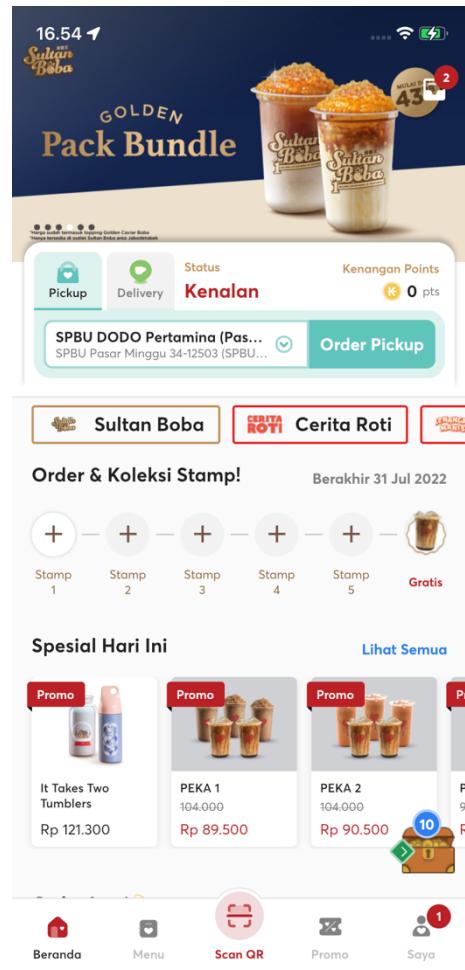
Gambar 3.2 Aplikasi Bakmi GM

PT GRIYA MIESEJATI adalah induk perusahaan restoran cepat saji BAKMI GM, yang merupakan ikon restoran bakmi di Jakarta. Restoran yang dulunya bernama BAKMI GAJAH MADA, didirikan pada tahun 1959 oleh sepasang suami istri Bapak Tjhai Sioe dan Ibu Loei Kwai Fong. Store cabang pertama hadiri di Melawai pada tahun 1971. Kini Bakmi GM memiliki outlet-outlet yang tersebar di Jabodetabek, Bandung dan Surabaya. Dengan pengalaman lebih dari 50 tahun di industri restoran, BAKMI GM saat ini melayani lebih dari 30.000 pelanggan per hari ([bakmigm.com](http://bakmigm.com)).

Bakmi GM memiliki sebuah fitur dimana saat customer melakukan scan barcode yang terdapat pada toko maka customer akan langsung masuk kedalam aplikasi untuk melihat

menu-menu yang terdapat di Bakmi GM, dari aplikasi tersebut customer juga dapat melakukan order.

### 3.2.1.2 Kopi Kenangan



Gambar 3.3 Aplikasi Kopi Kenangan

Kopi kenangan merupakan merupakan perusahaan yang memiliki rantai kopi grab-and-go yang tumbuh paling cepat di Indonesia ([kopikenangan.com](http://kopikenangan.com)). Kopi kenangan memiliki aplikasi kopi kenangan yang memiliki fitur self pick up. Fitur ini memungkinkan pengguna memesan produknya melalui aplikasi mobile tanpa perlu memesan ke kasir. Setelah pesanan tersebut dilakukan, pengguna dapat mengambilnya di outlet yang telah dipilih oleh pengguna. Hal ini berfungsi agar pengguna dapat langsung mengambil tanpa perlu mengantri.

Tabel 3.5 Perbandingan Aplikasi Sejenis

Perbandingan	Bakmi GM	Kopi Kenangan	ComanderApp
Jenis Platform	Web & Android	Android	Android
Menampilkan daftar produk merchant	v	v	v
Dapat melakukan transaksi	v	v	v
Menampilkan transaksi pembelian	v	v	v
Dapat melakukan pembayaran	x	v	v
Dapat melakukan top-up saldo	x	x	v

### 3.2.2 Analisis Kebutuhan User

#### 3.2.2.1 Analisis User Berdasarkan Kuesioner

Salah satu metode yang digunakan dalam proses pengumpulan data dan keperluan analisis kebutuhan user adalah melalui survey kuisoner. Berikut adalah pertanyaan dan alasan dari pertanyaan dari kuesioner yang digunakan.

Tabel 3.6 Pertanyaan dan Alasan Pertanyaan Kuesioner

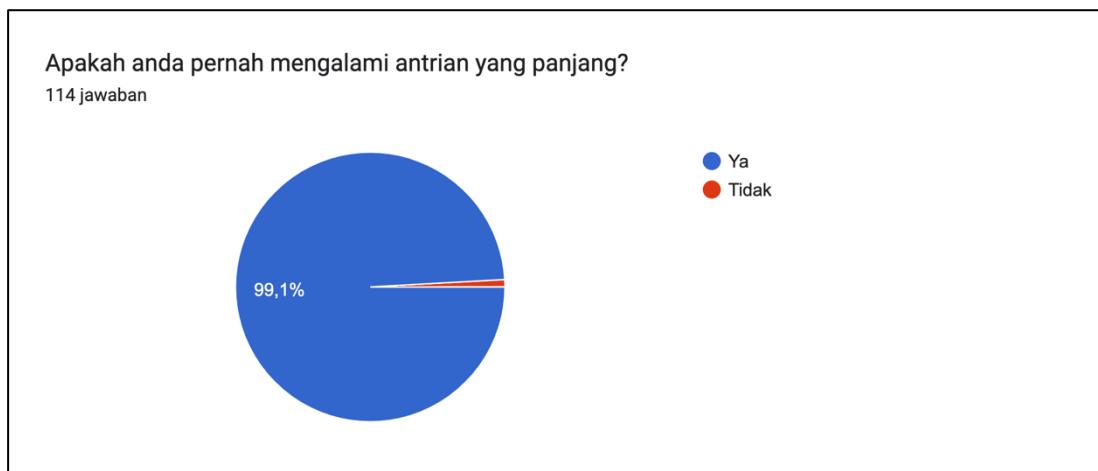
Pertanyaan	Alasan Pertanyaan
Apakah anda pernah mengalami antrian yang Panjang?	Untuk mengetahui apakah responden pernah mengalami antrian yang panjang

Bagaimana situasi antrian yang pernah anda alami?	Untuk mengetahui situasi antrian yang pernah dialami oleh responden
Apakah terdapat perbedaan antara antrian saat pandemi dengan normal?	Untuk mengetahui apakah terdapat perbedaan antrian saat pandemi dengan normal
Apakah antrian lebih lama saat pandemic?	Untuk mengetahui apakah antrian saat pandemi lebih lama atau tidak
Apakah anda pernah menggunakan layanan self service seperti di McDonald, Bakmi GM dan Kopi Kenangan?	Untuk mengetahui apakah responden pernah menggunakan aplikasi <i>self service</i> seperti di McDonald, Bakmi GM dan Kopi Kenangan
Apakah layanan tersebut membantu anda mengurangi antrian?	Untuk mengetahui apakah layanan <i>self service</i> tersebut dapat mengurangi antrian
Bagaimana tingkat pelayanan yang diberikan?	Untuk mengetahui tingkat pelayanan yang diberikan oleh layanan <i>self service</i> diatas
Apakah layanan tersebut dapat berguna apabila diterapkan di setiap rumah makan atau toko-toko lainnya?	Untuk mengetahui apakah layanan <i>self service</i> tersebut dapat berguna apabila diterapkan di rumah makan atau toko-toko lainnya
Apakah layanan tersebut dapat meningkatkan social-distancing selama masa ppkm?	Untuk mengetahui apakah layanan tersebut dapat meningkatkan <i>social distancing</i> selama masa ppkm berlangsung
Apakah layanan tersebut tetap efektif diterapkan apabila keadaan sudah embali normal?	Untuk mengetahui apakah layanan <i>self service</i> tersebut tetap efektif diterapkan apabila keadaan sudah kembali normal

Bagaimana pandangan kalian apabila terdapat suatu aplikasi yang memudahkan kalian dalam memesan hanya dengan melakukan scan barcode dan melakukan transaksi dalam aplikasi tersebut sehingga kalian tidak perlu berdiri didepan kasir?

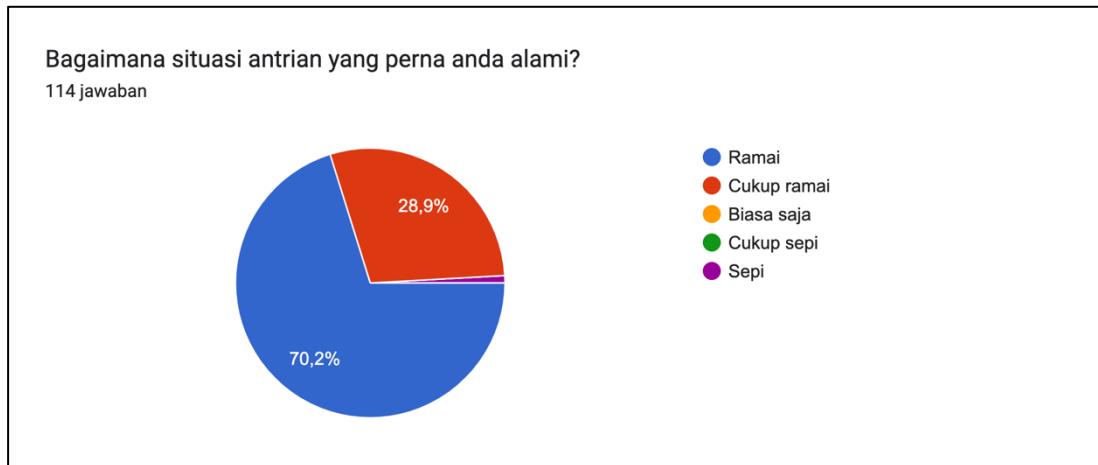
Apakah anda akan menggunakan Untuk mengetahui ketertarikan aplikasi yang menerapkan sistem (*self service*) tersebut?

Berikut adalah hasil survey dari pertanyaan-pertanyaan diatas dan telah diisi oleh 114 responden:



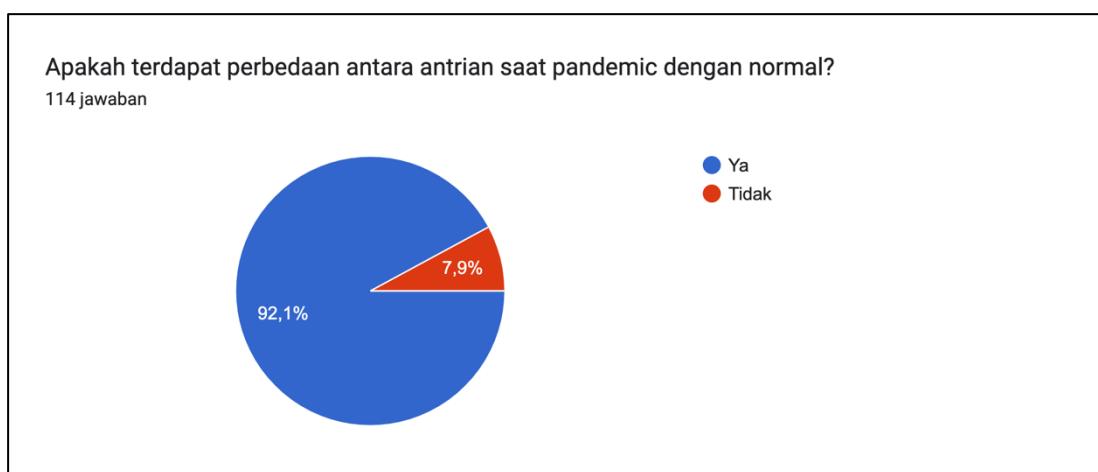
Gambar 3.4 Diagram Pertanyaan 1

Pertanyaan No. 1, “Apakah anda pernah mengalami antrian yang Panjang?”. Dari 100% responden sebanyak 99.1% menjawab “Ya” dan sebanyak 0.9% menjawab “Tidak”. Kesimpulan, lebih banyak responden pernah mengalami antrian yang Panjang.



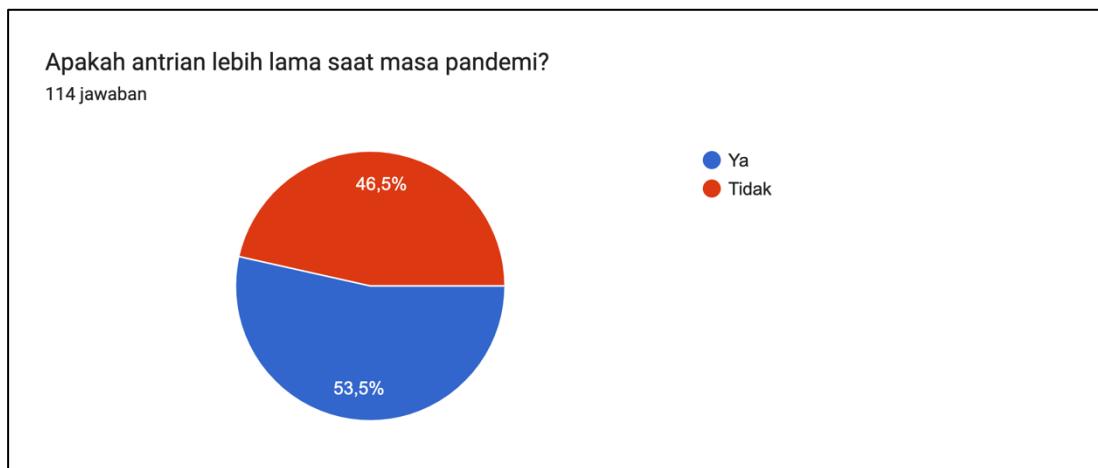
Gambar 3.5 Diagram Pertanyaan 2

Pertanyaan No. 2, “Bagaimana situasi antrian yang pernah anda alami?”. Dari 100% responden sebanyak 70.2% menjawab “Ramai”, sebanyak 28.9% menjawab “Cukup ramai” dan sebanyak 0.9% menjawab “Sepi”. Kesimpulan, lebih banyak responden mengalami situasi antrian yang Ramai.



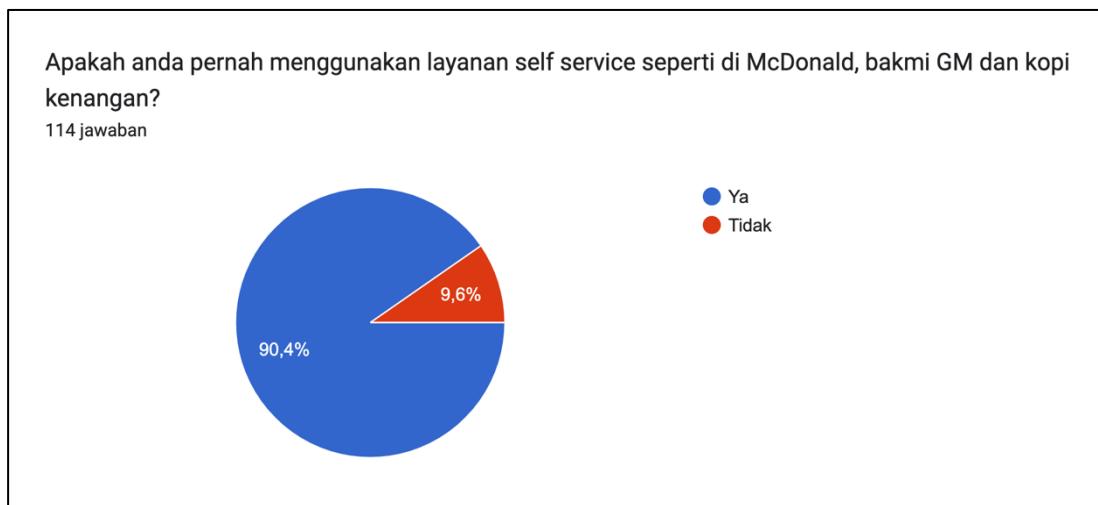
Gambar 3.6 Diagram Pertanyaan 3

Pertanyaan No. 3, “Apakah terdapat perbedaan antara antrian saat pandemic dengan normal?”. Dari 100% responden, sebanyak 92.1% menjawab “Ya”, sebanyak 7.9% menjawab “Tidak”. Kesimpulan, lebih banyak responden menyatakan terdapat perbedaan antara antrian saat pandemic dengan normal.



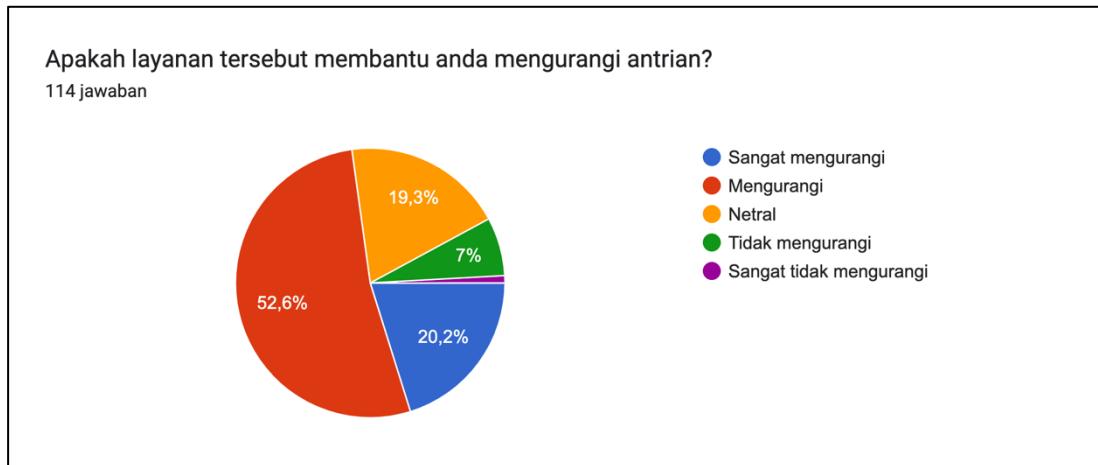
Gambar 3.7 Diagram Pertanyaan 4

Pertanyaan No. 4, “Apakah antrian lebih lama saat masa pandemic?”. Dari 100% responden sebanyak 53.5% menjawab “Ya”, sebanyak 46.5% menjawab “Tidak”. Kesimpulan, lebih banyak responden menyatakan antrian lebih lama saat masa pandemic.



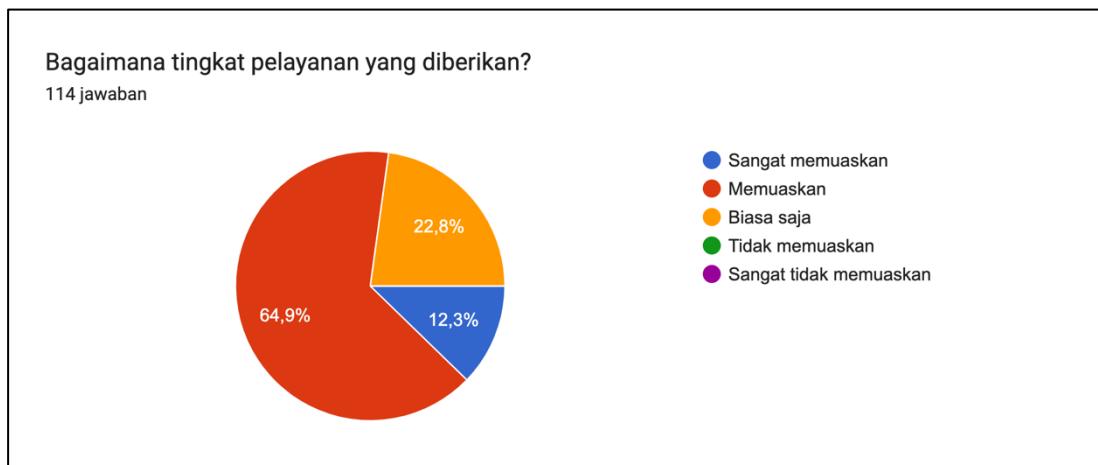
Gambar 3.8 Diagram Pertanyaan 5

Pertanyaan No. 5, “Apakah anda pernah menggunakan layanan *self service* seperti di McDonald, Bakmi GM dan Kopi Kenangan?”. Dari 100% responden sebanyak 90.4% menjawab “Ya” dan sebanyak 9.6% menjawab “Tidak”. Kesimpulan, lebih banyak responden pernah menggunakan layanan *self service* seperti di McDonald, Bakmi GM dan Kopi Kenangan.



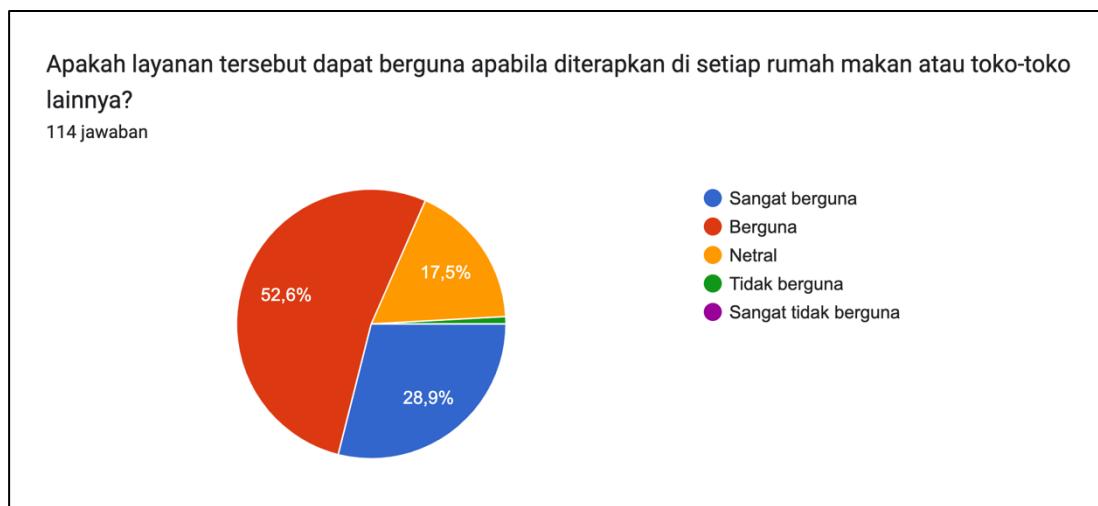
Gambar 3.9 Diagram Pertanyaan 6

Pertanyaan No. 6, “Apakah layanan tersebut membantu anda mengurangi antrian?”. Dari 100% responden sebanyak 52.6% menjawab “Mengurangi”, sebanyak 20.2% menjawab “Sangat mengurangi”, sebanyak 19.3% menjawab “Netral”, sebanyak 7% menjawab “Tidak mengurangi”, sebanyak 0.9% menjawab “Sangat tidak mengurangi”. Kesimpulan, lebih banyak responden menyatakan layanan *self service* tersebut membant untuk mengurangi antrian.



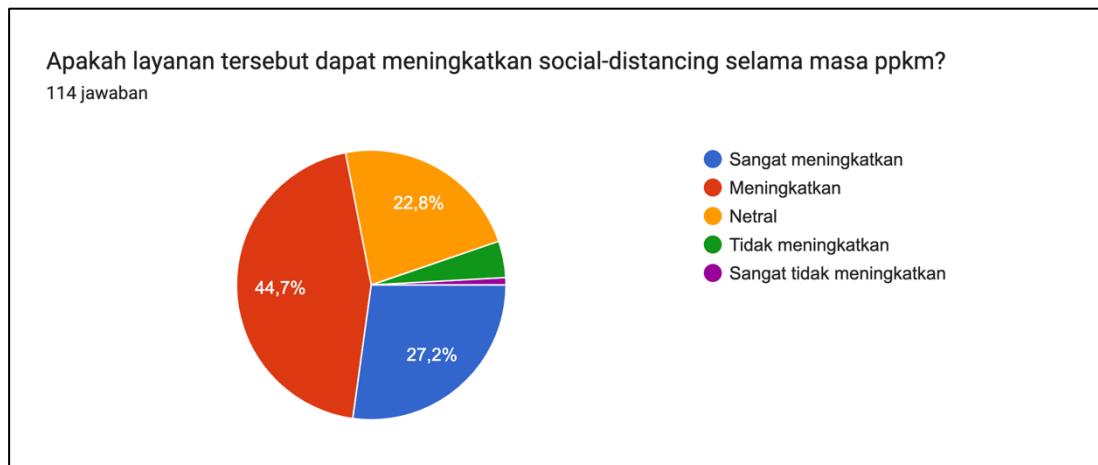
Gambar 3.10 Diagram Pertanyaan 7

Pertanyaan No. 7, “Bagaimana tingkat pelayanan yang diberikan?”. Dari 100% responden, sebanyak 64.9% menjawab “Memuaskan”, sebanyak 22.8% menjawab “Biasa saja”, sebanyak 12.3% menjawab “Sangat Memuaskan”. Kesimpulan, lebih banyak responden menyatakan tingkat pelayanan yang diberikan Memuaskan.



Gambar 3.11 Diagram Pertanyaan 8

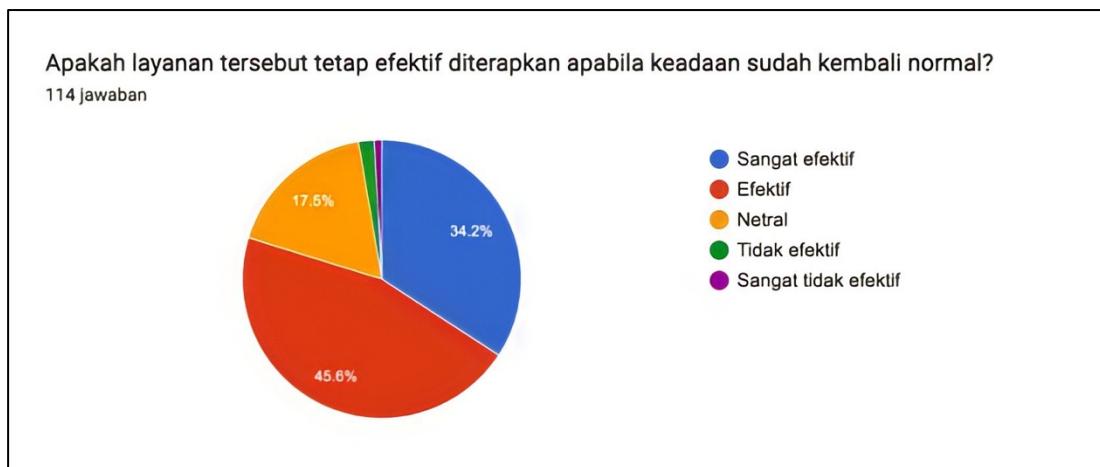
Pertanyaan No. 8, “Apakah layanan tersebut dapat berguna apabila diterapkan disetiap rumah makan atau toko-toko lainnya?”. Dari 100% responden, sebanyak 52.6% menjawab “Berguna”, sebanyak 28.9% menjawab “Sangat berguna”, sebanyak 17.5% menjawab “Netral” dan sebanyak 0.9% menjawab “Tidak berguna”. Kesimpulan, lebih banyak responden menyatakan layanan tersebut Berguna apa bila diterapkan di setiap rumah makan atau toko-toko lainnya.



Gambar 3.12 Diagram Pertanyaan 9

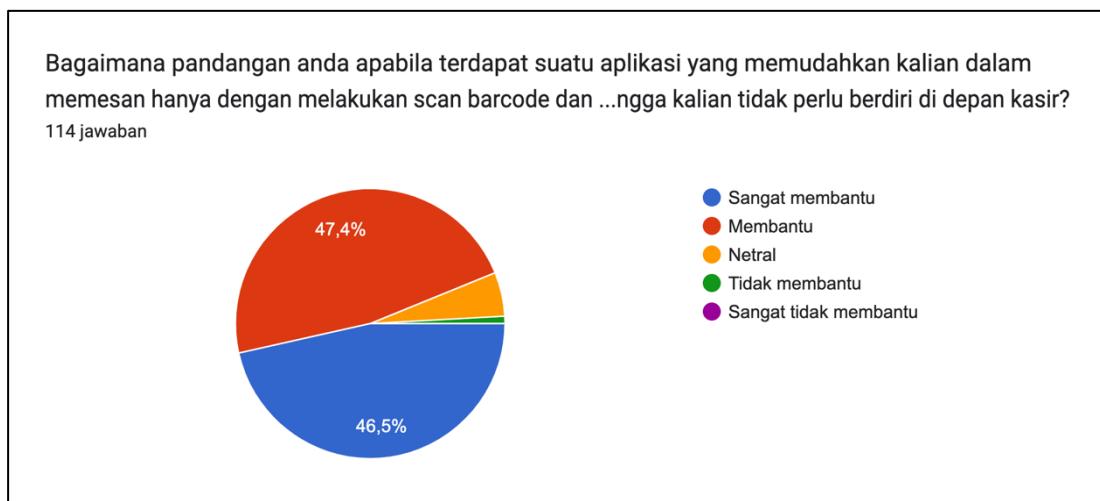
Pertanyaan No. 9, “Apakah layanan tersebut dapat meningkatkan *social-distancing* selama masa ppkm?”. Dari 100% responden, sebanyak

44.7% menjawab “Meningkatkan”, sebanyak 27.2% menjawab “Sangat meningkatkan”, sebanyak 22.8% menjawab “Netral”, sebanyak 4.4% menjawab “Tidak meningkatkan” dan sebanyak 0.9% menjawab “Sangat Tidak Meningkatkan”. Kesimpulan, lebih banyak responden berpendapat layanan tersebut Meningkatkan *social-distancing* selama masa ppkm.



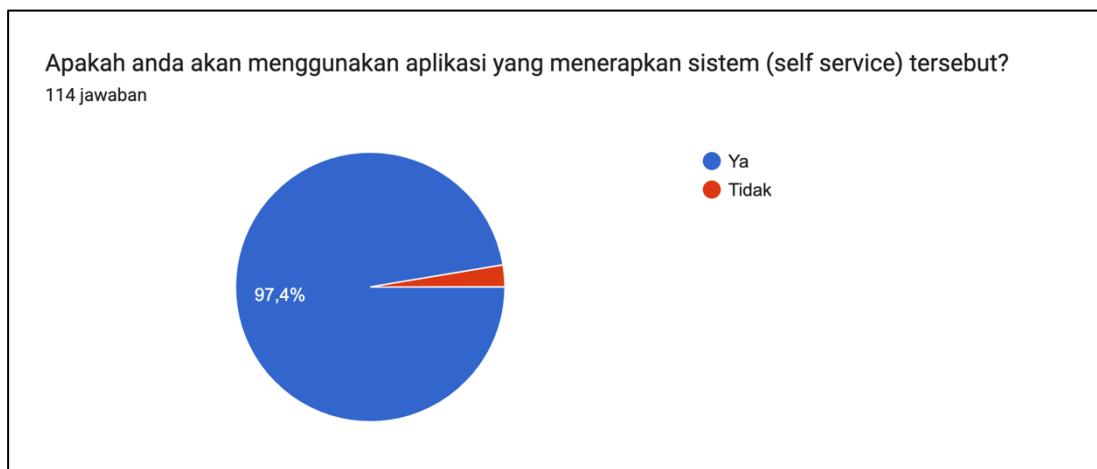
Gambar 3.13 Diagram Pertanyaan 10

Pertanyaan No.10, “Apakah layanan tersebut tetap efektif diterapkan apabila keadaan sudah Kembali normal?”. Dari 100% responden, sebanyak 45.6% menjawab “Efektif”, sebanyak 34.2% menjawab “Sangat efektif”, sebanyak 17.5% menjawab “Netral”, sebanyak 1.8% menjawab “Tidak efektif” dan sebanyak 0.9% menjawab “Sangat tidak efektif”. Kesimpulan, lebih banyak responden berpendapat layanan tersebut Efektif diterapkan apabila keadaan sudah kembali normal.



Gambar 3.14 Diagram Pertanyaan 11

Pertanyaan No. 11, “Bagaimana pandangan anda apabila terdapat suatu aplikasi yang memudahkan kalian dalam memesan hanya dengan melakukan scan barcode dan melakukan transaksi dalam aplikasi tersebut sehingga kalian tidak perlu berdiri didepan kasir?”. Dari 100% responden sebanyak 47.7% menjawab “Sangat membantu”, sebanyak 46.5% menjawab “Membantu”, sebanyak 5.3% menjawab “Netral” dan sebanyak 0.9% menjawab “Tidak membantu”. Kesimpulan, lebih banyak responden berpendapat aplikasi tersebut Sangat membantu.



Gambar 3.15 Diagram Pertanyaan 12

Pertanyaan No. 12, “Apakah anda akan menggunakan aplikasi yang menerapkan sistem (self service) tersebut?”. Dari 100% responden, sebanyak 97.4% menjawab “Ya” dan sebanyak 2.6% menjawab “Tidak”. Kesimpulan, lebih banyak responden tertarik menggunakan aplikasi yang menerapkan sistem *self service*.

### 3.2.2.2 Analisis User Berdasarkan Observasi

Penulis menerapkan jenis observasi partisipan, yaitu penulis mengikuti proses pemesanan yang ada pada 3 aplikasi sejenis yang memiliki fitur self-service. hal ini bertujuan agar penulis memperoleh kesimpulan seberapa berpengaruh proses aplikasi pemesanan melalui aplikasi mobile tersebut.

Pada aplikasi Bakmi GM, penulis melakukan scan barcode yang tedapat di setiap meja pada restoran tersebut. Setalah memesan, penulis menuju kasir untuk melakukan pembayaran. Penulis melakukan antrian selama 5 menit. Lalu melakukan pembayaran selama kurang lebih 2 menit dan pesanan sudah di proses.

Pada aplikasi Kopi Kenangan. Penulis memilih lokasi outlet terdekat yang dapat melakukan fitur pickup. Setalah itu penulis melakukan pemesanan. Setelah itu penulis menuju ke lokasi outlet pemesanan. Penulis menunggu selama 5 menit dan pesanan dapat diambil.

Kesimpulannya, metode pemesanan tanpa aplikasi mobile lebih memakan waktu lebih banyak dibandingkan dengan pemesanan melalui aplikasi mobile.

### 3.2.2.3 Analisis User Berdasarkan Wawancara

Selain melakukan pengumpulan data dengan menggunakan metode kuesioner dan observasi, penulis juga melakukan pengumpulan data menggunakan metode wawancara kepada pemilik UMKM. Berikut adalah pertanyaan dan alasan pertanyaan yang diajukan oleh penulis ke narasumber.

Tabel 3.7 Pertanyaan dan Alasan Pertanyaan Wawancara

Pertanyaan	Alasan Pertanyaan
Sudah berapa lama anda menjalani usaha?	Untuk mengetahui berapa lama narasumber menjalani usaha
Bagaimana cara anda selama ini mencatat transaksi penjualan yang berlangsung?	Untuk mengetahui cara narasumber mencatat transaksi penjualan yang berlangsung
Apakah selama menjalani usaha anda mengalami kesulitan dalam melihat laporan penjualan dan transaksi penjualan?	Untuk mengetahui kesulitan yang dialami narasumber dalam melihat laporan penjualan dan transaksi penjualan

Apakah anda membutuhkan laporan penjualan selama menjalani usaha?	Untuk mengetahui apakah narasumber membutuhkan laporan penjualan selama menjalani usaha
Data apa saja yang anda perlukan dalam membuat laporan penjualan?	Untuk mengetahui data apa yang diperlukan narasumber dalam membuat laporan penjualan
Apakah anda perlu mengetahui laba kotor, penjualan kotor, penjualan bersih, total transaksi, dan margin kotor?	Untuk mengetahui apakah narasumber memrlukan data-data tersebut dalam laporan penjualan
Bagaimana cara anda selama ini membuat laporan penjualan?	Untuk mengetahui cara narasumner membuat laporan penjualan
Apabila terdapat suatu aplikasi yang dapat mencatat seluruh transaksi yang dilakukan oleh cutstomer dan dapat menampilkan analisis penjualan apakah aplikasi tersebut dapat membantu usaha anda?	Untuk mengetahui ketertarikan narasumber terhadap aplikasi yang akan dikembangkan

Berikut adalah hasil dari pertanyaan wawancara yang diajukan kepada narasumber, nsrasumber terdiri dari 2 orang yaitu Bapak Zikri selaku pemilik UMKM Noisy Time Clothing dan Bapak Yoga Zanades selaku .

1. Sudah berapa lama anda menjalani usaha?

Zikri : Saya menjalani usaha ini dari tahun 2015 dan selama pandemi juga masih terus berjalan karena kami cuma ada offline doang mba belum bisa online. Saya juga punya 4 toko di Jakarta ada 3 sama di serang 1.

Yoga : saya sudah menjalani usaha dari 2012

2. Bagaimana cara anda selama ini mencatat transaksi penjualan yang berlangsung?

Zikri : Selama ini saya cuma catet manual aja pake buku ini sehari-hari barang yang kejual apa saja

Yoga : Kalo mencatatnya berdasarkan pembayaran invoice lunas dan belum lunas, dari yang lunas kita pisahkan untuk mengetahui uang masuk sama keluar

3. Apakah selama menjalani usaha anda mengalami kesulitan dalam melihat laporan penjualan dan transaksi penjualan?

Zikri : Sulit mba, karna karyawan-karyawan saya juga suka ga bener kerjanya jadi banyak barang yang ilang ada juga yang ga kecatet. Saya juga pernah minta orang buat bisa masukin data-data barang ke sistem tapi orangnya minta mahal mba 150 juta untuk satu toko sama computer sama CPU.

Yoga : karena sudah memiliki sistem sendiri, kesulitan itu tidak ada. Namun perhitungan yang tidak tepat ada. Karena saat terdapat promosi dapat error dan data input secara manual ke excel.

4. Apakah anda membutuhkan laporan penjualan selama menjalani usaha?

Zikri : Butuh mba, soalnya saya juga pernah kena di salah satu toko, barang saja dimaling sama salah satu karyawan sekitar 100 lebih.

Yoga : pasti diperlukan untuk menghitung uang masuk, uang keluar dan uang piutang

5. Data apa saja yang anda perlukan dalam membuat laporan penjualan?

Zikri : yang penting kuantiti barang mba buat saya, berapa barang yang kejual sehari berapa barang yang masuk soalnya disini susah mba buat jualan harga bandrol, sudah dikasi harga tetap masih ditawar

Yoga : laporan itu diperlukan, laporan asset, laporan pajak, dan laporan keuangan,

6. Apakah anda perlu mengetahui laba kotor, penjualan kotor, penjualan bersih, total transaksi, dan margin kotor?

Zikri : Perlu sih mba itu bisa sangat membantu saya lagi

Yoga : perlu karena kan acuannya di akhir bulan untuk menghitung stok kita, berapa keuntungannya, dan selisih itu perlu.

7. Bagaimana cara anda selama ini membuat laporan penjualan?

Zikri : Masih manual aja, soalnya saya sudah pernah minta dibikinin sistem tapi terlalu mahal mba

Yoga : untuk saat ini cara membuat laporan masih input secara manual ke dalam excel, namun perhitungannya sudah ada

8. Apabila terdapat suatu aplikasi yang dapat mencatat seluruh transaksi yang dilakukan oleh customer dan dapat menampilkan analisis penjualan apakah aplikasi tersebut dapat membantu usaha anda?

Zikri: Sangat membantu sih itu, jadi saya bisa tenang ninggalin toko kapanpun gaperlu khawatir barang kecuri sama karyawan, saya bisa sambil mantau toko juga jadinya

Yoga : kalo dalam usaha, iya sangat membantu

### 3.2.3 Usulan Pemecahan Masalah

Setelah melakukan analisis dari beberapa aplikasi sejenis, observasi dan menyebarkan kuesioner ke masyarakat umum mengenai situasi antrian saat

pandemi dengan normal dan pengaruh aplikasi *self-service*, kami juga menyebarkan kuesioner kepada UMKM mengenai kesultanan mengelola usaha. Kami memberikan solusi dengan membuat aplikasi yang memiliki sistem self-service untuk menghubungkan customer dengan UMKM dan membuat fitur reporting untuk memudahkan UMKM mengelola usahanya.

Penulis dapat mengimplementasikan dari observasi, kuesioner dan wawancara kedalam fitur-fitur sebagai berikut:

1. Membuat aplikasi self-service via *smartphone* untuk mengurangi antrian
2. Membuat fitur scan barcode untuk menampilkan data merchant
3. Mencatat semua transaksi yang dilakukan
4. Membuat fitur reporting untuk melihat laporan penjualan merchant

### 3.3 Perancangan Aplikasi

#### 3.3.1 Software Design Documentation

##### 3.3.1.1 Deskripsi *Software*

ComanderApp merupakan sebuah aplikasi *self service* berbasis *mobile application* yang dapat memfasilitasi calon customer untuk melakukan transaksi secara mandiri. Pada aplikasi ini terdapat 3 role user yang berbeda yaitu Customer, Merchant dan Admin. Melalui aplikasi ini Customer dan Merchant dapat saling terhubung. Customer dapat memilih dan melihat data merchant melalui fitur Scan Barcode dan langsung melakukan transaksi melalui aplikasi dan Merchant dapat memproses transaksi yang dilakukan oleh Customer. Selain itu Merchant juga dapat melihat *reporting* penjualan untuk membantu dan mempermudah mengelola usaha yang dimilikinya.

##### 3.3.1.2 Fungsi-Fungsi Software

Sebagai aplikasi *self-service* dengan fitur *reporting* yang kami rancang yaitu “ComanderApp” terdapat beberapa fungsi dan kegunaan yang dapat digunakan. Berikut penjelasan mengenai fungsi-fungsi yang dimiliki oleh aplikasi ComanderApp:

a. Register

User baru dapat melakukan registrasi untuk mendaftarkan akun dengan role Customer atau Merchant agar dapat mengakses fitur-fitur yang terdapat dalam aplikasi

b. Login

User yang sudah terdaftar dapat masuk kedalam aplikasi dengan melakukan login

c. Scan barcode

User dengan role customer dapat mengakses fitur scan barcode untuk melihat data merchant dan daftar produk yang dijualnya

d. Top up

User dengan role customer dapat melakukan top up sebagai metode pembayaran pada aplikasi

e. Checkout

Customer yang sudah menambahkan barang kedalam cart dapat melakukan transaksi dengan menggunakan fitur *checkout*.

f. Reporting

*Reporting* merupakan salah satu fitur utama aplikasi ComanderApp dimana user dengan role *merchant* dapat melihat rangkuman laporan penjualan seperti penjualan kotor, penjualan bersih, laba kotor, margin kotor dan total transaksi.

g. View Product

Customer dan merchant dapat melihat daftar produk yang dimiliki oleh *merchant*.

h. Export excel

Melalui fitur ini, merchant dapat mengunduh laporan penjualan dalam bentuk file excel.

i. Add Product

Fitur ini terdapat pada halaman manage dimana *merchant* dapat menambahkan data produk baru dengan menggunakan fitur add.

j. Edit Product

Selain fitur add product pada halaman manage, merchant juga dapat mengubah data produk menggunakan fitur edit

k. Delete Product

Selain menambahkan dan mengubah data produk, merchant juga dapat menghapus data produk dari database menggunakan fitur delete.

### 3.3.1.3 Kebutuhan Teknologi

Dalam perancangan dan pengembangan aplikasi “ComanderApp” ini, ada beberapa teknologi yang diperlukan antara lain; Laravel, Flutter, Mysql, Vscode.

#### 1. *Database*

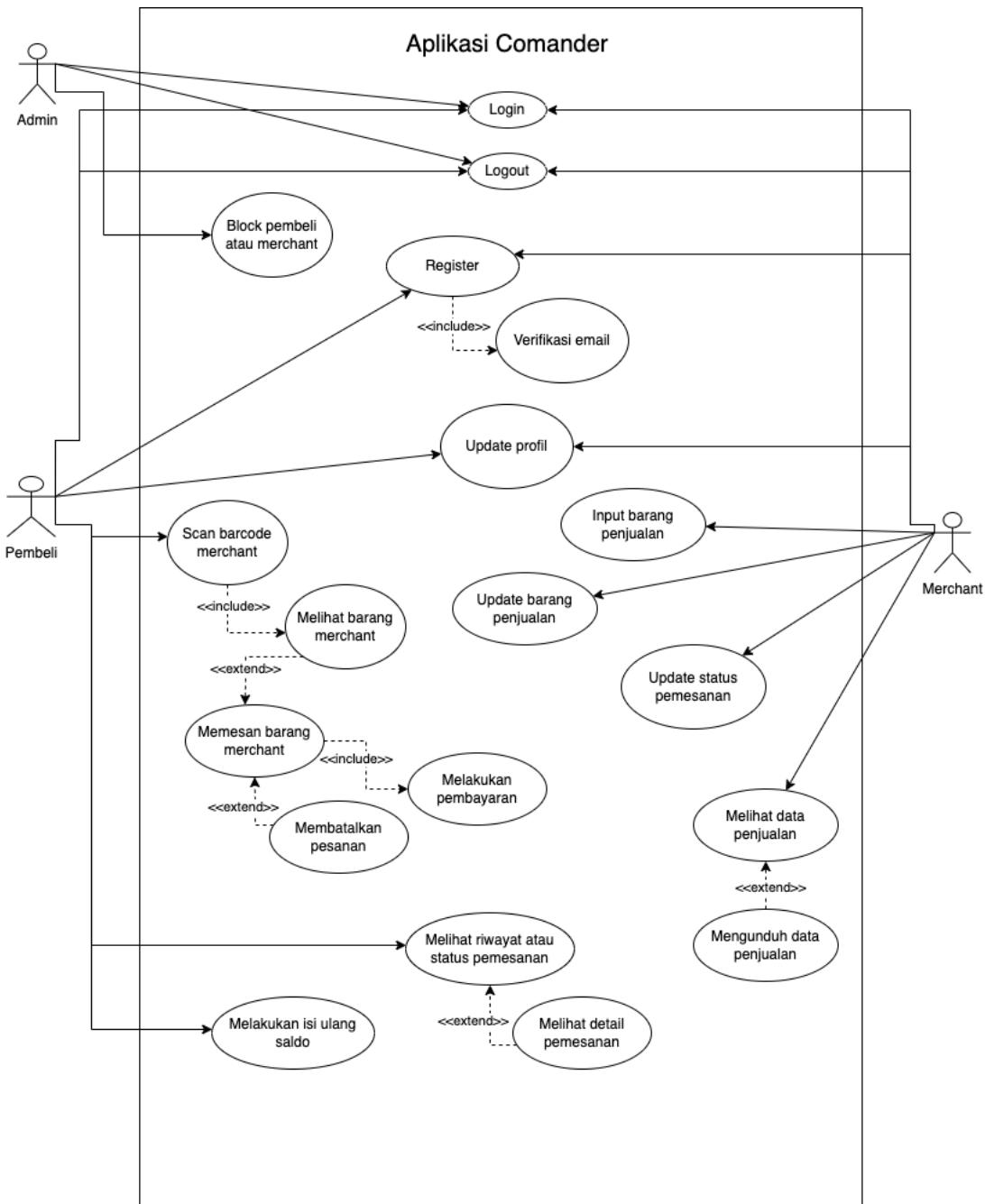
Basis data yang digunakan dalam mengembangkan aplikasi adalah MySQL sebagai DBMS (*Database Management System*) untuk mempermudah dalam memanipulasi data.

#### 2. *Programming Language*

Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan framework Laravel untuk *backend* dan Dart dengan framework Flutter untuk frontend. Untuk menghubungkan antara fronted dengan backend akan digunakan *Application Programming Interface (API)*

### 3.3.2 Perancangan Sistem

#### 3.3.2.1 Use Case Diagram



Gambar 3.16 Use Case Diagram

### 3.3.2.2 Use Case Description

Table 3.8 Use Case Description Login

<i>Use Case Name : Login</i>	<i>ID : 1</i>	<i>Importance Level : High</i>		
<i>Primary Actor : Pembeli, Penjual, Admin</i>	<i>Use Case Type : Detail, Essential</i>			
<p>Stakeholders and Interest :</p> <ul style="list-style-type: none"> <li>• Pembeli <i>login</i> ke dalam aplikasi menggunakan akun</li> <li>• Toko <i>login</i> ke dalam aplikasi menggunakan akun</li> <li>• Admin <i>login</i> ke situs admin.comander.com</li> </ul>				
<p>Brief Description :</p> <p>Proses yang dilakukan oleh para Actor untuk <i>login</i> ke aplikasi menggunakan akun</p>				
<p>Trigger :</p> <ul style="list-style-type: none"> <li>• Pembeli atau penjual menekan tombol <i>login</i> pada aplikasi</li> <li>• Admin mengakses situs admin.comander.com</li> </ul>				
<p>Type : <i>External</i></p>				
<p>Relationships :</p> <p><i>Relationship</i> : Pembeli, Penjual, Admin</p> <p><i>Include</i> : -</p> <p><i>Extend</i> : -</p> <p><i>Generalization</i> : -</p>				
<p>Normal Flow of Events :</p> <ol style="list-style-type: none"> <li>1. <i>Actor</i> membuka halaman masuk pada aplikasi comander. Khusus admin menggunakan situs admin.comander.com</li> <li>2. <i>Actor</i> menekan tombol masuk</li> <li>3. Sistem akan menerapkan validasi data yang telah diinput oleh actor</li> <li>4. Sistem akan mengecek role pengguna</li> <li>5. Jika sebagai pengguna, sistem akan mengarahkan ke halaman utama pengguna aplikasi. Jika sebagai toko, sistem akan mengarahkan ke halaman utama toko. Dan jika sebagai admin, sistem akan mengarahkan ke halaman utama admin.</li> </ol>				
<p>SubFlows : -</p>				

Alternative/Exceptional Flows :
---------------------------------

- |  |
|--|
| 3a. apabila data tidak sesuai, maka sistem memunculkan pesan “email atau password salah” |
|--|

Table 3.9 Use Case Description Keluar

<i>Use Case Name : Logout</i>	<i>ID : 2</i>	<i>Importance Level : High</i>		
<i>Primary Actor : Pembeli, Penjual, Admin</i>	<i>Use Case Type : Detail, Essential</i>			
Stakeholders and Interest :				
<ul style="list-style-type: none"> <li>Pembeli ingin melakukan <i>logout</i></li> </ul>				
Brief Description :				
Proses yang dilakukan oleh para Actor untuk mengakhiri <i>session</i>				
Trigger :				
<ul style="list-style-type: none"> <li>Pembeli atau Penjual menekan tombol keluar atau <i>logout</i> pada halaman profile</li> <li>Admin menekan tombol keluar atau <i>logout</i> pada navbar</li> </ul>				
Type : <i>External</i>				
Relationships :				
<i>Relationship</i> : Pembeli, Penjual, Admin <i>Include</i> : - <i>Extend</i> : - <i>Generalization</i> : -				
Normal Flow of Events :				
<ol style="list-style-type: none"> <li>Actor menekan tombol <i>logout</i></li> <li>Sistem akan mengatur ulang semua <i>session</i> dari actor</li> <li>Sistem akan mengarahkan ke halaman masuk</li> </ol>				
SubFlows : -				
Alternative/Exceptional Flows : -				

Table 3.10 Use Case Description Pendaftaran Pengguna

<i>Use Case Name</i> : Daftar Sebagai pengguna	<i>ID</i> : 3	<i>Importance Level</i> : High		
<i>Primary Actor</i> : Pembeli	<i>Use Case Type</i> : Detail, Essential			
Stakeholders and Interest :				
<ul style="list-style-type: none"> <li>Pembeli ingin mendaftar atau membuat akun</li> </ul>				
Brief Description :				
Proses yang dilakukan oleh para Actor pembeli untuk membuat akun baru				
Trigger :				
<ul style="list-style-type: none"> <li>Pembeli menekan tombol daftar sebagai pengguna</li> </ul>				
Type : <i>External</i>				
Relationships :				
<i>Relationship</i> : Pembeli <i>Include</i> : - <i>Extend</i> : - <i>Generalization</i> : -				
Normal Flow of Events :				
<ol style="list-style-type: none"> <li>Actor menekan tombol daftar sebagai pengguna</li> <li>Sistem akan mengarahkan ke halaman pendaftaran pengguna</li> <li>Actor memasukkan nama, email, kata sandi, konfirmasi kata sandi dan nomor telepon dan mencentang kotak syarat dan persetujuan ketentuan dan kondisi aplikasi commander</li> <li>Sistem akan memeriksa dan melakukan validasi pada data yang telah diinput oleh actor</li> </ol> <p>Sistem akan mengarahkan ke halaman utama pengguna</p>				
SubFlows : -				
Alternative/Exceptional Flows :				
3a. apabila email sudah pernah dipakai oleh pengguna atau toko terdahulu, maka sistem akan memberikan pemberitahuan dan actor harus menggunakan email yang berbeda 3b. apabila konfirmasi kata sandi tidak sesuai dengan kata sandi sebelumnya, maka actor harus menginput ulang kata sandi hingga sesuai				

Table 3.11 Use Case Daftar sebagai merchant

<i>Use Case Name</i> : Daftar Sebagai <i>Merchant</i>	<i>ID</i> : 4	Importance Level : High
<i>Primary Actor</i> : <i>Merchant</i>	<i>Use Case Type</i> : <i>Detail, Essential</i>	
<b>Stakeholders and Interest :</b>		
<ul style="list-style-type: none"> <li>• Penjual ingin mendaftar atau membuat akun</li> </ul>		
<b>Brief Description :</b>		
Proses yang dilakukan oleh para Actor penjual untuk membuat akun baru		
<b>Trigger :</b>		
<ul style="list-style-type: none"> <li>• Penjual menekan tombol daftar sebagai <i>Merchant</i></li> </ul>		
<b>Type</b> : <i>External</i>		
<b>Relationships :</b>		
<i>Relationship</i> : Penjual <i>Include</i> : - <i>Extend</i> : - <i>Generalization</i> : -		
<b>Normal Flow of Events :</b>		
<ol style="list-style-type: none"> <li>1. Actor menekan tombol daftar sebagai <i>Merchant</i></li> <li>2. Sistem akan mengarahkan ke halaman pendaftaran <i>Merchant</i></li> <li>3. Actor memasukkan nama, email, kata sandi, konfirmasi kata sandi, nomor telepon, alamat, dan mencentang kotak syarat dan persetujuan ketentuan dan kondisi aplikasi commander</li> <li>4. Sistem akan memeriksa dan melakukan validasi pada data yang telah diinput oleh actor</li> </ol> <p>Sistem akan mengarahkan ke halaman utama pengguna</p>		
<b>SubFlows</b> : -		
<b>Alternative/Exceptional Flows :</b>		
3a. apabila email sudah pernah dipakai oleh pengguna atau <i>Merchant</i> terdahulu, maka sistem akan memberikan pemberitahuan dan actor harus menggunakan email yang berbeda		
3b. apabila konfirmasi kata sandi tidak sesuai dengan kata sandi sebelumnya, maka actor harus menginput ulang kata sandi hingga sesuai		

Table 3.12 Use Case Tambah Barang

<i>Use Case Name</i> : Menambah barang penjualan	<i>ID</i> : 5	Importance Level : High		
<i>Primary Actor</i> : Merchant	<i>Use Case Type</i> : Detail, Essential			
Stakeholders and Interest :				
<ul style="list-style-type: none"> <li>• <i>Merchant</i> ingin menambahkan data barang penjualan</li> </ul>				
Brief Description :				
Proses yang dilakukan oleh para Actor <i>Merchant</i> untuk menambahkan barang penjualan				
Trigger :				
<ul style="list-style-type: none"> <li>• <i>Merchant</i> menekan tombol tambah barang</li> </ul>				
Type : <i>External</i>				
Relationships :				
<i>Relationship</i> : <i>Merchant</i> <i>Include</i> : - <i>Extend</i> : - <i>Generalization</i> : -				
Normal Flow of Events :				
<ol style="list-style-type: none"> <li>1. Actor menekan tombol tambah barang pada halaman daftar barang penjualan</li> <li>2. Actor memasukkan nama barang, jenis barang, harga pokok, harga jual, deskripsi barang, gambar barang.</li> <li>3. User menekan tombol submit</li> <li>4. Sistem akan menyimpan data barang yang telah diinput</li> </ol>				
SubFlows : -				
Alternative/Exceptional Flows : -				

Tabel 3.13 Use Case Description Update Barang Merchant

<i>Use Case Name</i> : Update data barang <i>merchant</i>	<i>ID</i> : 6	Importance Level : High
<i>Primary Actor</i> : Merchant	<i>Use Case Type</i> : Detail, Essential	

Stakeholders and Interest :
<ul style="list-style-type: none"> <li>• Merchant ingin update barang</li> </ul>
Brief Description :
Proses yang dilakukan oleh para Actor Merchant untuk menyunting barang penjualan
Trigger :
<ul style="list-style-type: none"> <li>• Merchant menekan tombol update pada card barang</li> </ul>
Type : External
Relationships :
<p>Relationship : Merchant</p> <p>Include : -</p> <p>Extend : membatalkan update</p> <p>Generalization : -</p>
Normal Flow of Events :
<ol style="list-style-type: none"> <li>1. Actor menekan tombol update</li> <li>2. Sistem menampilkan data barang</li> <li>3. Actor memperbarui data yang barang yang ingin diperbarui</li> <li>4. Actor menekan tombol submit</li> <li>5. Sistem memperbarui data barang</li> </ol>
SubFlows : -
Alternative/Exceptional Flows :
3a. apabila actor menekan tombol cancel saat update barang, maka update barang gagal dan sistem mengarahkan ke halaman daftar barang

Tabel 3.14 Use Case Description Update Profil

Use Case Name : update profil	ID : 7	Importance Level : Medium		
Primary Actor : Merchant, Pembeli	Use Case Type : Detail, Essential			
Stakeholders and Interest :				
<ul style="list-style-type: none"> <li>• Pembeli atau merchant ingin update profil</li> </ul>				

<p>Brief Description :</p> <p>Proses yang dilakukan oleh para Actor pembeli untuk mengedit profil</p>
<p>Trigger :</p> <ul style="list-style-type: none"> <li>• Pembeli menekan tombol <i>update</i> profil pada halaman profil</li> </ul> <p>Type : <i>External</i></p>
<p>Relationships :</p> <p><i>Relationship</i> : <i>Merchant</i>, pembeli</p> <p><i>Include</i> : -</p> <p><i>Extend</i> :</p> <ul style="list-style-type: none"> <li>- <i>update</i> foto</li> <li>- <i>update</i> data profil</li> <li>- ubah kata sandi</li> </ul> <p><i>Generalization</i> : -</p>
<p>Normal Flow of Events :</p> <ol style="list-style-type: none"> <li>1. actor menekan edit profil</li> <li>2. sistem mengarahkan ke halaman edit profil</li> <li>3. actor melakukan edit foto apabila actor ingin mengunggah foto     S-1 unggah foto Apabila actor ingin menghapus foto     S-2 Hapus foto</li> <li>4. actor melakukan edit data profil <ul style="list-style-type: none"> <li>- Pembeli Actor mengisi data pembeli</li> <li>- <i>Merchant</i> Actor mengisi data <i>merchant</i></li> </ul> Apabila actor ingin mengubah kata sandi     S-3 ubah kata sandi </li> </ol>
<p>SubFlows :</p> <p>S-1. Unggah foto profil</p> <ol style="list-style-type: none"> <li>1. Actor menekan tombol ubah foto profil</li> </ol>

	<ol style="list-style-type: none"> <li>2. Sistem menampilkan dialog edit foto</li> <li>3. Actor menekan tombol unggah foto</li> <li>4. Actor memilih foto yang ingin diunggah dari penyimpanan lokal</li> <li>5. Sistem akan menampilkan hasil foto yang diunggah pengguna</li> <li>6. Actor menekan tombol unggah foto</li> <li>7. Sistem menampilkan dialog update berhasil</li> </ol>
S-2.	<p>Hapus foto profil</p> <ol style="list-style-type: none"> <li>1. Actor menekan tombol hapus foto</li> <li>2. Sistem menampilkan dialog untuk memastikan apakah actor benar ingin menghapus foto</li> <li>3. Actor menekan tombol hapus</li> <li>4. Sistem menampilkan dialog hapus foto berhasil</li> </ol>
S-3.	Alternative/Exceptional Flows : -

Table 3.15 Use Case Description Merchant Memperoses Pesanan

<i>Use Case Name : merchant memproses pesanan</i>	<i>ID : 8</i>	<i>Importance Level : High</i>		
<i>Primary Actor : Merchant, pembeli</i>	<i>Use Case Type : Detail, Essential</i>			
Stakeholders and Interest :				
<ul style="list-style-type: none"> <li>• <i>Merchant</i> memproses dan mengubah status pesanan</li> </ul>				
Brief Description :				
Proses yang dilakukan oleh para Actor <i>Merchant</i> untuk memproses pesanan				
Trigger :				
<ul style="list-style-type: none"> <li>• <i>Merchant</i> menekan card pesanan</li> </ul>				
Type : <i>External</i>				
Relationships :				
<i>Relationship : Merchant, pembeli</i>				
<i>Include : -</i>				

*Extend :*

- mengubah status pemesanan
- update pesanan

*Generalization : -*

Normal Flow of Events :

1. Actor merchant menerima pesanan
2. Actor menekan card pesanan
3. Sistem mengarahkan ke halaman detail pesanan
4. Actor *merchant* memproses pesanan

Apabila pesanan sedang di proses atau telah selesai di proses

S-1. Memproses pesanan

Apabila pesanan sedang di proses atau telah selesai di proses

S-2. Mengakhiri proses pesanan

Apabila salah satu pesanan habis

S-3. Update pesanan

SubFlows :

S-1. Memproses pesanan

1. merchant mengubah status pesanan menjadi sedang di proses
2. sistem memperbarui status pemesanan
3. sistem mengirim notifikasi ke pembeli

S-2. Mengakhiri proses pesanan

1. merchant mengubah status pesanan menjadi selesai di proses
2. sistem memperbarui status pemesanan
3. sistem mengirim notifikasi ke pembeli dan menampilkan dialog rating

S-3. Update pesanan

1. merchant meminta pembeli untuk update pesanan
2. sistem mengirimkan pesan merchant ke pembeli
3. pembeli menyetujui permintaan merchant memperbarui pesanan

<p>4. pembeli menekan tombol submit</p> <p>5. sistem memperbarui pesanan</p> <p style="padding-left: 40px;">sistem mengirimkan pesanan terbaru ke <i>merchant</i></p>
Alternative/Exceptional Flows :
<p>S-3, 3a. apabila pembeli menolak</p> <ol style="list-style-type: none"> <li>1. sistem menampilkan dialog untuk memastikan apakah pembeli ingin membatalkan pesanan</li> <li>2. sistem memperbarui status pesanan dibatalkan</li> </ol>

Table 3.16 Use Case Description Melihat Riwayat Transaksi

<i>Use Case Name</i> : Melihat Riwayat transaksi	<i>ID</i> : 9	<i>Importance Level</i> : High		
<i>Primary Actor</i> : <i>Merchant</i> , pembeli	<i>Use Case Type</i> : Detail, Essential			
Stakeholders and Interest :				
<ul style="list-style-type: none"> <li>• <i>Merchant</i> ingin melihat Riwayat transaksi barang</li> <li>• Pembeli ingin melihat Riwayat transaksi barang</li> </ul>				
Brief Description :				
Proses yang dilakukan oleh para Actor untuk melihat Riwayat transaksi				
Trigger :				
<ul style="list-style-type: none"> <li>• <i>Merchant</i> menekan tombol <i>update</i> pada <i>card</i> barang</li> </ul>				
Type : <i>External</i>				
Relationships :				
<i>Relationship</i> : <i>Merchant</i> , pembeli <i>Include</i> : - <i>Extend</i> : - <i>Generalization</i> : -				
Normal Flow of Events :				
<ol style="list-style-type: none"> <li>1. Actor menekan tombol Transaksi pada <i>bottom navigation bar</i> aplikasi</li> <li>2. Sistem mengarahkan ke halaman utama transaksi</li> </ol>				

3. Sistem menampilkan transaksi yang sedang berlangsung, yang sudah selesai, atau yang gagal.
SubFlows : -
Alternative/Exceptional Flows :
3a. apabila actor menekan salah satu card transaksi <ul style="list-style-type: none"> <li>- Sistem mengarahkan ke halaman detail transaksi</li> </ul>

Table 3.17 Use Case Description Melihat Data Penjualan

<i>Use Case Name</i> : Melihat data penjualan	<i>ID</i> : 10	<i>Importance Level</i> : High		
<i>Primary Actor</i> : Merchant	<i>Use Case Type</i> : Detail, Essential			
Stakeholders and Interest :				
<ul style="list-style-type: none"> <li>• Merchant ingin melihat data penjualan</li> </ul>				
Brief Description :				
Proses yang dilakukan oleh para Actor untuk melihat data penjualan				
Trigger :				
<ul style="list-style-type: none"> <li>• Merchant menekan tombol data penjualan pada halaman penjualan</li> </ul>				
Type : External				
Relationships :				
<i>Relationship</i> : Merchant <i>Include</i> : - <i>Extend</i> : mengunduh data <i>Generalization</i> : -				
Normal Flow of Events :				
<ol style="list-style-type: none"> <li>1. Actor menekan tombol data penjualan</li> <li>2. Actor memilih rentang waktu penjualan</li> <li>3. Actor menekan tombol tampilkan</li> <li>4. Sistem menampilkan data penjualan</li> <li>5. Actor melihat data penjualan</li> </ol>				

Apabila actor ingin mengunduh S-1. Unduh data penjualan
SubFlows : S-2. Unduh data penjualan 1. Actor menekan tombol unduh 2. Actor menentukan tempat penyimpanan data pada penyimpanan lokal 3. Sistem mengirim data penjualan
Alternative/Exceptional Flows : -

Table 3.18 Use Case Description Pembeli memesan pesanan

<i>Use Case Name</i> : pembeli memesan pesanan	<i>ID</i> : 11	<i>Importance Level</i> : High		
<i>Primary Actor</i> : pembeli	<i>Use Case Type</i> : Detail, Essential			
Stakeholders and Interest :				
<ul style="list-style-type: none"> <li>Pembeli ingin memesan makan</li> </ul>				
Brief Description : Proses yang dilakukan oleh para Actor pembeli untuk memesan				
Trigger : <ul style="list-style-type: none"> <li>Pembeli melakukan scan barcode</li> </ul>				
Type : <i>External</i>				
Relationships :  <i>Relationship</i> : pembeli <i>Include</i> : - <i>Extend</i> : <ul style="list-style-type: none"> <li>Menambah atau mengurangi pesanan</li> </ul> <i>Generalization</i> : -				
Normal Flow of Events : <ol style="list-style-type: none"> <li>Actor menekan tombol pesan</li> </ol>				

<p>2. Sistem mengarahkan ke halaman detail pesanan</p> <p>3. Actor memeriksa pesanan</p> <p>Apabila actor ingin menambah pesanan</p> <p>S-1. Tambahkan pesanan</p> <p>Apabila actor ingin mengurangi pesanan</p> <p>S-2. Mengurangi pesanan</p> <p>4. Actor memilih metode pembayaran</p> <p>5. Actor menekan tombol bayar</p> <p>6. Sistem menyimpan data dan memberikan notifikasi pesanan masuk pada merchant</p>
<p>SubFlows :</p> <p>S-1. Tambahkan pesanan</p> <ol style="list-style-type: none"> <li>1. Actor menekan tombol + pada card produk</li> <li>2. Sistem menambahkan jumlah produk</li> </ol> <p>S-2. Mengurangi pesanan</p> <ol style="list-style-type: none"> <li>1. Actor menekan tombol –</li> <li>2. Sistem mengurangi jumlah pesanan</li> </ol> <p>Apabila kuantitas produk lebih dari 1</p> <ul style="list-style-type: none"> <li>– Sistem mengurangi kuantitas</li> </ul> <p>Apabila jumlah produk sama dengan 1</p> <ul style="list-style-type: none"> <li>– Sistem menghapus produk tersebut</li> </ul> <p>Apabila tidak ada lagi data produk sama sekali pada detail pesanan</p> <ul style="list-style-type: none"> <li>– Sistem mengarahkan ke halaman menu</li> </ul>
<p>Alternative/Exceptional Flows :</p> <p>3a. apabila actor ingin menambah pesanan baru, actor menekan <i>text button add more items</i></p>

Table 3.19 Use Case Description Rating

<i>Use Case Name : Rating</i>	<i>ID : 12</i>	<i>Importance Level : High</i>		
<i>Primary Actor : Pembeli</i>	<i>Use Case Type : Detail, Essential</i>			
<i>Stakeholders and Interest :</i>				
<ul style="list-style-type: none"> <li>• Pemebil ingin memberi rating <i>merchant</i></li> </ul>				

Brief Description :
Proses yang dilakukan oleh Actor untuk memberi rating <i>merchant</i>
Trigger :
<ul style="list-style-type: none"> <li>Actor menekan tombol bintang pada dialog rating</li> </ul>
Type : <i>External</i>
Relationships :
<p><i>Relationship</i> : Pembeli</p> <p><i>Include</i> : -</p> <p><i>Extend</i> : -</p> <p><i>Generalization</i> : -</p>
Normal Flow of Events :
<ol style="list-style-type: none"> <li>Actor menekan tombol bintang sesuai dengan jumlah bintang yang ingin diberikan</li> <li>Actor memberikan ulasan</li> <li>Sistem menyimpan ulasan</li> </ol>
SubFlows : -
Alternative/Exceptional Flows : -

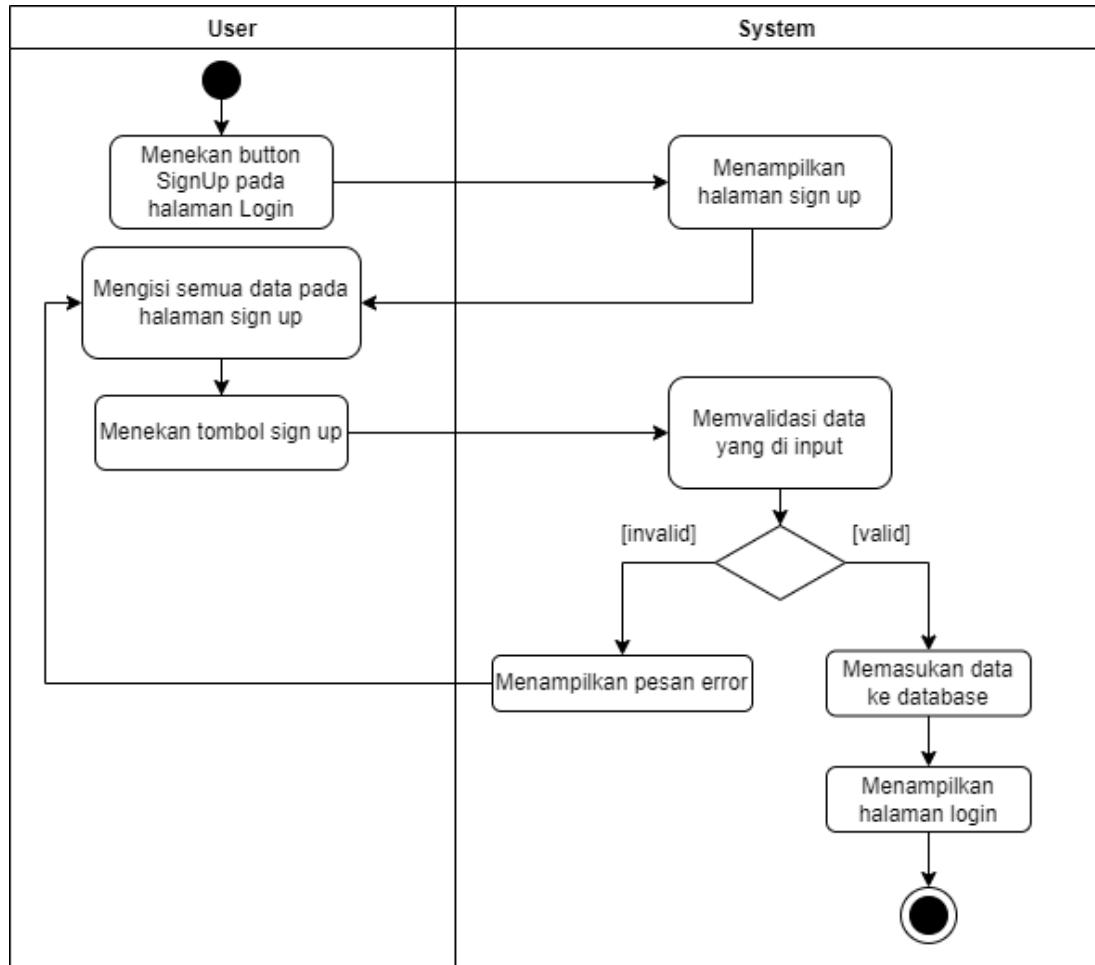
Table 3.20 Use Case Description Tinjau Merchant

<i>Use Case Name</i> : Tinjau Merchant	<i>ID</i> : 13	Importance Level : High
<i>Primary Actor</i> : Admin	<i>Use Case Type</i> : Detail, Essential	
Stakeholders and Interest :		
<ul style="list-style-type: none"> <li>Admin ingin meninjau <i>merchant</i></li> </ul>		
Brief Description :		
Proses yang dilakukan oleh Actor untuk meninjau <i>merchant</i>		
Trigger :		
<ul style="list-style-type: none"> <li>Admin menekan tombol tinjau pada card merchant</li> </ul>		

Type : <i>External</i>
<p>Relationships :</p> <p><i>Relationship</i> : Admin</p> <p><i>Include</i> : -</p> <p><i>Extend</i> : block merchant</p> <p><i>Generalization</i> : -</p>
<p>Normal Flow of Events :</p> <ol style="list-style-type: none"> <li>4. Actor menekan tombol tinjau merchant pada card <i>merchant</i></li> <li>5. Sistem mengarahkan ke halaman detail tinjau <i>merchant</i></li> <li>6. Actor melakukan peninjauan terhadap <i>merchant</i></li> </ol> <p>Apabila actor menerima</p> <p>S-1. Block merchant</p> <p>Apabila actor menolak</p> <p>S-2. Menolak tinjauan</p>
<p>SubFlows :</p> <ol style="list-style-type: none"> <li>1. Block merchant             <ol style="list-style-type: none"> <li>1. Actor menekan tombol block</li> <li>2. Actor memberikan alasan.</li> <li>3. Sistem menangguhkan merchant</li> </ol> </li> <li>S-2. Menolak tinjauan             <ol style="list-style-type: none"> <li>1. Actor menekan tombol tolak</li> <li>2. Sistem menghapus data merchant yang ditinjau pada data tinjauan</li> </ol> </li> </ol>
Alternative/Exceptional Flows : -

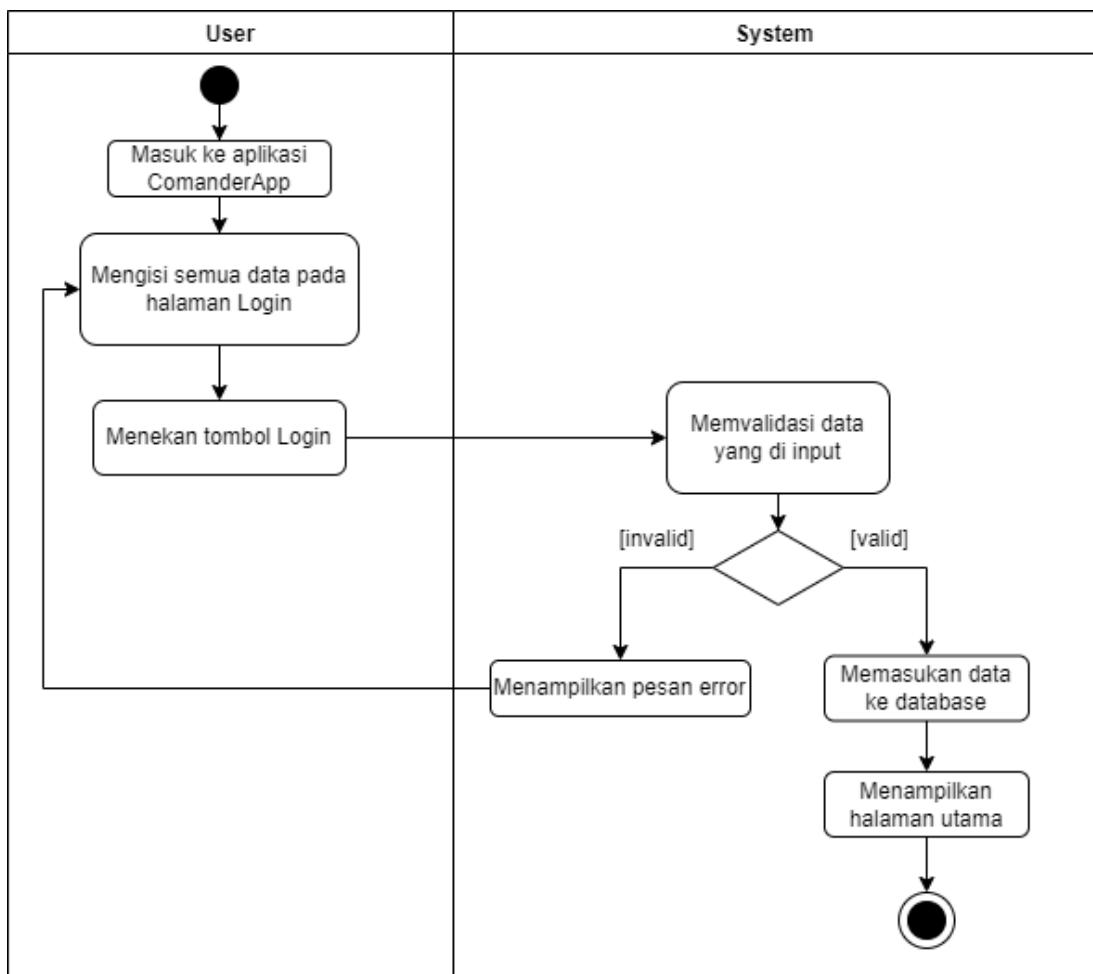
### 3.3.2.3 Activity Diagram

#### 3.3.2.3.1 Activity Diagram Melakukan Sign up



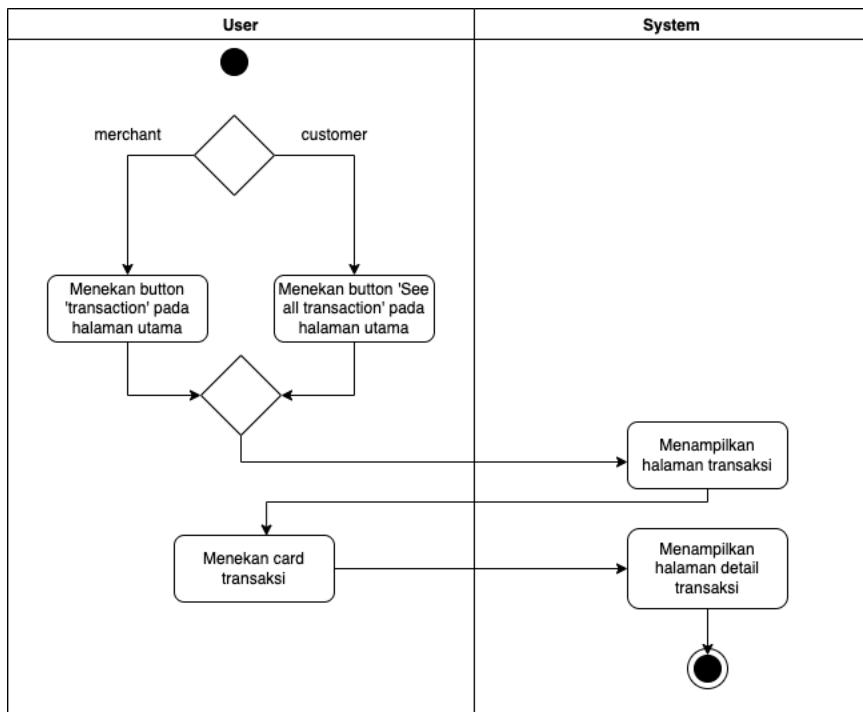
Gambar 3.17 Activity Diagram melakukan sign up

### 3.3.2.3.2 Activity Diagram Melakukan Login



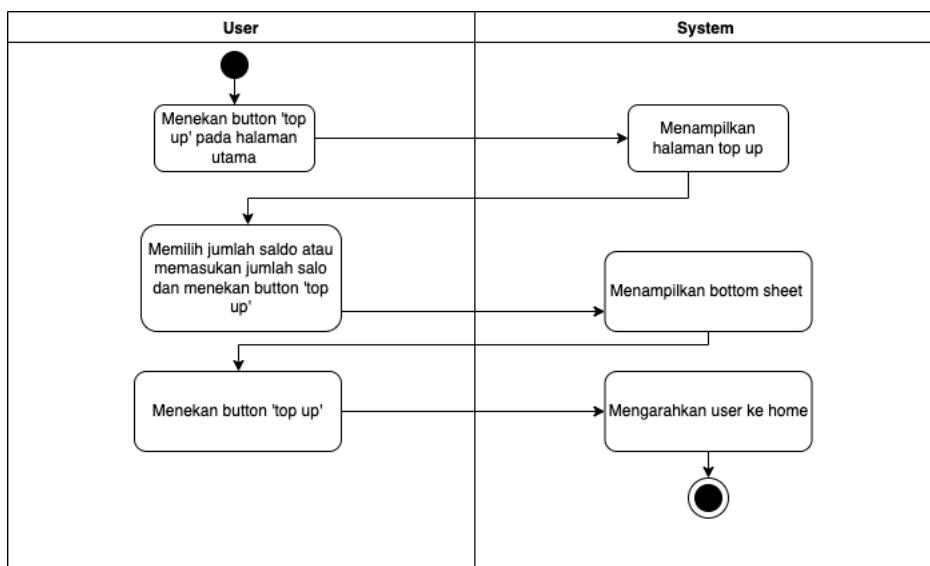
Gambar 3.18 Activity Diagram Melakukan Login

### 3.3.2.3.3 Activity Diagram Melihat Status Pesanan



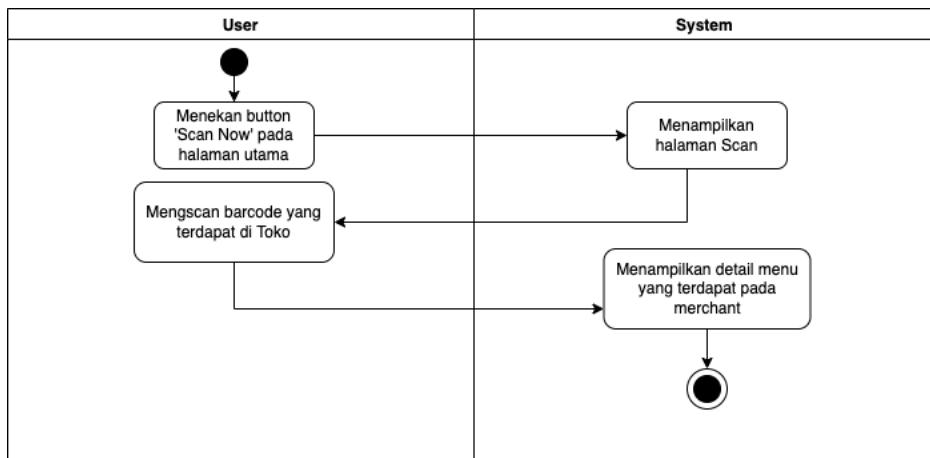
Gambar 3.19 Activity Diagram Melihat Status Pesanan

### 3.3.2.3.4 Activity Diagram Melakukan Isi Ulang Saldo



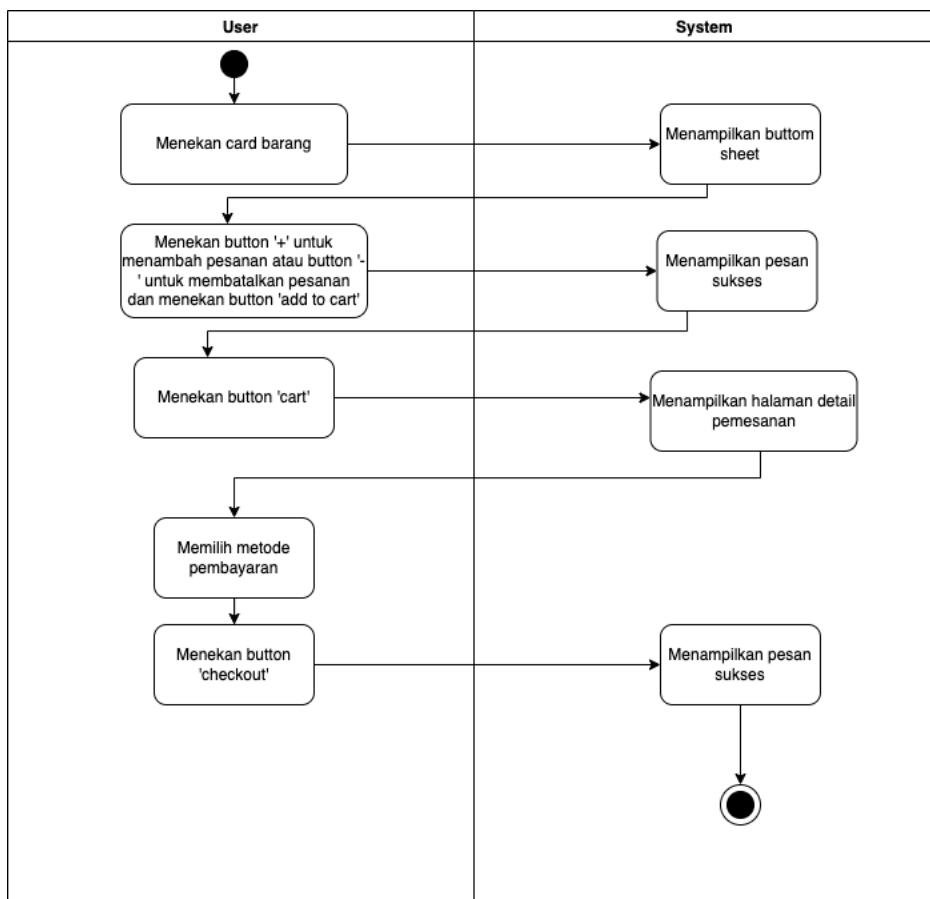
Gambar 3.20 Activity Diagram Melakukan Isi Ulang Saldo

### 3.3.2.3.5 Activity Diagram Memilih Merchant



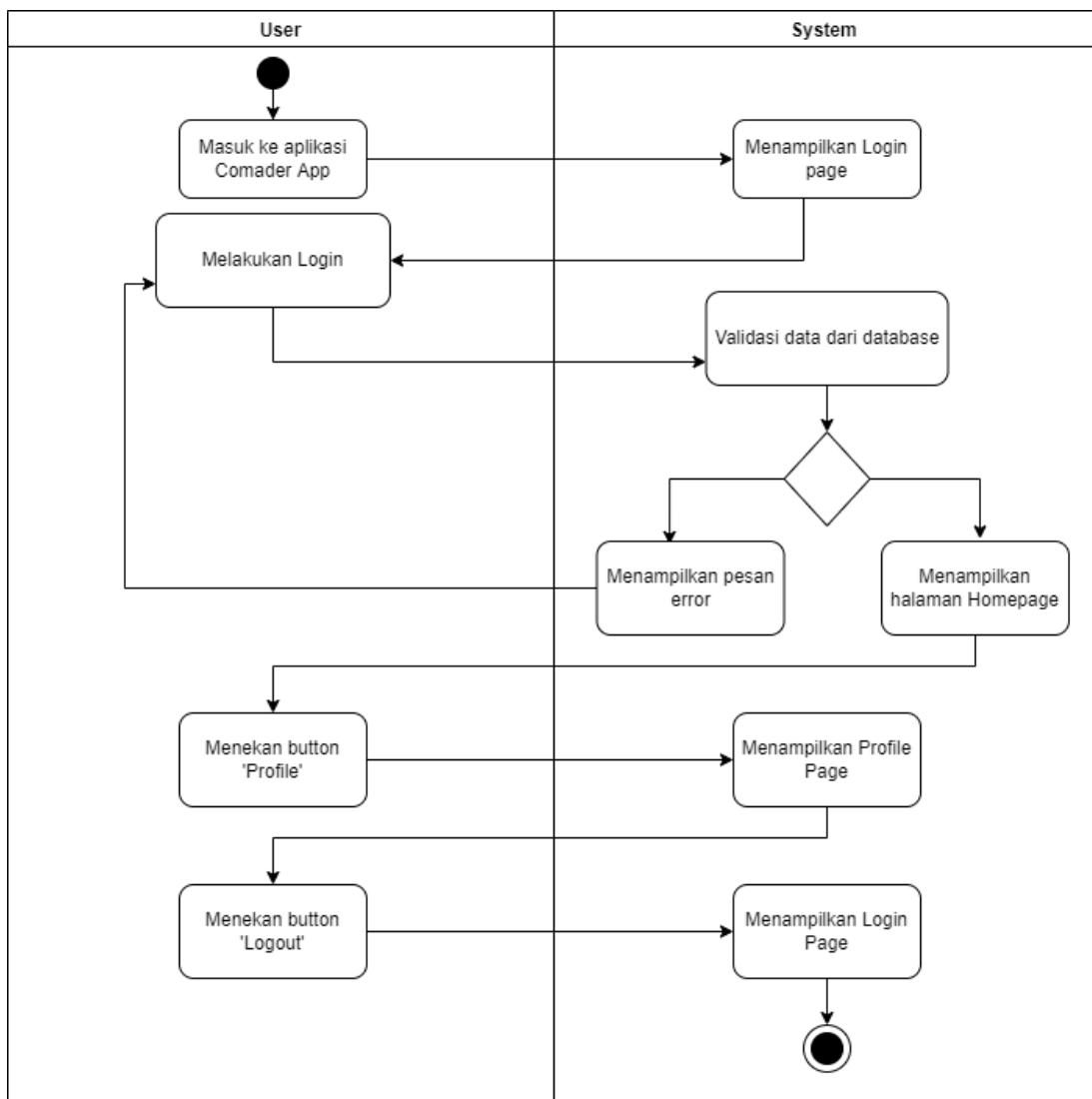
Gambar 3.21 Activity Diagram Memilih Merchant

### 3.3.2.3.6 Activity Diagram Memesan Barang



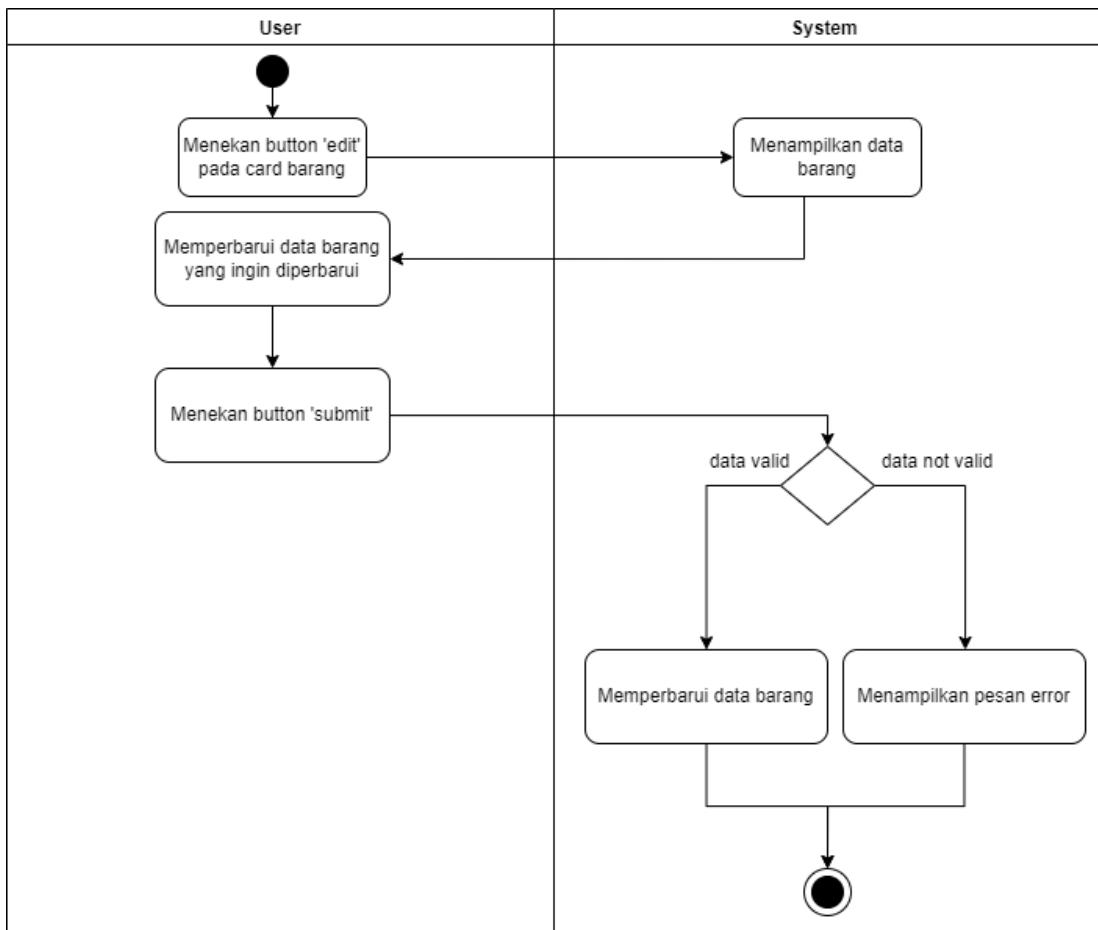
Gambar 3.22 Activity Diagram Memesan Barang

### 3.3.2.3.7 Activity Diagram Melakukan Logout



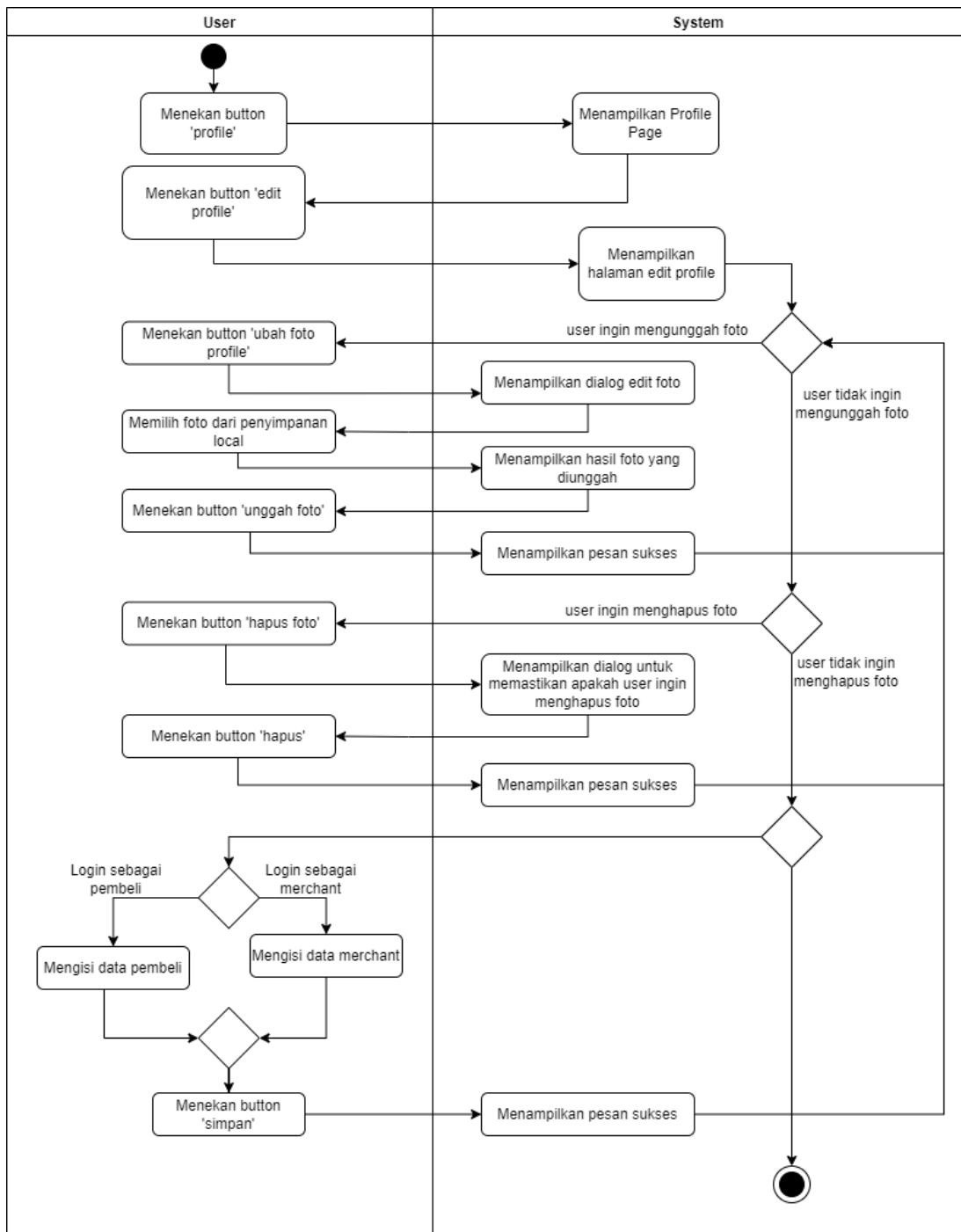
Gambar 3.23 Activity Diagram Melakukan Logout

### 3.3.2.3.8 Activity Diagram Memperbarui Data Barang



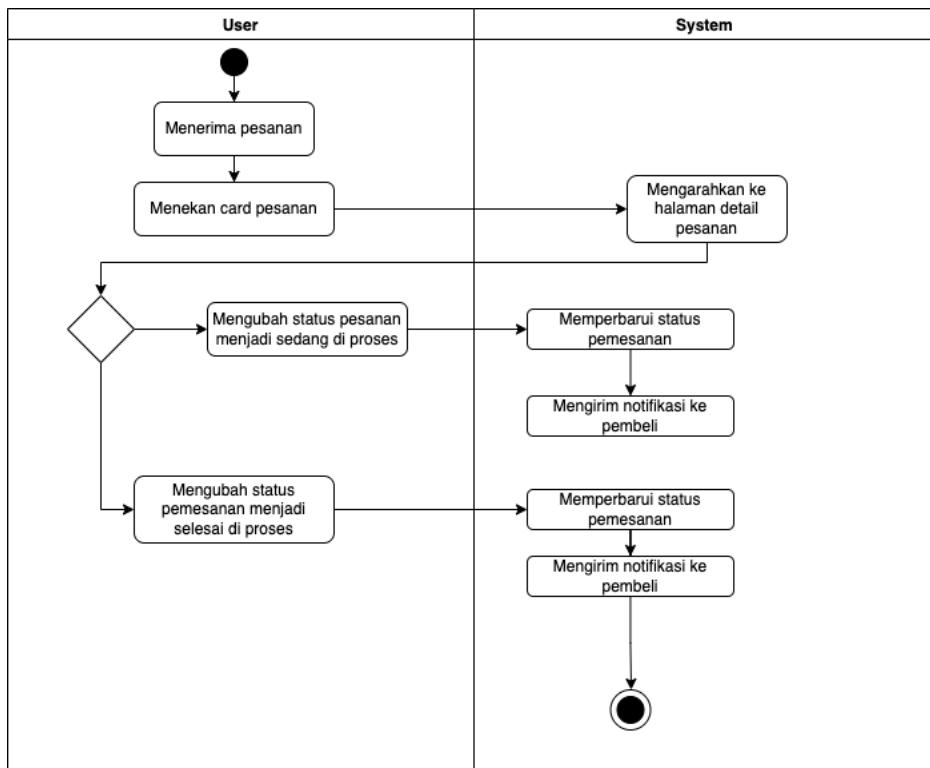
Gambar 3.24 Memperbaharui Data Barang

### 3.3.2.3.9 Activity Diagram Melakukan Update Profil



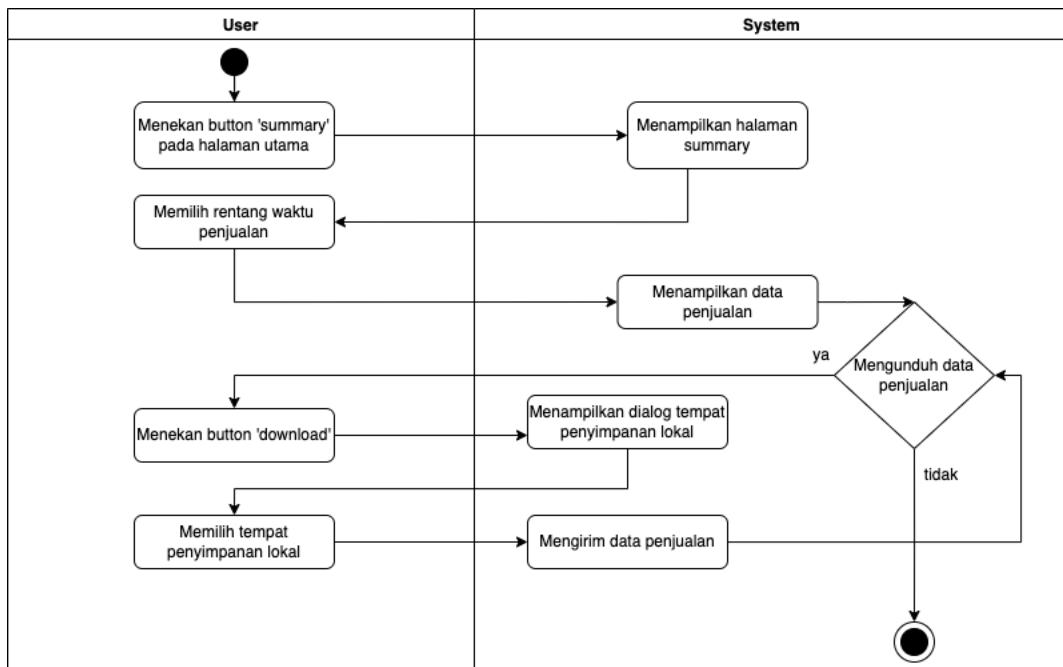
Gambar 3.25 Activity Diagram Melakukan Update Profil

### 3.3.2.3.10 Activity Diagram Memproses Pesanan



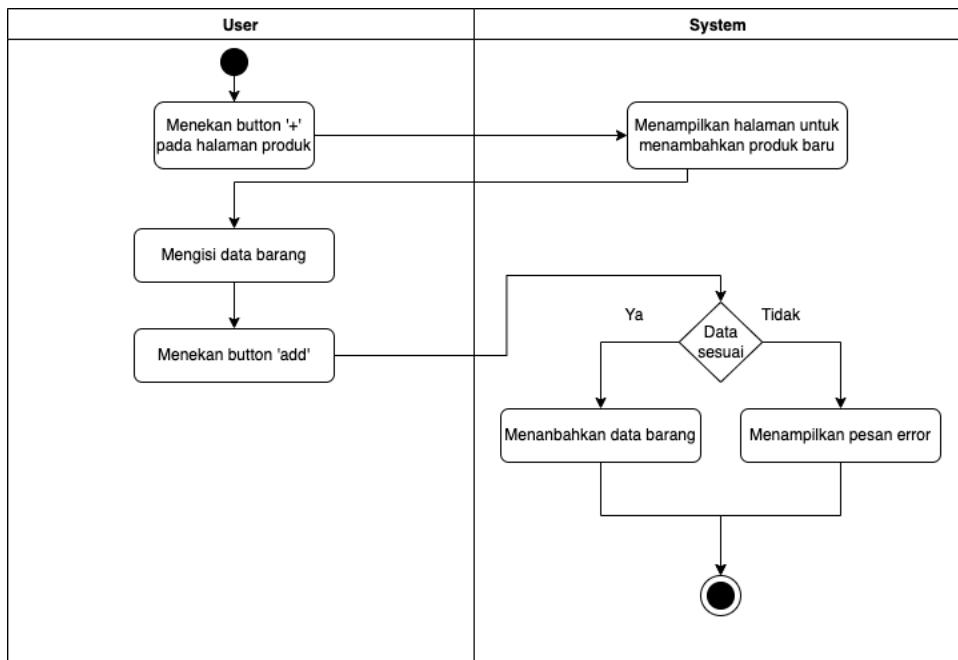
Gambar 3.26 Activity Diagram Memproses Pesanan

### 3.3.2.3.11 Activity Diagram Melihat Data Penjualan



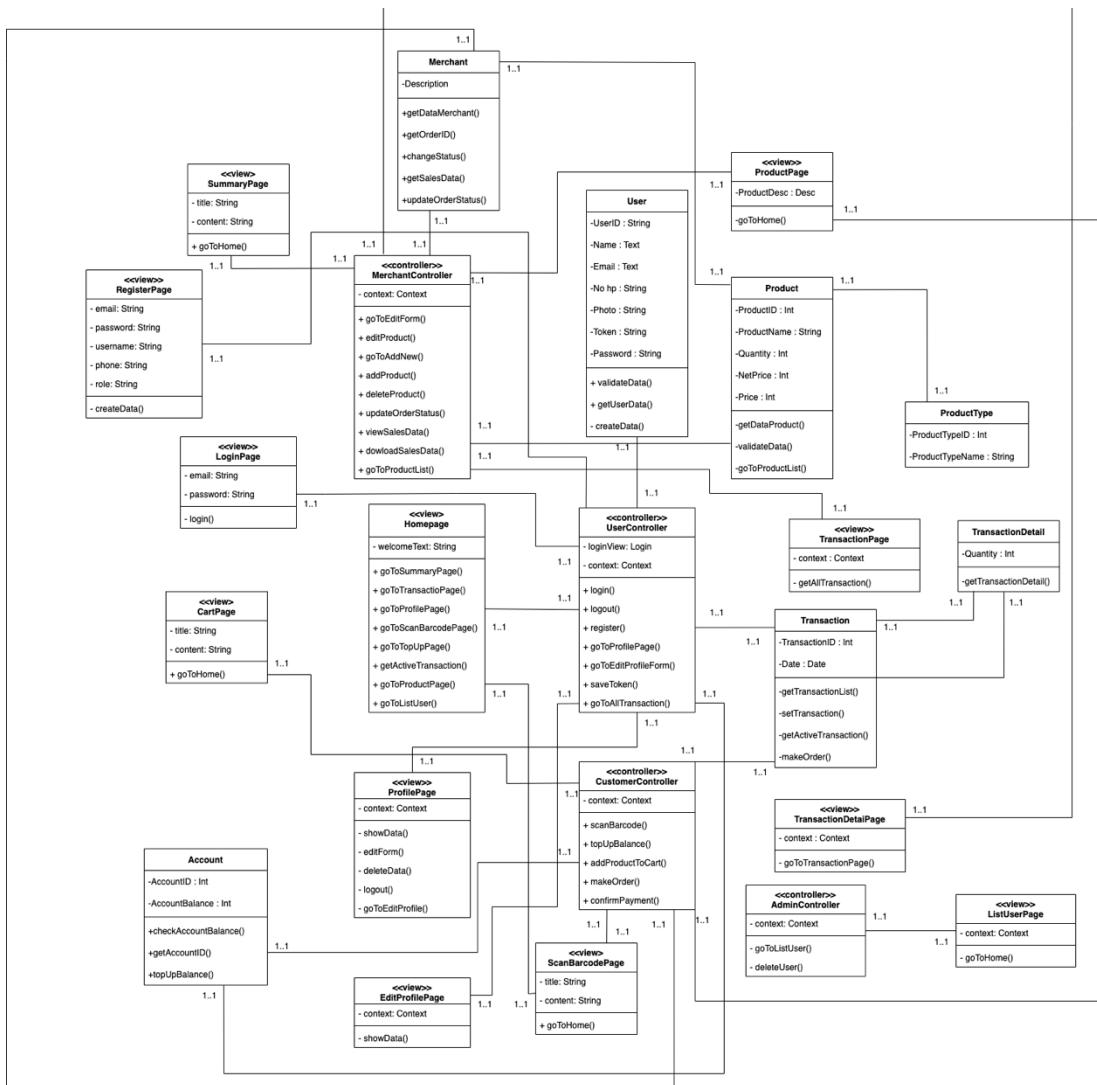
Gambar 3.27 Activity Diagram Melihat Data Penjualan

### 3.3.2.3.12 Activity Diagram Input Barang Penjualan



Gambar 3.28 Activity Diagram Input Barang Penjualan

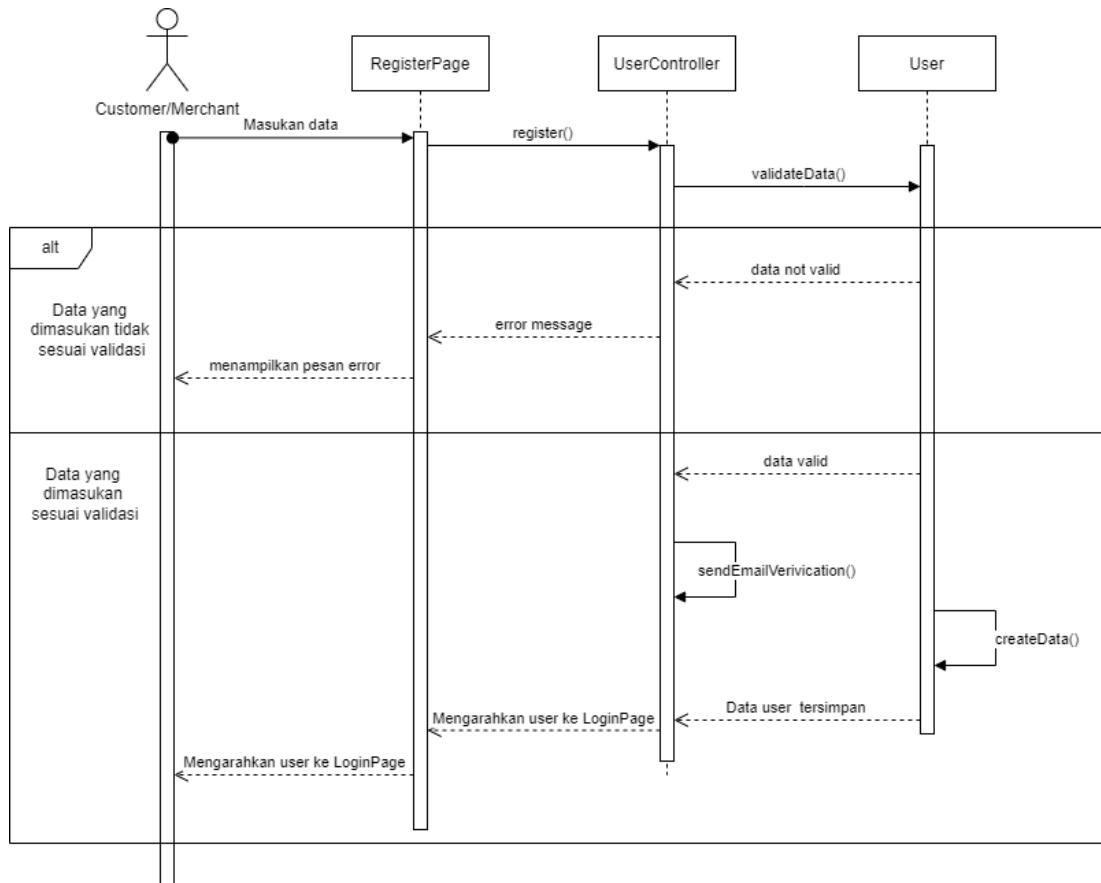
### 3.3.2.5 Class Diagram



Gambar 3.29 Class Diagram Aplikasi ComanderApp

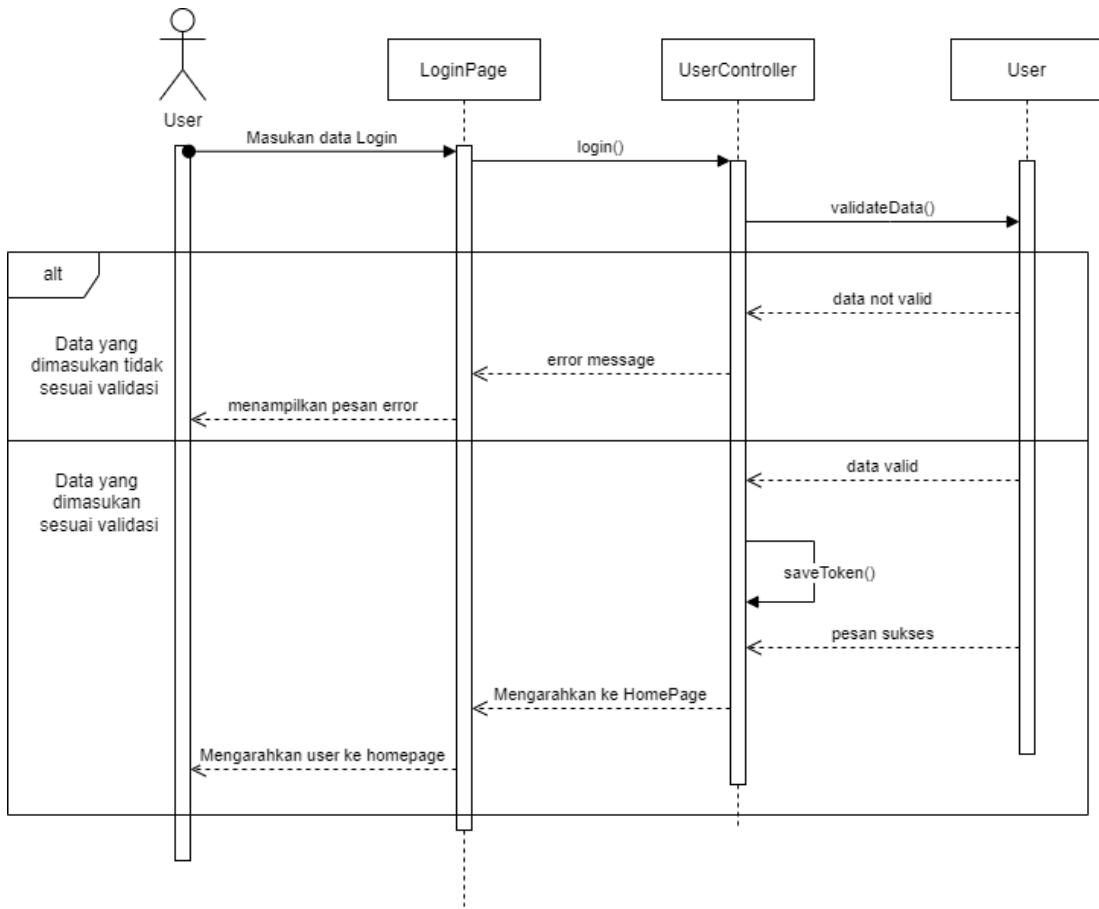
### 3.3.2.5 Sequence Diagram

#### 3.3.2.5.1 Sequence Diagram Sign Up



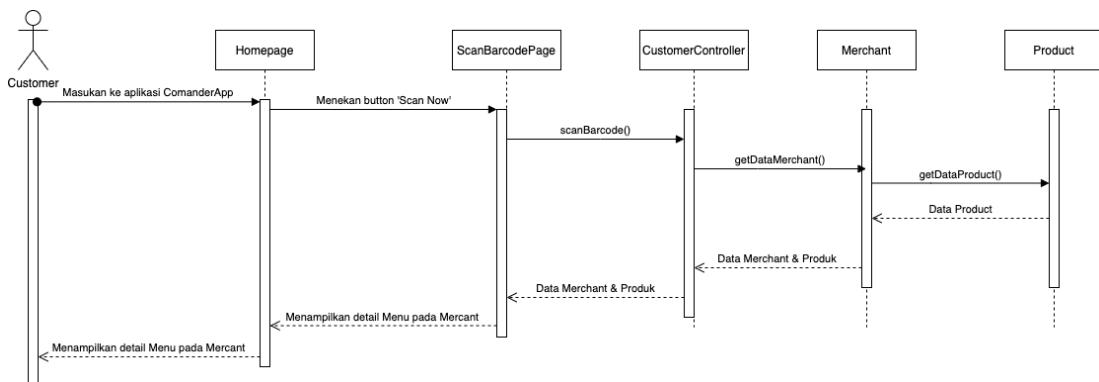
Gambar 3.30 Sequence Diagram Sign Up

### 3.3.2.5.2 Sequence Diagram Login



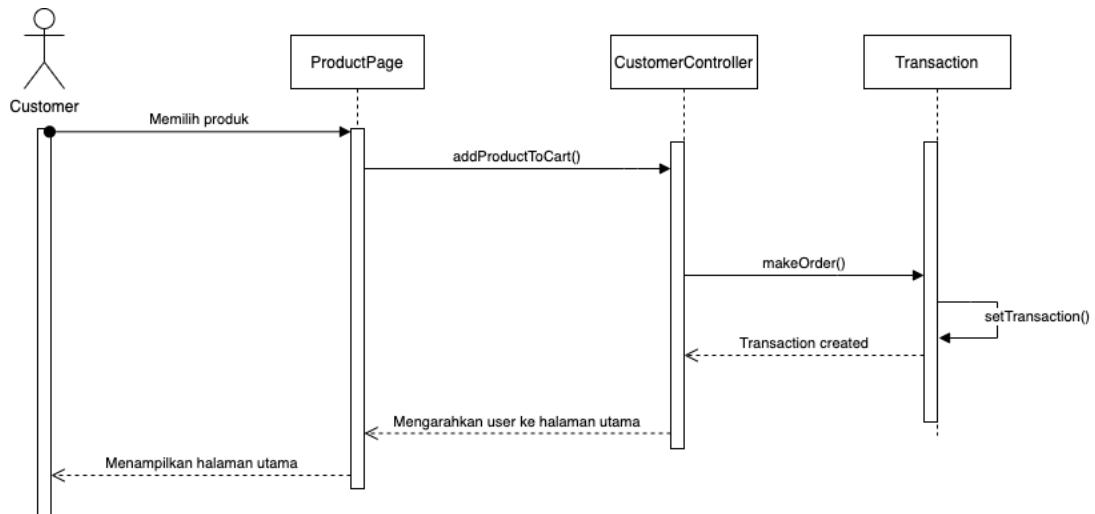
Gambar 3.31 Sequence Diagram Login

### 3.3.2.5.3 Sequence Diagram Memilih Merchant



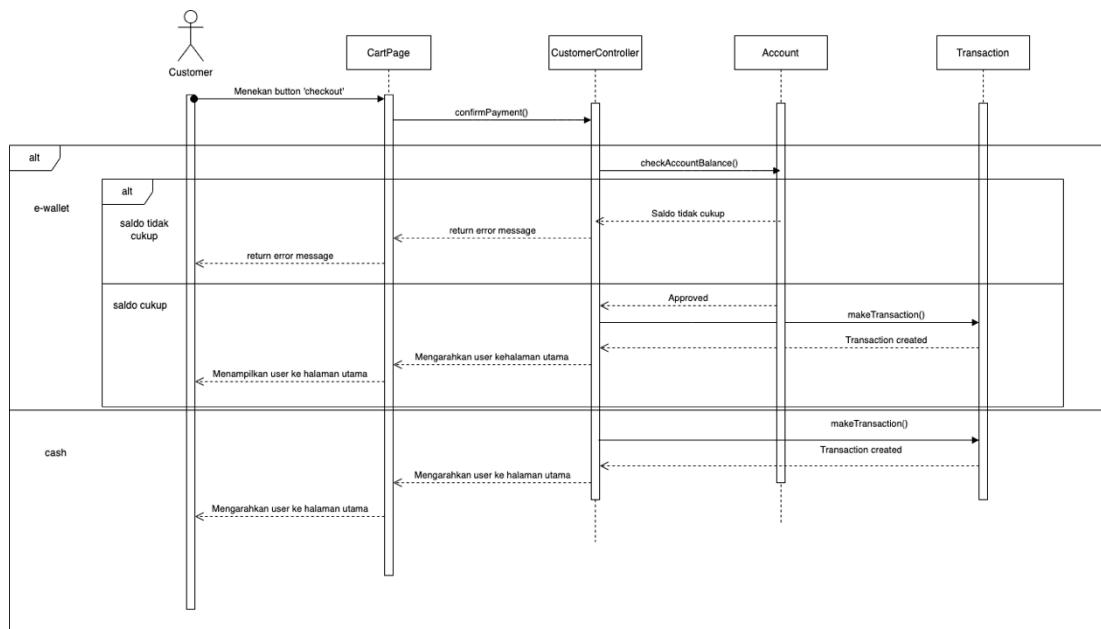
Gambar 3.32 Sequence Diagram Memilih Merchant

### 3.3.2.5.4 Sequence Diagram Memesan Barang



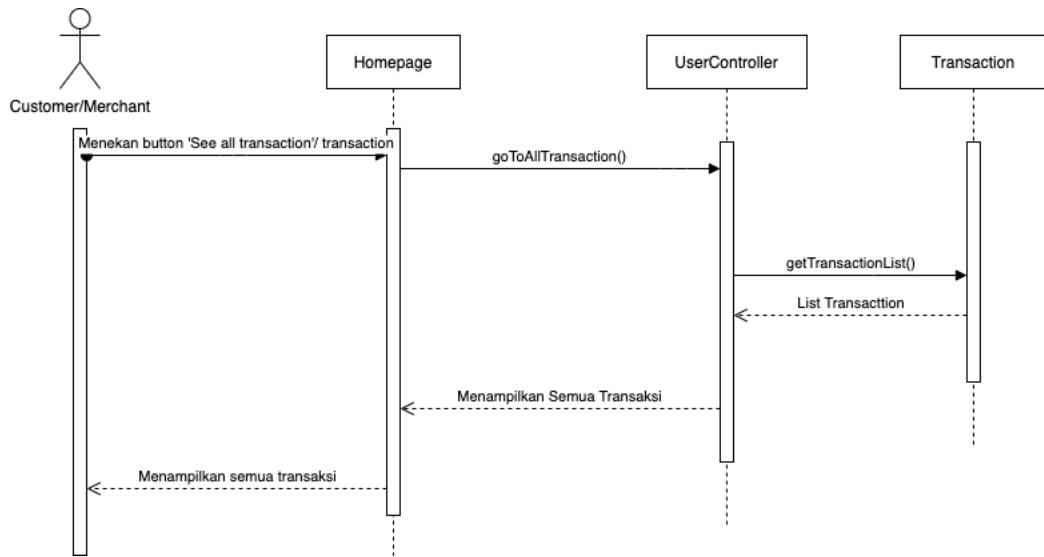
Gambar 3.33 Sequence Diagram Memesan Barang

### 3.3.2.5.5 Sequence Diagram Melakukan Pembayaran



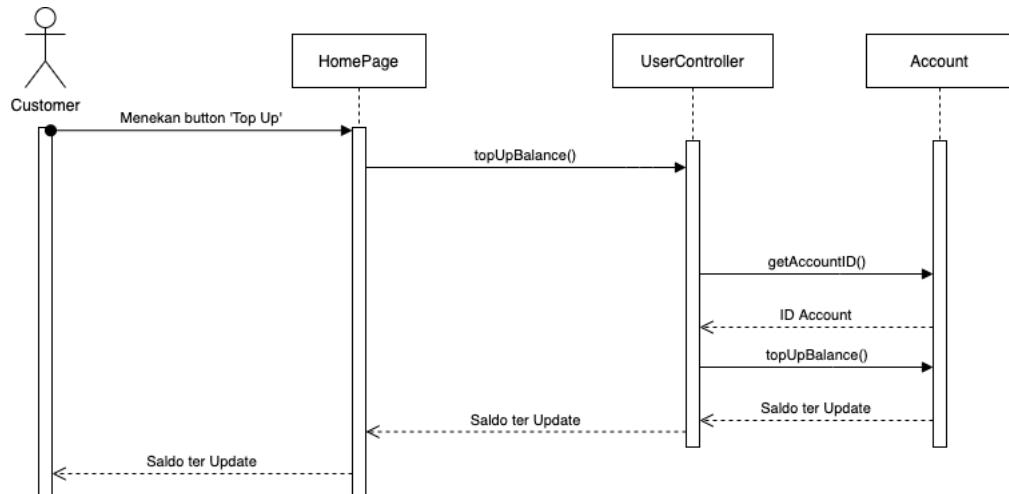
Gambar 3.34 Sequence Diagram Melakukan Pembayaran

### 3.3.2.5.6 Sequence Diagram Melihat Status Pemesanan



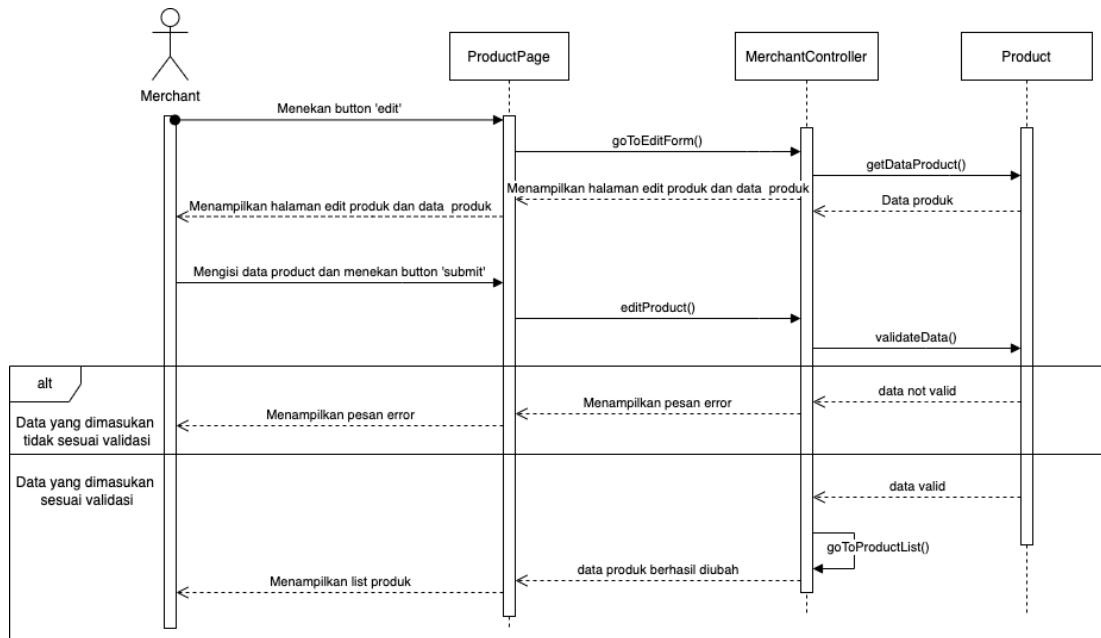
Gambar 3.35 Sequence Diagram Melihat Status Pemesanan

### 3.3.2.5.7 Sequence Diagram Melakukan Isi Ulang Saldo



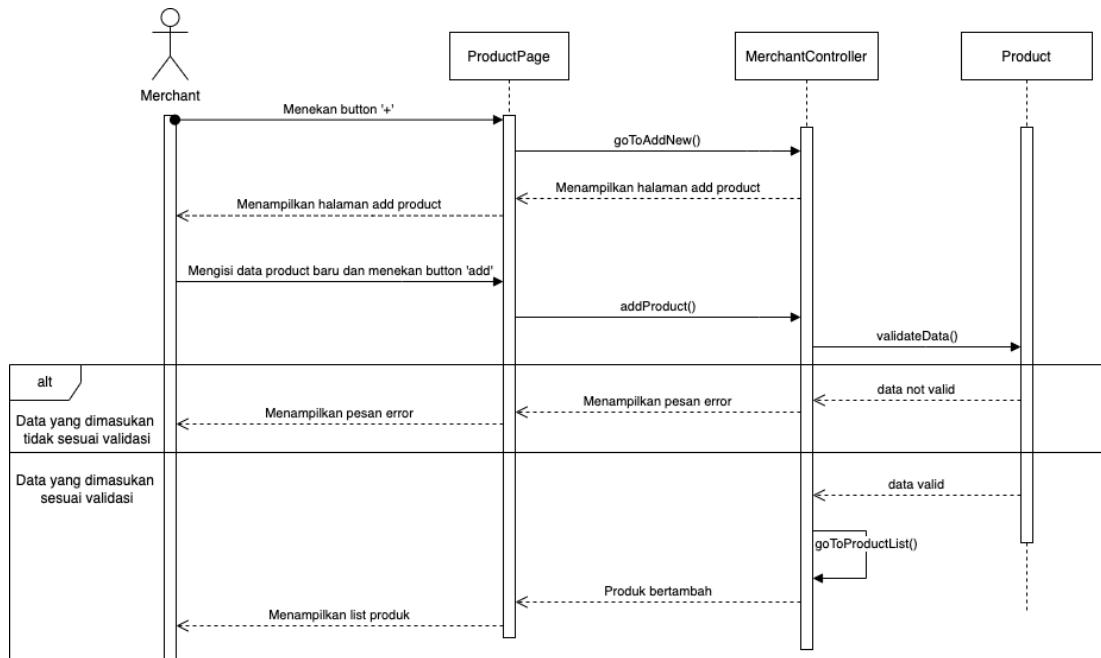
Gambar 3.36 Sequence Diagram Melakukan Isi Ulang Saldo

### 3.3.2.5.8 Sequence Diagram Memperbarui Data Barang



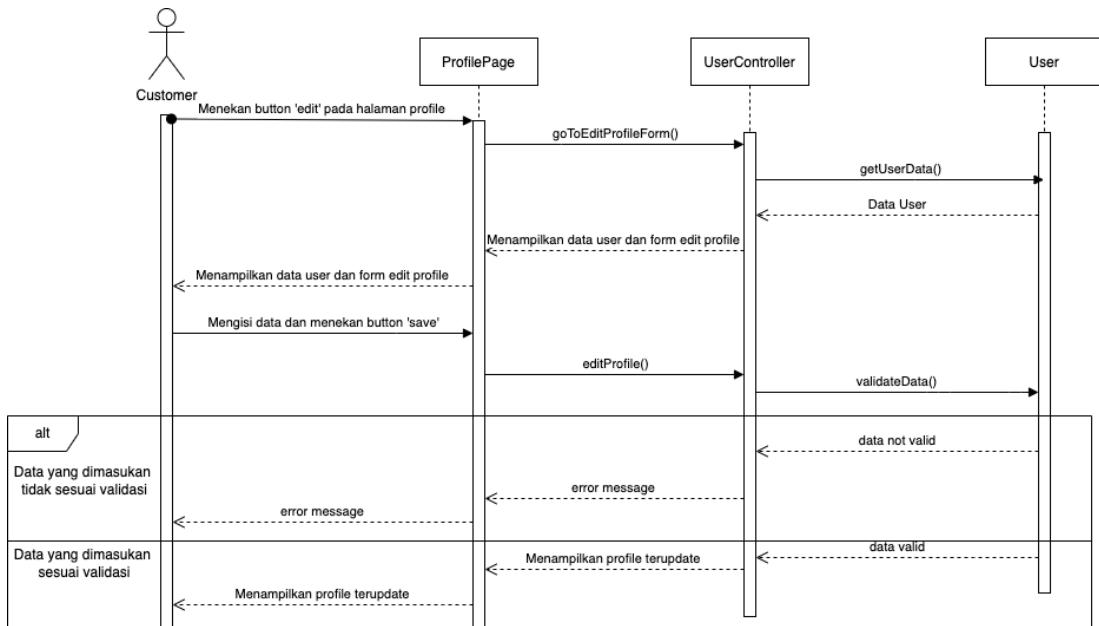
Gambar 3.37 Sequence Diagram Memperbarui Data Barang

### 3.3.2.5.9 Sequence Diagram Input data barang penjualan



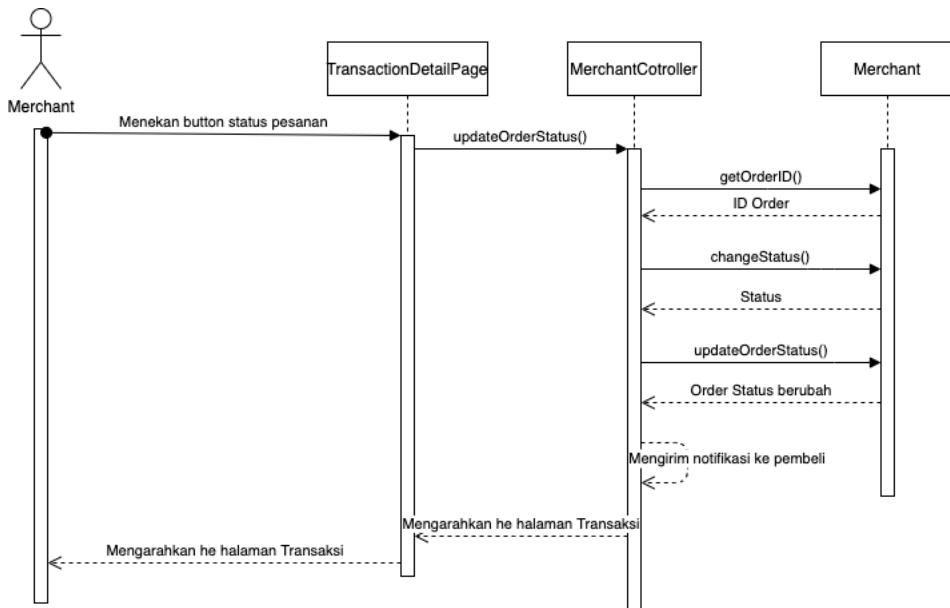
Gambar 3.38 Sequence Diagram Input Data Barang Penjualan

### 3.3.2.5.10 Sequence Diagram Mengubah Profil



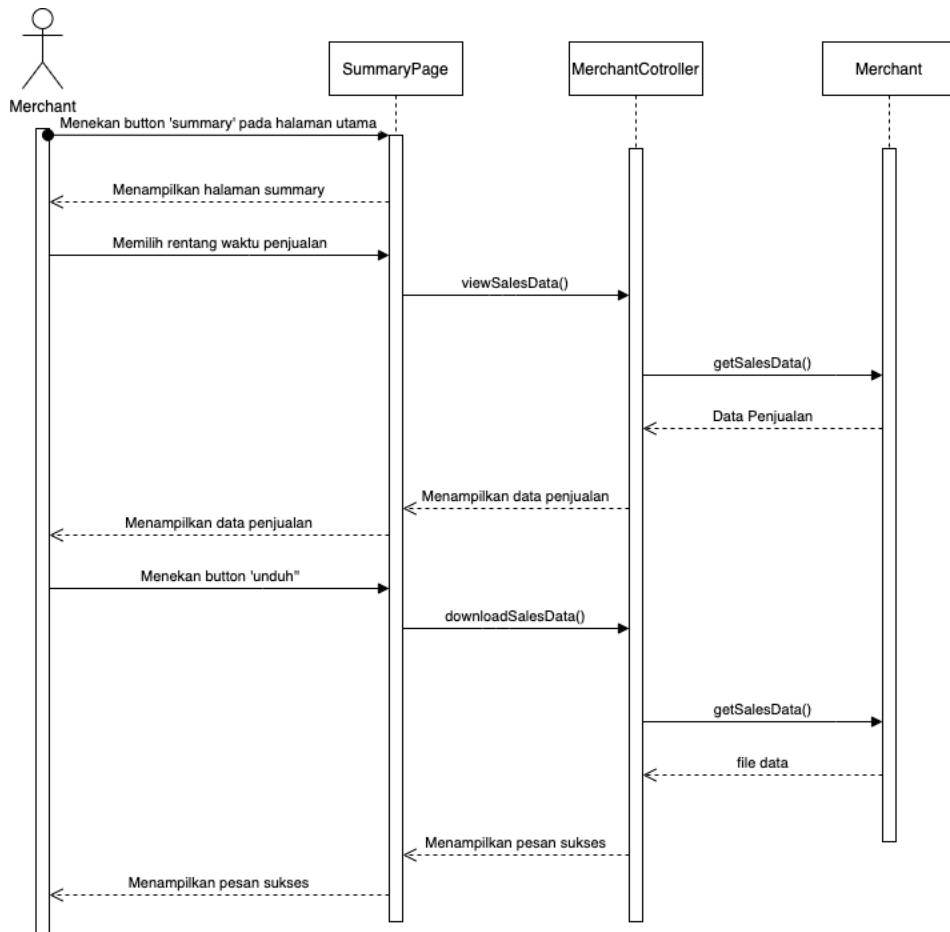
Gambar 3.39 Sequence Diagram Mengubah Profil

### 3.3.2.5.11 Sequence Diagram Memproses Pesanan



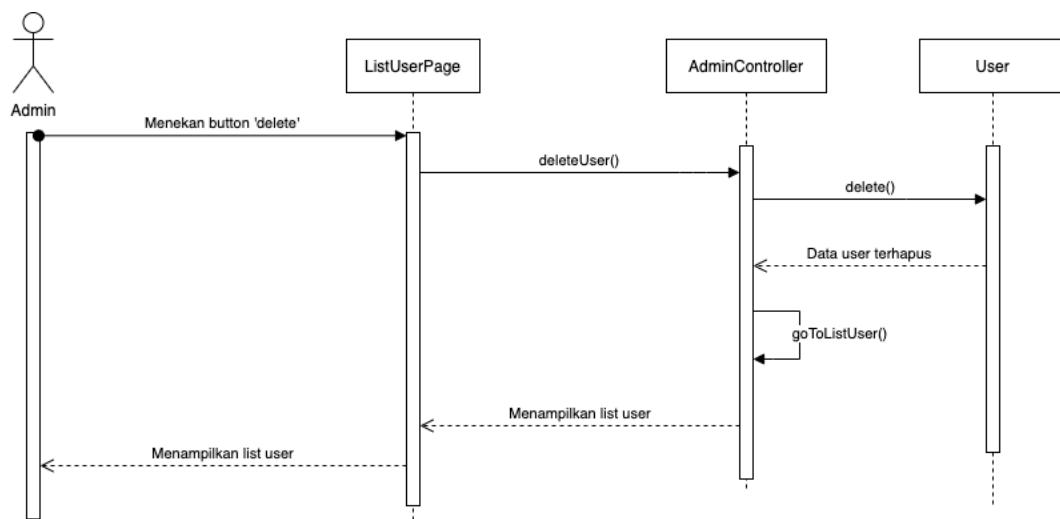
Gambar 3.40 Sequence Diagram Memproses Pesanan

### 3.3.2.5.12 Sequence Diagram Melihat Data Penjualan



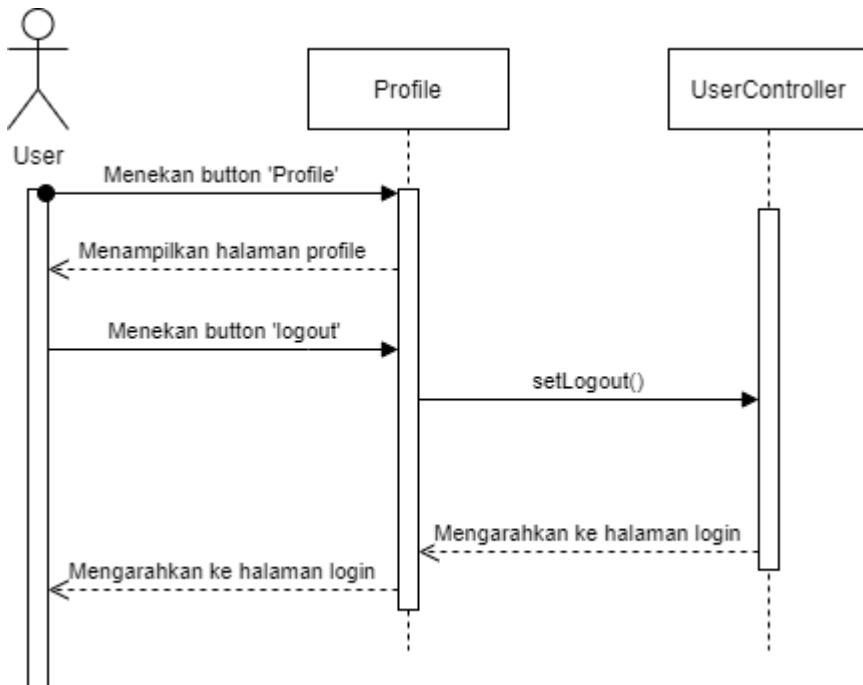
Gambar 3.41 Sequence Diagram Melihat Data Penjualan

### 3.3.2.5.13 Sequence Diagram Hapus Pembeli atau Merchant



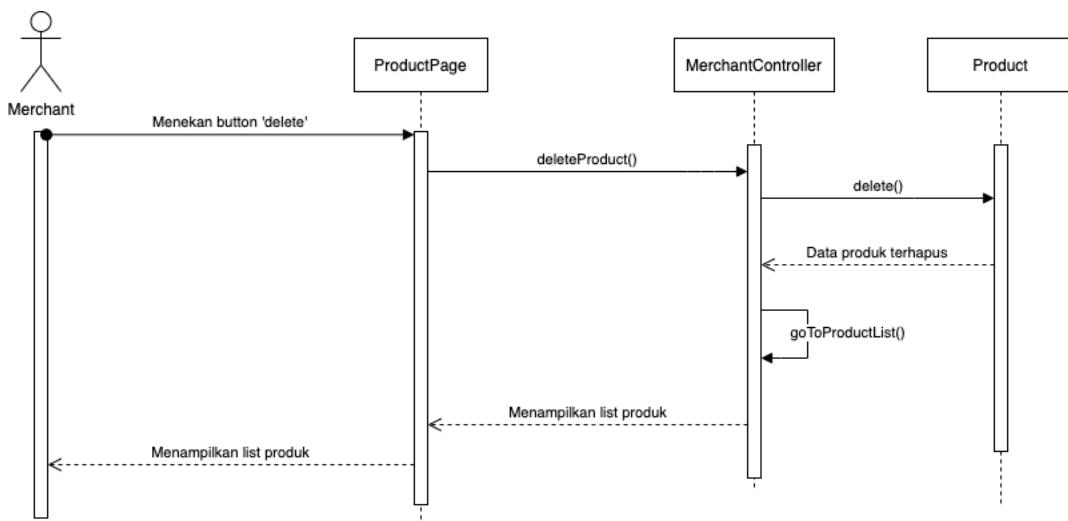
Gambar 3.42 Sequence Diagram Hapus Pembeli atau Merchant

### 3.3.2.5.14 Sequence Diagram Logout



Gambar 3.43 Sequence Diagram Logout

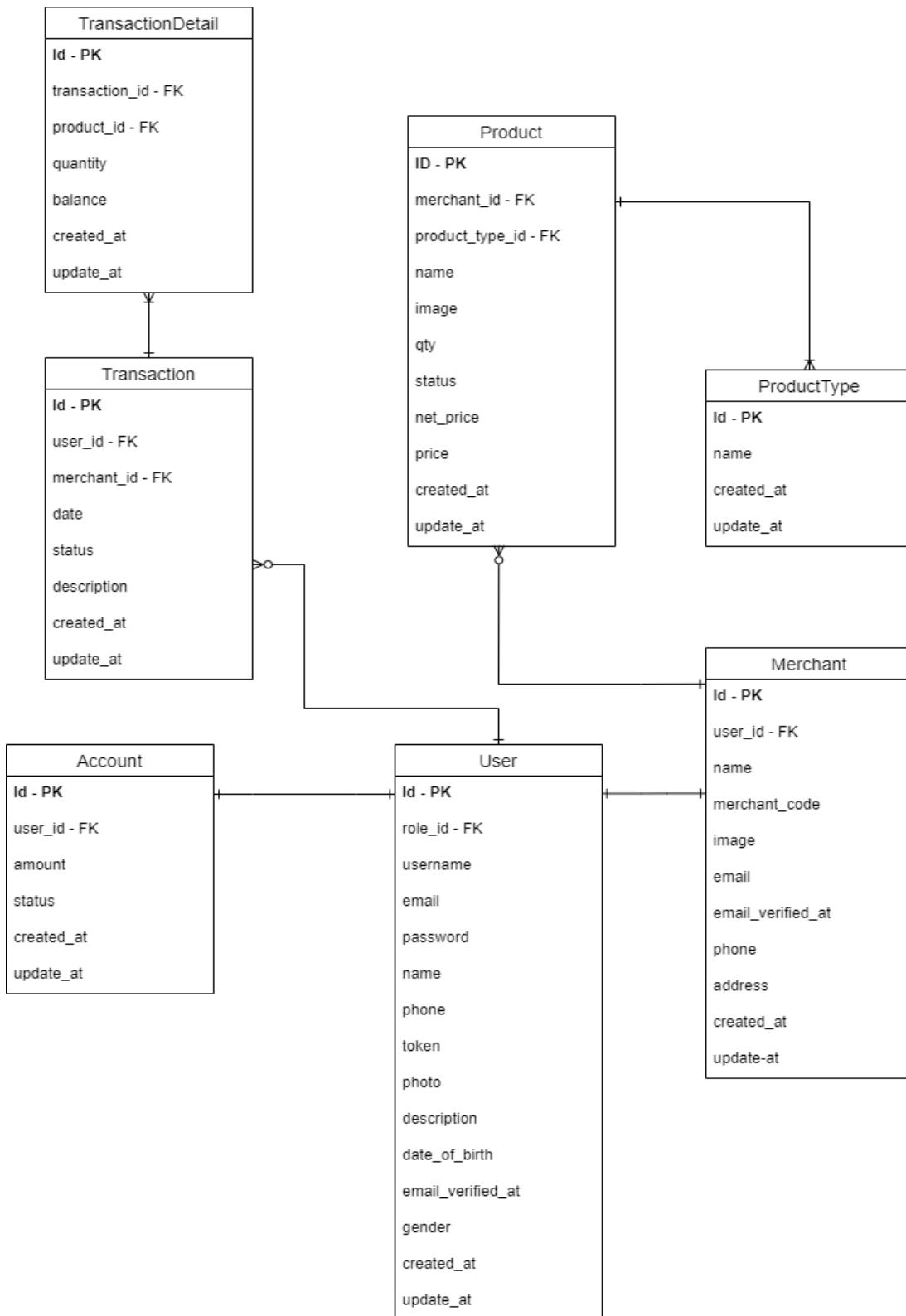
### 3.3.2.5.14 Sequence Diagram Logout



Gambar 3.44 Sequence Diagram Hapus Produk

### 3.3.3 Perancangan Database

#### 3.3.2.1 Entity Relationship Diagram



Gambar 3.45 ERD Aplikasi Comander App

### 3.3.2.1 Data Dictionary

#### 3.3.2.1.1 Data User

User
<b>Id - PK</b>
role_id - FK
username
email
password
name
phone
token
photo
description
date_of_birth
email_verified_at
gender
created_at
update_at

Gambar 3.46 Tabel User

Tabel 3.21 Data Dictionary User

<b>Entity</b>	<b>Attribute</b>	<b>Data Type</b>	<b>NULL/NOT NULL</b>	<b>Description</b>
User	Id	int (11)	NOT NULL	<i>Primary Key</i>
	role_id	varchar (225)	NOT NULL	<i>Foreign Key</i>
	username	varchar (225)	NOT NULL	Username yang didaftarkan user
	token	int (11)	NOT NULL	Token yang diterima user
	name	varchar (225)	NOT NULL	Nama yang didaftarkan user

	gender	varchar (225)	NOT NULL	Gender yang didaftarkan user
	date_of_birth	varchar (225)	NOT NULL	Tanggal lahir yang didaftarkan user
	email	varchar (225)	NOT NULL	Email yang didaftarkan suer
	email_verified_at	timestamp	NOT NULL	Waktu saat email terverifikasi
	phone	varchar (225)	NOT NULL	Nomor handphone yang didaftarkan user
	password	varchar (225)	NOT NULL	Password yang didaftarkan user
	created_at	timestamp	NOT NULL	Waktu yang menunjukan User dibuat
	update_at	timestamp	NOT NULL	Waktu yang menunjukan

### 3.3.2.1.2 Data Account

Account
<b>Id - PK</b>
user_id - FK
amount
status
created_at
update_at

Gambar 3.47 Tabel Account

Tabel 3.22 Data Dictionary Account

<b>Entity</b>	<b>Attribute</b>	<b>Data Type</b>	<b>NULL/NOT NULL</b>	<b>Description</b>
Account	Id	int (11)	NOT NULL	<i>Primary Key</i>
	user_id	varchar (225)	NOT NULL	<i>Foreign Key</i>
	amount	varchar (225)	NOT NULL	Amount yang dimiliki account
	status	int (11)	NOT NULL	Status account
	created_at	timestamp	NOT NULL	Waktu yang menunjukkan Account dibuat
	update_at	timestamp	NOT NULL	Waktu yang menunjukkan Account terupdate

### 3.3.2.1.3 Data Product

Product
<b>ID - PK</b>
merchant_id - FK
product_type_id - FK
name
image
qty
status
net_price
price
created_at
update_at

Gambar 3.48 Tabel Product

Tabel 3.23 Data Dictionary Product

<b>Entity</b>	<b>Attribute</b>	<b>Data Type</b>	<b>NULL/NOT NULL</b>	<b>Description</b>
Product	Id	int (11)	NOT NULL	<i>Primary Key</i>
	merchant_id	varchar (225)	NOT NULL	<i>Foreign Key</i>
	product_type_id	varchar (225)	NOT NULL	<i>Foreign Key</i>
	name	int (11)	NOT NULL	Nama produk
	image	varchar (225)	NOT NULL	Gambar produk
	qty	varchar (225)	NOT NULL	Kuantitas produk
	status	boolean	NOT NULL	Status produk
	net_price	varchar (225)	NOT NULL	Harga bersih produk
	price	varchar (225)	NOT NULL	Harga produk
	created_at	timestamp	NOT NULL	Waktu yang menunjukan Product dibuat
	update_at	timestamp	NOT NULL	Waktu yang menunjukan Product terupdate

#### 3.3.2.1.4 Data Product Type

ProductType
<b>Id - PK</b>
name
created_at
update_at

Gambar 3.49 Tabel ProductType

Tabel 3.24 Data Dictionary ProductType

<b>Entity</b>	<b>Attribute</b>	<b>Data Type</b>	<b>NULL/NOT NULL</b>	<b>Description</b>
ProductType	Id	int (11)	NOT NULL	<i>Primary Key</i>
	name	varchar (225)	NOT NULL	Nama tipe produk
	created_at	timestamp	NOT NULL	Waktu yang menunjukkan kapan produk dibuat
	update_at	timestamp	NOT NULL	Waktu yang menunjukkan kapan produk terupdate

### 3.3.2.1.5 Data Transaction

Transaction
<b>Id - PK</b>
user_id - FK
merchant_id - FK
date
status
description
created_at
update_at

Gambar 3.50 Tabel Transaction

Tabel 3.25 Data Dictionary Transaction

<b>Entity</b>	<b>Attribute</b>	<b>Data Type</b>	<b>NULL/NOT NULL</b>	<b>Description</b>
Transaction	Id	int (11)	NOT NULL	<i>Primary Key</i>
	user_id	varchar (225)	NOT NULL	<i>Foreign Key</i>
	merchant_id	varchar (225)	NOT NULL	<i>Foreign Key</i>
	date	date	NOT NULL	Tanggal transaksi

	status	int (11)	NOT NULL	Status transaksi
	description	varchar (225)	NOT NULL	Deskripsi transaksi
	created_at	timestamp	NOT NULL	Waktu yang menunjukkan kapan transaksi dibuat
	update_at	timestamp	NOT NULL	Waktu yang menunjukkan kapan transaksi terupdate

### 3.3.2.1.6 Data TransactionDetail

TransactionDetail
<b>Id - PK</b>
transaction_id - FK
product_id - FK
quantity
balance
created_at
update_at

Gambar 3.51 Tabel TransactionDetail

Tabel 3.26 Data Dictionary TransactionDetail

<b>Entity</b>	<b>Attribute</b>	<b>Data Type</b>	<b>NULL/NOT NULL</b>	<b>Description</b>
Transaction Detail	Id	int (11)	NOT NULL	<i>Primary Key</i>
	transaction_id	varchar (225)	NOT NULL	<i>Foreign Key</i>
	product_id	varchar (225)	NOT NULL	Foreign Key
	quantity	varchar (225)	NOT NULL	Kuantitas tiap produk

	balance	varchar (225)	NOT NULL	Biaya yang digunakan dalam transaksi
	created_at	timestamp	NOT NULL	Waktu yang menunjukan transaksi dibuat
	update_at	timestamp	NOT NULL	Waktu yang menunjukan transaksi terupdate

### 3.3.2.1.7 Data Merchant

Merchant	
<b>Id - PK</b>	
user_id - FK	
name	
merchant_code	
image	
email	
email_verified_at	
phone	
address	
created_at	
update-at	

Gambar 3.52 Tabel Merchant

Tabel 3.27 Data Dictionary Merchant

<b><i>Entity</i></b>	<b><i>Attribute</i></b>	<b><i>Data Type</i></b>	<b><i>NULL/NOT NULL</i></b>	<b><i>Description</i></b>
Merchant	Id	int (11)	NOT NULL	<i>Primary Key</i>
	user_id	varchar (225)	NOT NULL	<i>Foreign Key</i>
	name	varchar (225)	NOT NULL	Nama Merchant
	merchant_code	varchar (225)	NOT NULL	Kode Merchant
	image	varchar (225)	NOT NULL	Foto Merchant
	email	varchar (225)	NOT NULL	Email Merchant
	email_verified_at	varchar (225)	NOT NULL	Waktu saat email terverifikasi
	phone	varchar (225)	NOT NULL	Nomor handphone yang didaftarkan Merchant
	address	varchar (225)	NOT NULL	Alamat yang didaftarkan Merchant
	created_at	timestamp	NOT NULL	Waktu saat Merchant dibuat
	update_at	timestamp	NOT NULL	Waktu saat data Merchant terupdate

### 3.3.4 Perancangan *User Interface* (UI)

#### 3.3.3.1 User Customer

##### 3.3.3.1.1 Halaman Register

The image shows a user interface for a customer registration form. At the top, the word "Comander" is written in a purple font. Below it is a series of input fields for personal information:

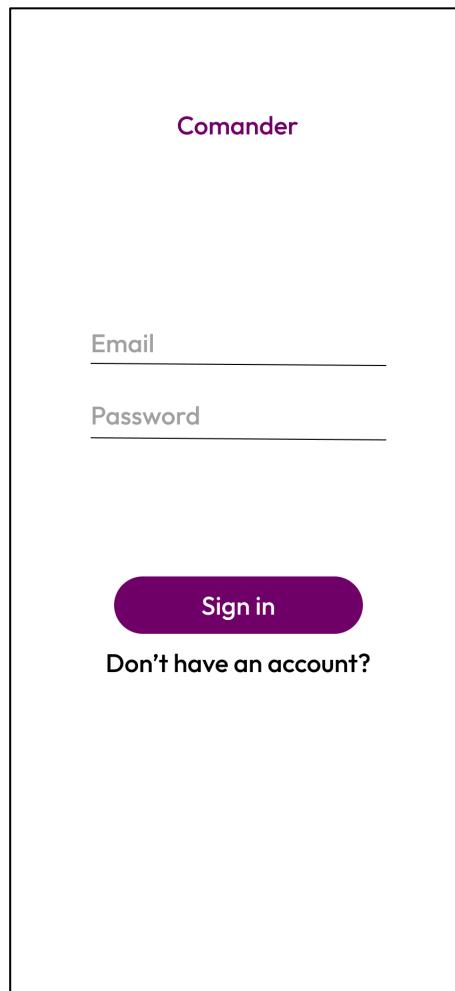
- Name: An input field with the placeholder "Name".
- Gender: An input field with the placeholder "Gender".
- Date of Birth: An input field with the placeholder "Date of Birth".
- Phone: An input field with the placeholder "Phone".
- Username: An input field with the placeholder "Username".
- Email: An input field with the placeholder "Email".
- Password: An input field with the placeholder "Password".
- Confirm Password: An input field with the placeholder "Confirm Password".

At the bottom center of the form is a purple button with the text "Sign up" in white.

Gambar 3.53 User Interface Register Customer

User yang belum memiliki akun harus mendaftarkan diri melalui halaman register agar data user dapat tersimpan dalam database.

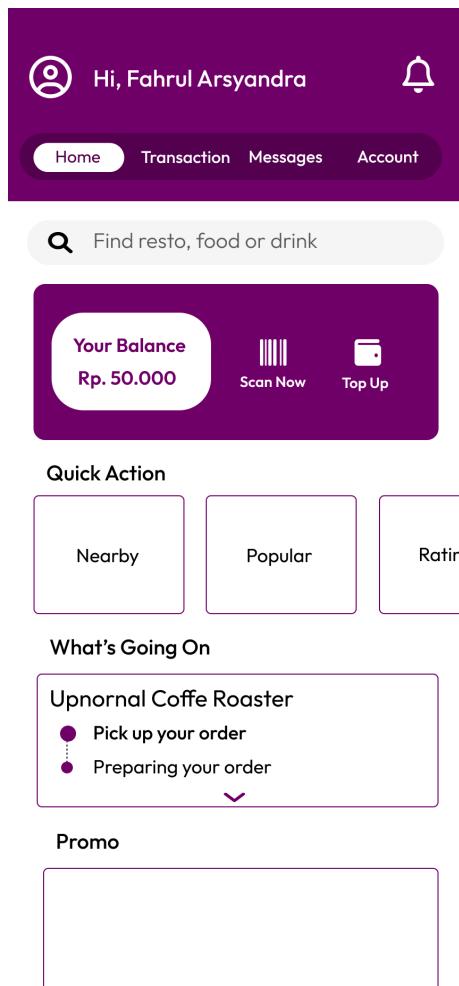
### 3.3.3.1.2 Halaman Login



Gambar 3.54 User Interface Login

Sebelum customer dapat mengakses fitur-fitur yang terdapat dalam aplikasi, customer harus melakukan login melalui page Login yang terdapat dalam aplikasi ComanderApp.

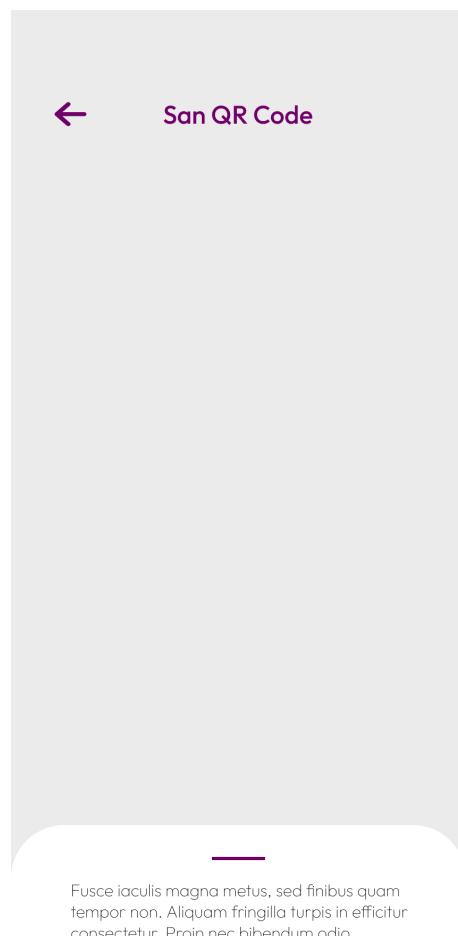
### 3.3.3.1.3 Halaman Home



Gambar 3.55 User Interface Homepage

Setelah melakukan login, customer akan dapat melihat halaman Home ComanderApp. Pada halaman ini customer dapat mengakses fitur Scan Barcode dan fitur Topup.

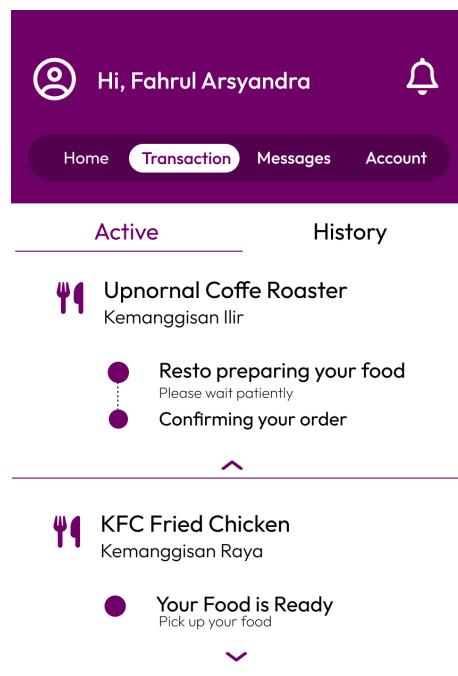
### 3.3.3.1.4 Halaman Scan Barcode



Gambar 3.56 User Interface halaman Scan Barcode

Pada halaman ini, customer dapat mengscan barcode yang terdapat pada tiap toko/merchant untuk melihat setiap menu atau produk yang dimiliki merchant dan untuk melakukan transaksi.

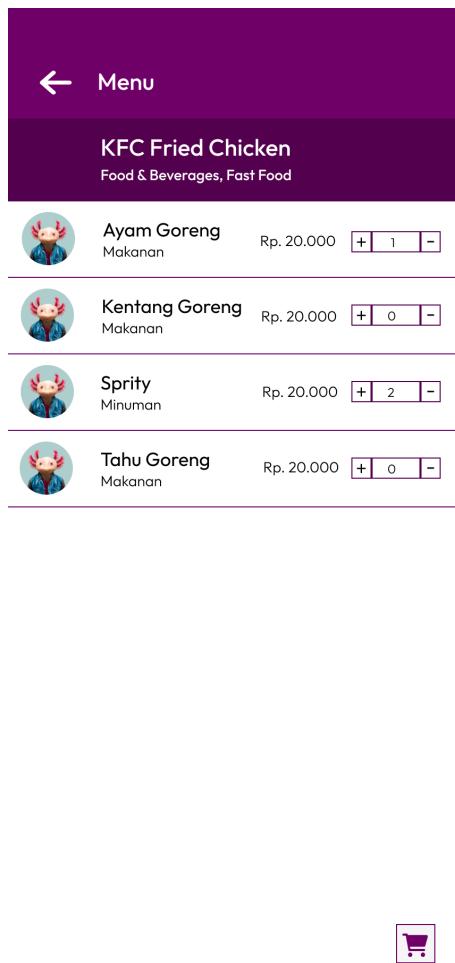
### 3.3.3.1.5 Halaman Transaksi



Gambar 3.57 User Interface halaman Transaksi

Pada halaman transaksi, customer dapat melihat transaksi yang sedang berlangsung beserta statusnya dan Riwayat transaksi yang pernah dilakukan.

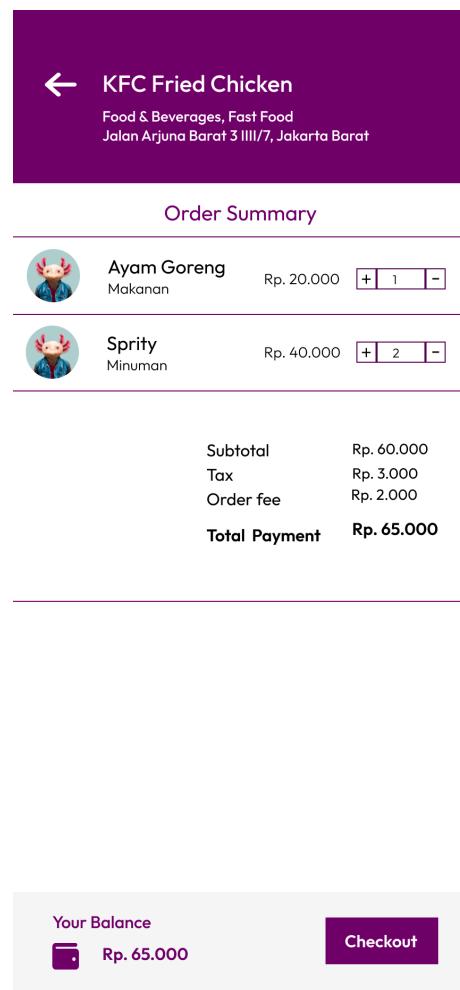
### 3.3.3.1.6 Halaman Menu



Gambar 3.58 User Interface Halaman Menu

Pada page ini customer dapat melihat data merchant dan list menu/produk yang dimiliki oleh setiap merchant, pada page ini customer juga dapat menambahkan produk kedalam cart.

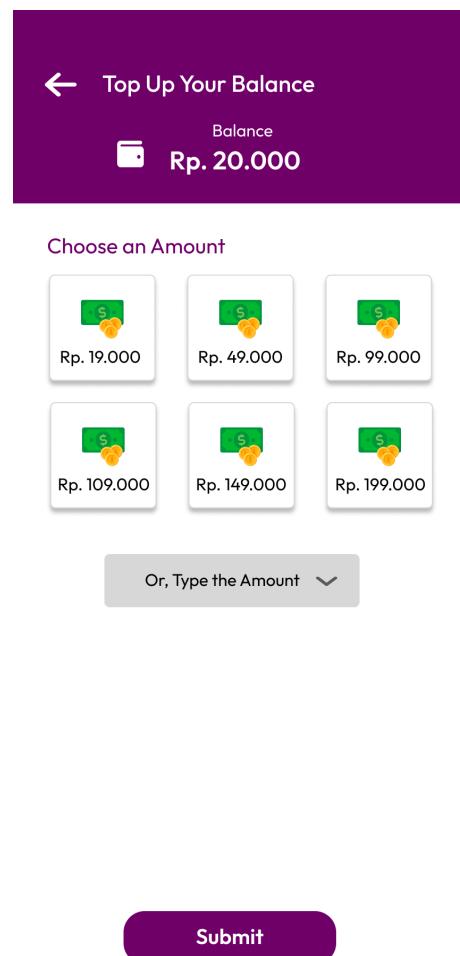
### 3.3.3.1.7 Halaman Cart



Gambar 3.59 User Interface Halaman Cart

Pada halaman ini customer dapat melihat *order summary* dan *payment summary*, customer juga dapat melakukan pembayaran dengan menekan button ‘checkout’

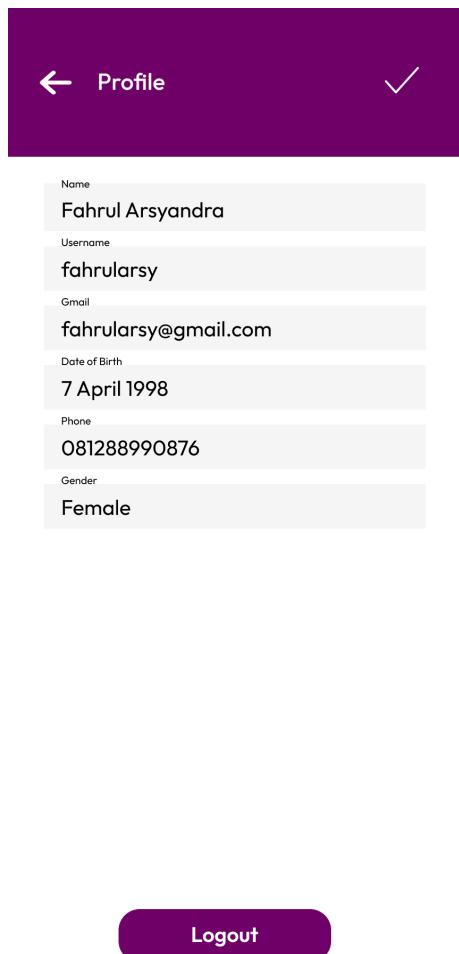
### 3.3.3.1.8 Halaman Topup



Gambar 3.60 User Interface Halaman Topup

Pada halaman ini customer dapat melihat balance dan melakukan *topup* akun dengan memilih saldo yang ingin di *topup*.

### 3.3.3.1.9 Halaman profile



Gambar 3.61 User Interface Halaman Profil

Pada halaman profil, customer dapat melakukan beberapa aksi yaitu mengubah data diri dengan menekan tombol checklist untuk menyimpan perubahan yang terjadi dan keluar dari aplikasi dengan menekan tombol logout.

### 3.3.3.2 User Merchant

#### 3.3.3.2.1 Halaman Register

The image shows a user interface for merchant registration. At the top, the word "Comander" is displayed in a purple font. Below it is a form with six input fields, each with a placeholder label above it:

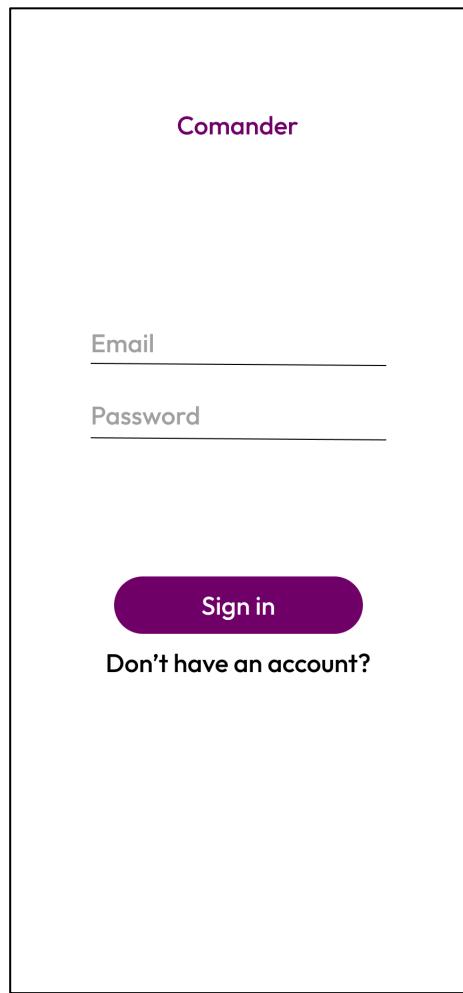
- Merchant name
- Gender
- Phone
- Username
- Email
- Password

Below the password field is another input field labeled "Confirm Password". At the bottom of the form is a purple button with the text "Sign up" in white.

Gambar 3.61 User Interface Halaman Register Merchant

User dengan role merchant harus mendaftarkan diri terlebih dahulu melalui halaman register agar data dapat tersimpan dalam database.

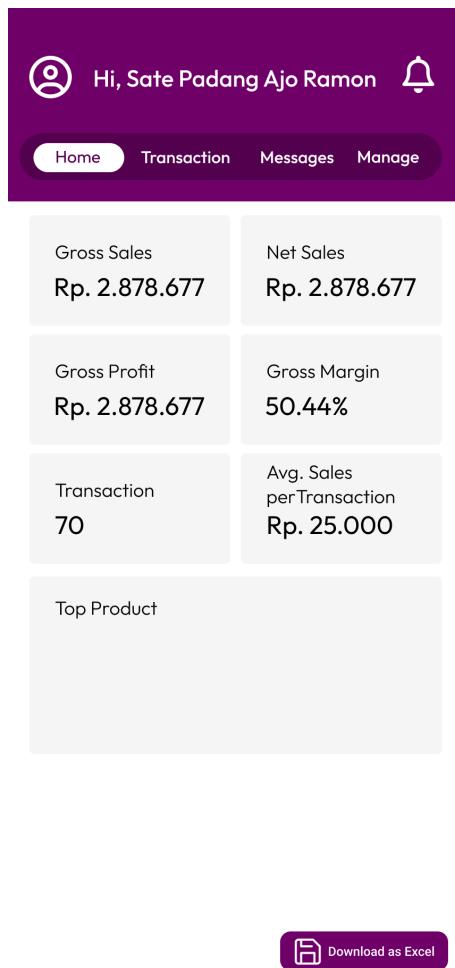
### 3.3.3.2.2 Halaman Login



Gambar 3.62 User Interface Halaman Login

Sebelum merchant dapat mengakses fitur-fitur yang terdapat dalam aplikasi, customer harus melakukan login melalui page Login yang terdapat dalam aplikasi ComanderApp.

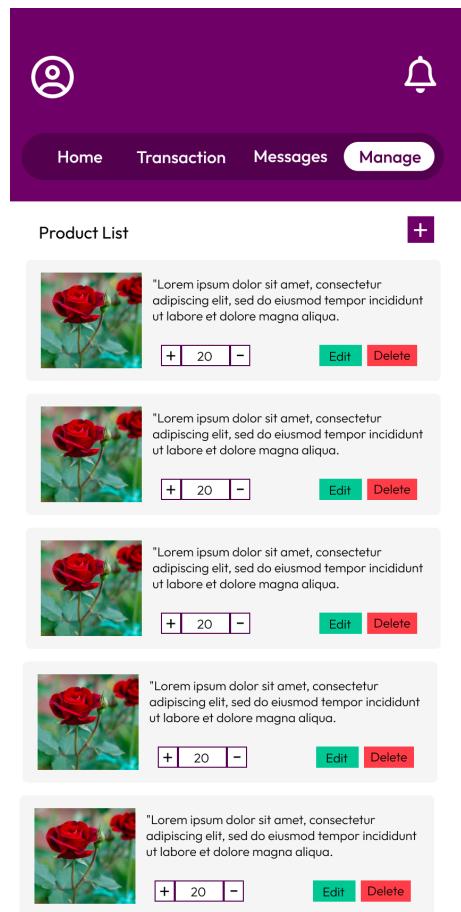
### 3.3.3.2.3 Halaman Home



Gambar 3.63 User Interface Halaman Home

Pada halaman ini merchant dapat melihat laporan penjualan yang berisi penjualan kotor, penjualan bersih, keuntungan kotor, margin kotor, total transaksi, rata-rata transaksi dan produk terbaik. Selain itu merchant juga dapat mengubah data tersebut menjadi file berupa excel dengan menekan tombol “Download as Excel”

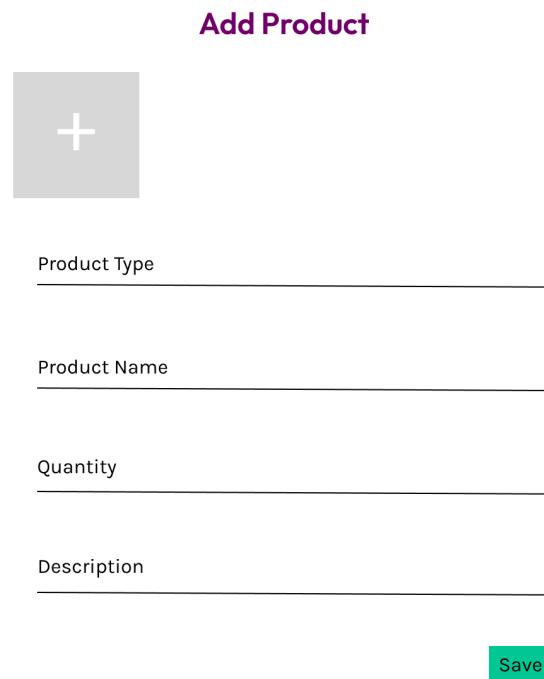
### 3.3.3.2.4 Halaman Produk



Gambar 3.64 User Interface Halaman Product

Pada halaman ini merchant dapat melihat produk/menu yang dimilikinya, selain itu merchant juga dapat menambahkan, mengubah dan menghapus produk.

### 3.3.3.2.5 Form Add Product



The image shows a user interface for adding a product. At the top center is a button labeled "Add Product". Below it is a large gray square button with a white plus sign (+) in the center. To the right of the plus sign is a vertical stack of input fields and a save button. The first field is labeled "Product Type" and has a horizontal line below it. The second field is labeled "Product Name" and also has a horizontal line. The third field is labeled "Quantity" and has a horizontal line. The fourth field is labeled "Description" and has a horizontal line. To the right of the "Description" field is a green rectangular button with the word "Save" in white.

Gambar 3.65 User Interface Form Add Product

Jika merchant ingin menambahkan produk kedalam menu maka merchant dapat menekan tombol ‘+’ pada halaman produk, maka sistem akan menampilkan form add produk. Merchant dapat mengisi data produk baru dan menekan tombol ‘save’ untuk menyimpan data produk baru kedalam database

### 3.3.3.2.6 Form Edit Product

**Edit Product**

  
[Edit photo](#)

---

Product Type

---

Product Name

---

Quantity

---

Description

---

Stock

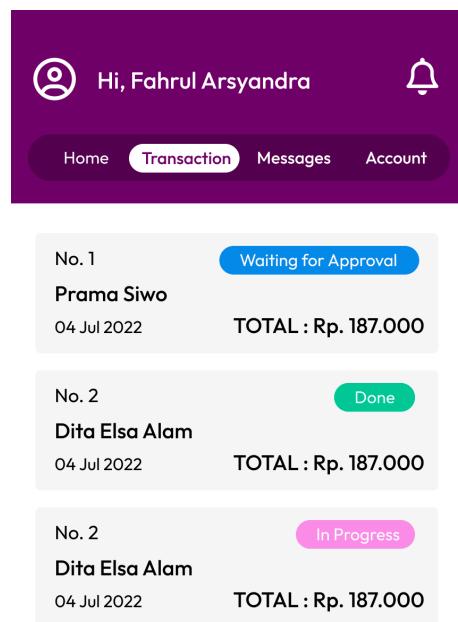
---

[Save](#)

Gambar 3.66 User Interface Form Edit Product

Merchant juga dapat melakukan pada produk yang telah ada dengan menekan tombol ‘edit’ pada halaman produk, kemudian sistem akan mengarahkan merchant ke form edit produk. Data produk akan terubah setelah merchant menekan tombol ‘save’.

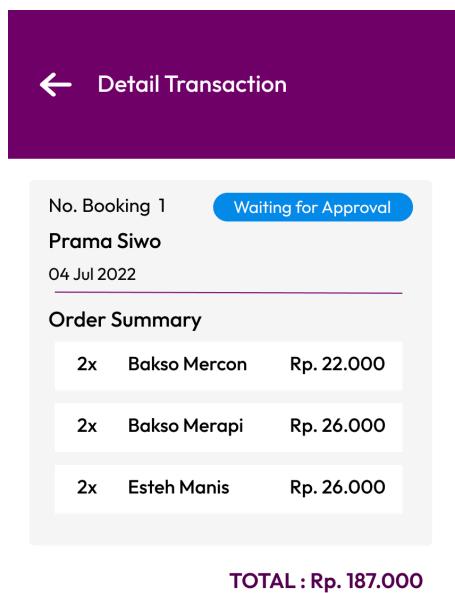
### 3.3.3.2.7 Halaman Transaksi



Gambar 3.67 User Interface Halaman Transaksi

Pada halaman transaksi merchant dapat melihat semua transaksi yang terjadi dan informasi-informasi lainnya seperti status tiap transaksi, nomor transaksi, nama pembeli, tanggal dilakukannya transaksi dan total harga.

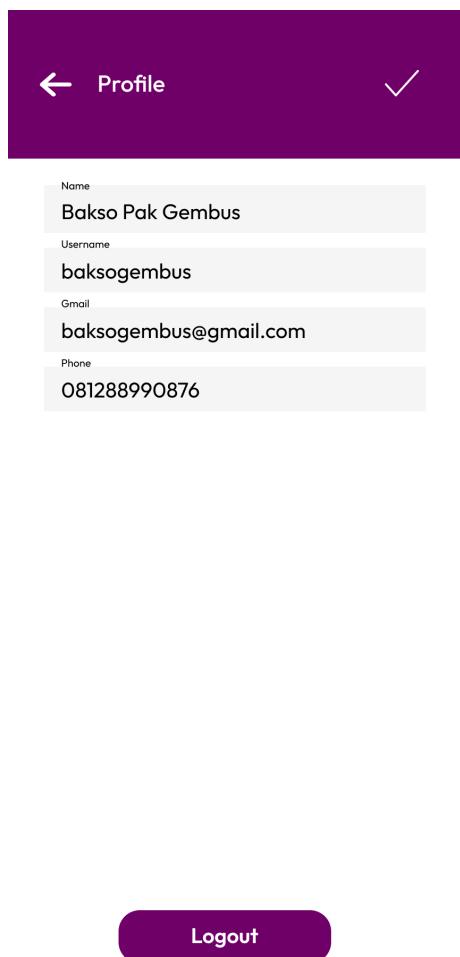
### 3.3.3.2.8 Halaman Detail Transaksi



Gambar 3.68 User Interface Halaman Detail Transaksi

Merchant dapat melihat halaman detail transaksi setelah menekan card transaksi yang terdapat pada halaman transaksi. Pada halaman ini akan terdapat detail informasi transaksi seperti nomor transaksi, status transaksi, nama pembeli, tanggal transaksi, daftar barang, kuantiti barang, harga barang dan total harga.

### 3.3.3.2.9 Halaman profile



Gambar 3.69 User Interface Halaman Profil

Pada halaman profil merchant dapat melakukan beberapa aksi diantara mengubah data diri dengan menekan tombol checklist untuk menyimpan perubahan yang terjadi dan keluar dari aplikasi dengan menekan tombol ‘logout’.

## **BAB 4.**

### **HASIL DAN PEMBAHASAN**

#### 4.1 Spesifikasi *System*

Setiap aplikasi tentunya memiliki spesifikasi minimum yang perlu dipenuhi oleh calon pengguna agar dapat menggunakan sebuah aplikasi secara optimal. Dalam spesifikasi sistem terdapat 2 bagian, yaitu spesifikasi perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*) sebagai berikut:

##### 4.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras pada aplikasi “ComanderApp” adalah sebagai berikut:

1. Layar sentuh (*touchscreen*)
2. *Random Access Memory* (RAM) minimal 2 GB
3. *Memory* berukuran setidaknya 15 MB
4. Memiliki koneksi internet

##### 4.1.2 Spesifikasi Perangkat Lunak

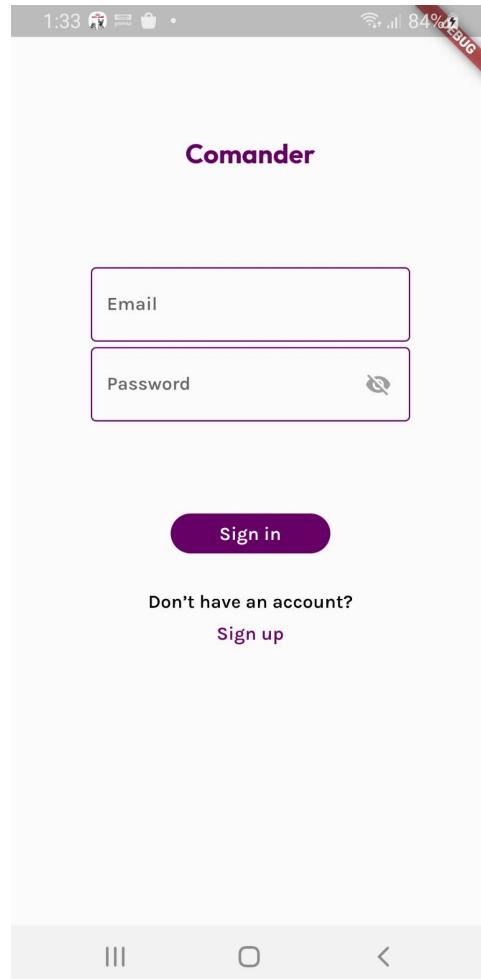
Spesifikasi perangkat lunak pada aplikasi “ComanderApp” adalah sistem operasi *android* dengan versi minimal 8.0 (*oreo*).

#### 4.2 Prosedur Penggunaan Aplikasi

Berikut prosedur penggunaan aplikasi dan penjelasan alur kerja aplikasi berbasis mobile “ComanderApp” adalah sebagai berikut:

#### 4.2.1 Prosedur Untuk Customer

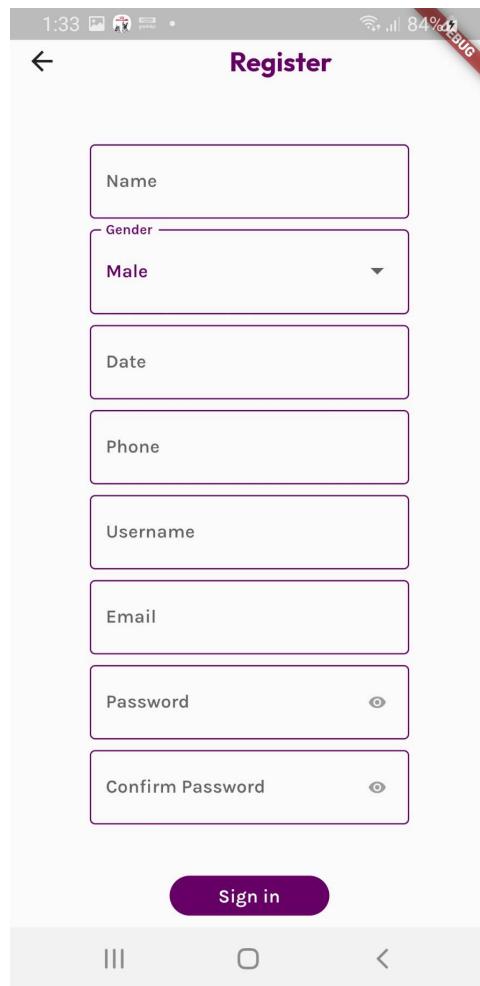
##### 4.2.1.1 Login



Gambar 4.1 Screenshot Halaman Login

Merupakan halaman utama saat user menggunakan aplikasi. Halaman ini memiliki 2 tombol yaitu, “Sign in” dan “Sign up”. Tombol “Sign in” berguna untuk mengarahkan user menuju halaman utama setelah user berhasil memasukan akun yang sudah terverifikasi, sedangkan tombol “Sign up” berguna untuk mengarahkan user ke halaman registrasi.

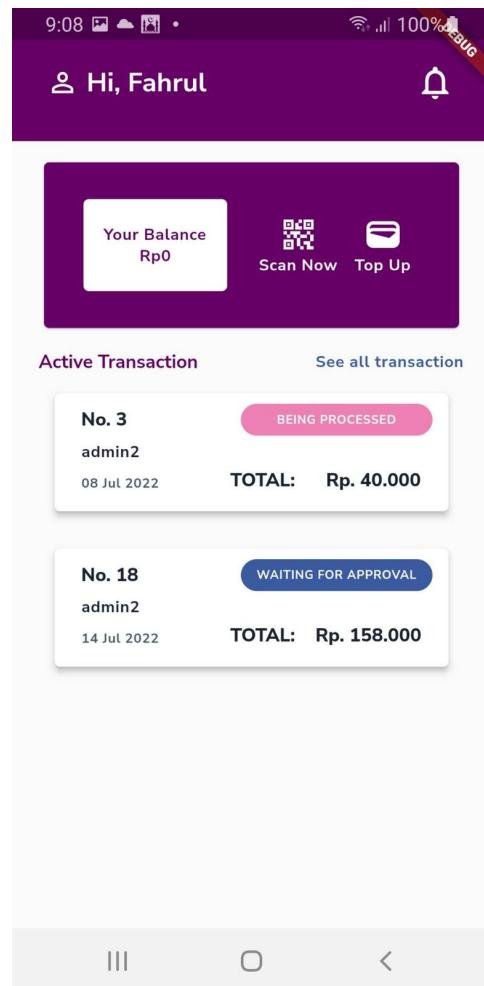
#### 4.2.1.2 Sign Up



Gambar 4.2 Screenshot Halaman Sign Up

User yang memilih untuk melakukan registrasi sebagai customer akan diarahkan ke halaman register untuk customer dengan mengisi data diri seperti nama, jenis kelamin, tanggal lahir, nomor telepon, username, email password dan konfirmasi password. Jika user sudah mengisi data diri dan menekan tombol “Sign in” maka user akan menerima email verifikasi dan akan diarahkan kembali ke halaman login untuk masuk ke dalam aplikasi setelah berhasil melakukan verifikasi email.

#### 4.2.1.3 Halaman Home

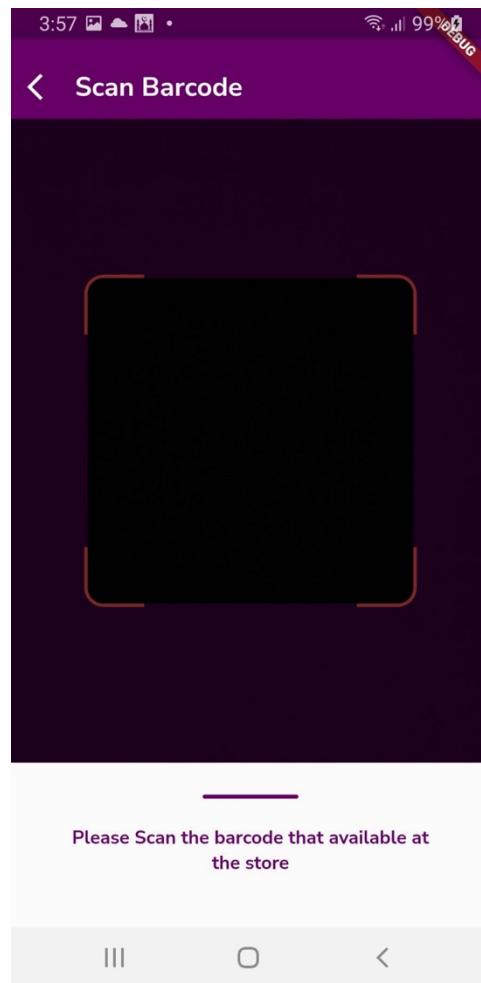


Gambar 4.3 Screenshot Halaman Home Customer

Halaman home terdiri dari header yang berisikan *button* akun, nama pengguna. Apabila user menekan *button* akun akan diarahkan ke halaman profile. Selanjutnya terdapat *card your balance* sebagai informasi saldo yang dimiliki oleh user. Selanjutnya ada *button scan now* yang akan mengarahkan ke halaman scan barcode merchant dan *button top up* yang mengarahkan ke halaman *top up balance* apabila ditekan oleh user. Halaman ini juga memiliki list transaksi user yang masih aktif di proses oleh merchant pada bagian *Active Transaction*. Transaksi yang aktif tersebut adalah transaksi yang statusnya *Waiting for Approval* dan *Being Processed*. Apabila salah satu *card*-nya ditekan

akan mengarahkan ke halaman detail transaksi tersebut. Untuk melihat semua transaksi dapat menekan *text button* “*See all transaction*”.

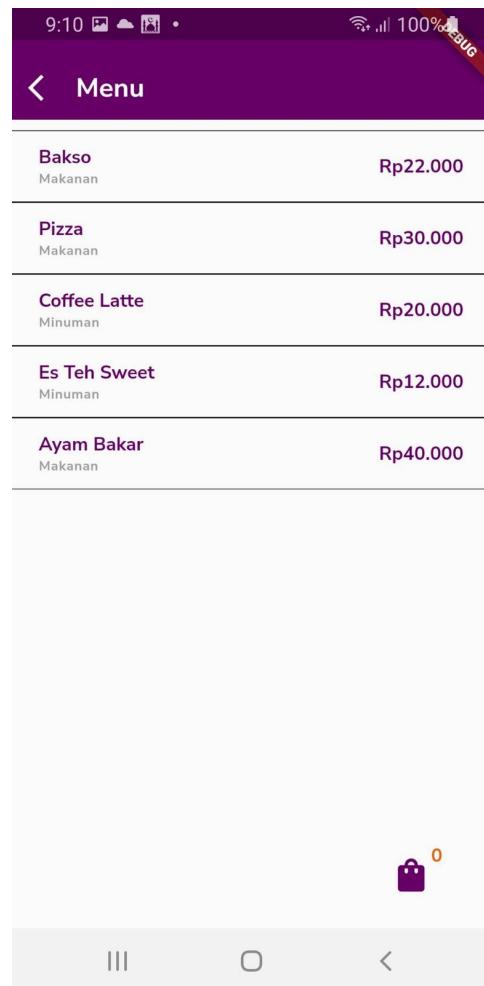
#### 4.2.1.4 Halaman Scan Barcode



Gambar 4.4 Screenshoot Halaman Scan Barcode

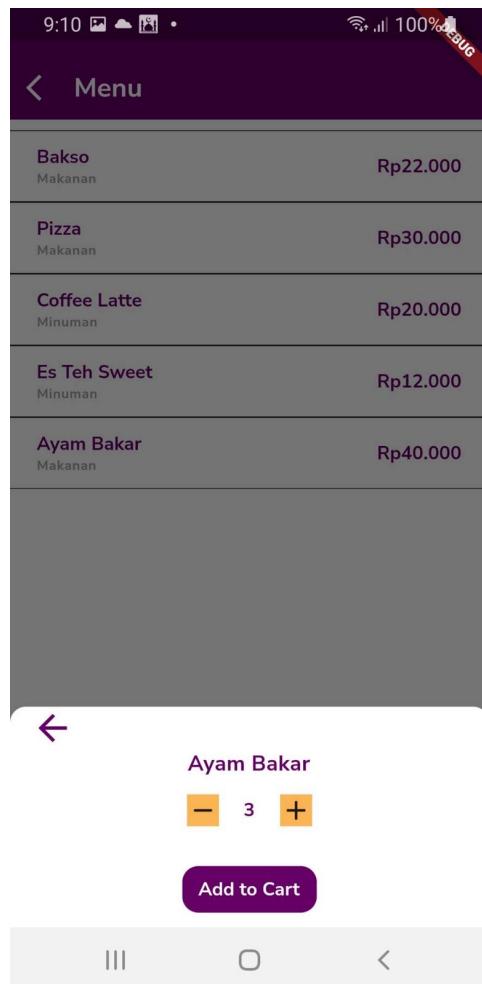
Pada halaman Qr berfungsi untuk customer memperoleh menu dari merchant dengan melakukan *scan barcode* pada *merchant* yang bersangkutan. Apabila barcode berhasil maka akan di arahkan ke halaman *menu merchant*

#### 4.2.1.5 Halaman Menu



Gambar 4.5 Screenshoot Halaman Daftar Menu Merchant

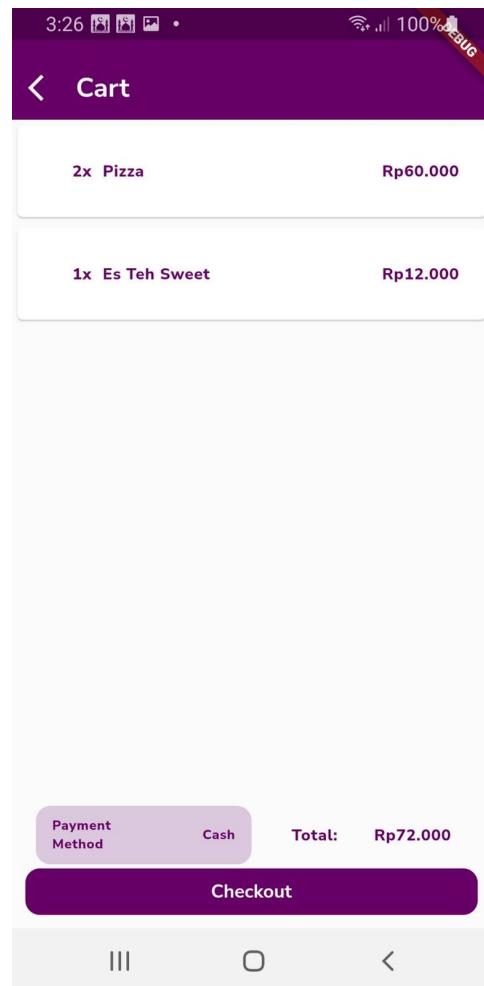
Halaman *menu merchant* terdiri dari daftar *menu merchant* dan tombol *cart*. Cart tidak akan mengarahkan ke halaman *cart* apabila data belum ada data yang dimasukkan



Gambar 4.6 Screenshot Bottom Sheet Quantity Produk

Bottom Sheet muncul setelah customer ingin memasukin jumlah produk yang ingin di pesan kedalam cart

#### 4.2.1.6 Halaman Cart



Gambar 4.7 Screenshoot Halaman Cart

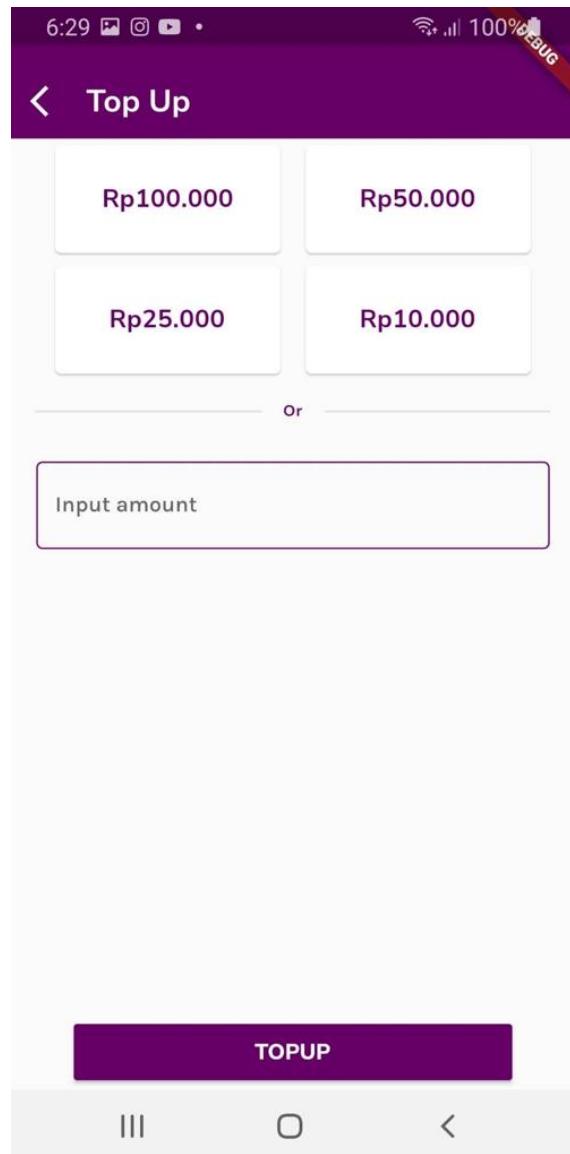
Halaman cart tediri dari Card product yang berisikan jumlah, nama dan harga pesanan. Di bawah terdapat Card payment method yang brfungsi untuk memilih metode pembayaran. Lalu tedapat total pesanan dan terakhir yaitu *button checkout* untuk finalisasi pemesanan.



Gambar 4.8 Screenshot Bottom Sheet Payment Method

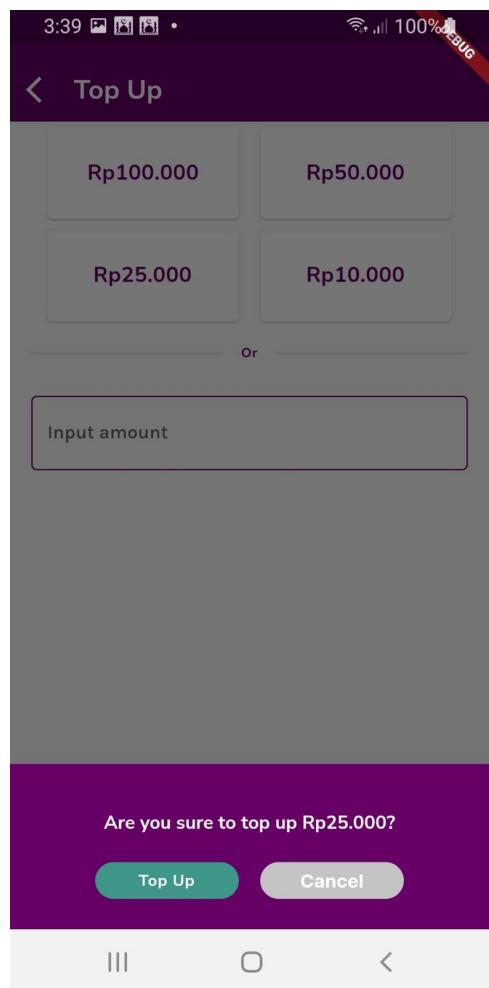
Bottom Sheet muncul setelah customer menekan *card payment method*. Customer dapat memilih metode pembayaran untuk order. Namun apabila E-Wallet tidak mencukupi, maka E-Wallet tidak bisa di tekan

#### 4.2.1.7 Halaman Top Up



Gambar 4.9 Screenshoot Halaman Top up Saldo

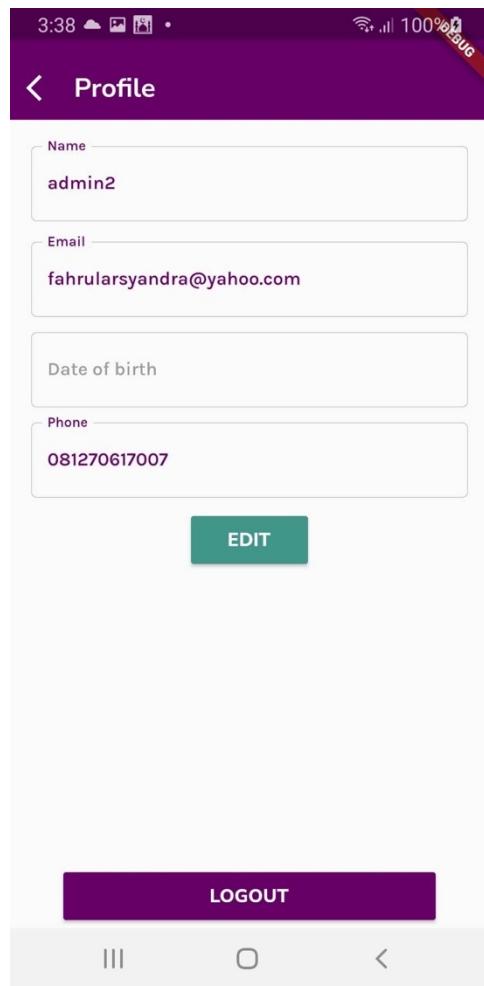
Halaman top up terdiri dari card jumlah yang ingin di top up dan input form jumlah yang ingin di top up. Namun customer dapat memasukkan jumlah amount yang diinginkan, lalu menekan tombol top up untuk finalisasi top up



Gambar 4.10 Bottom Sheet Validasi Top up Saldo

*Bottom sheet* keluar setelah costumer menekan card nominal jumlah yang ingin ditambahkan.

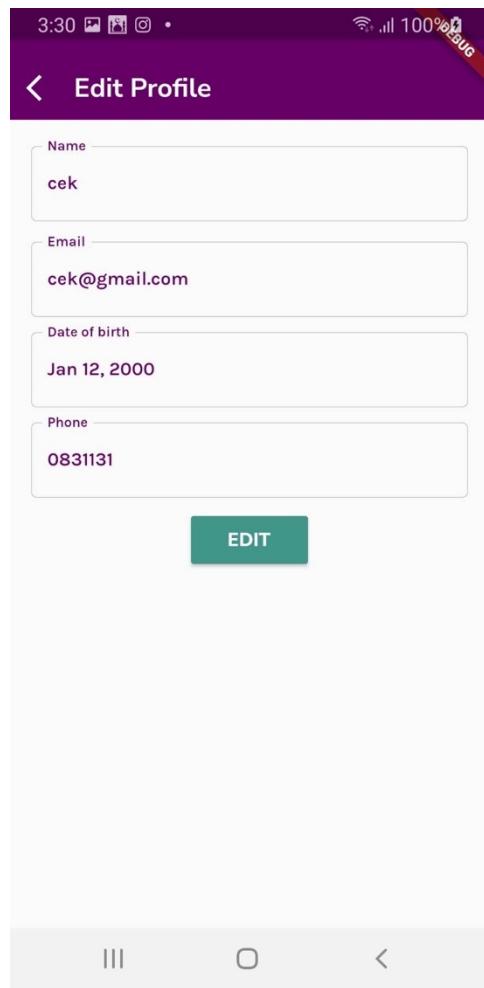
#### 4.2.1.8 Halaman Profil



Gambar 4.11 Screenshot Halaman Profile Customer

Pada halaman profil user dapat melihat data diri yang tersimpan dalam database, selain itu user juga dapat mengubah data diri dengan menekan *button* edit dan user akan diarahkan ke halaman edit profil, user juga dapat mengeluarkan akun dari aplikasi dengan menekan *button* logout

#### 4.2.1.9 Halaman Edit Profil

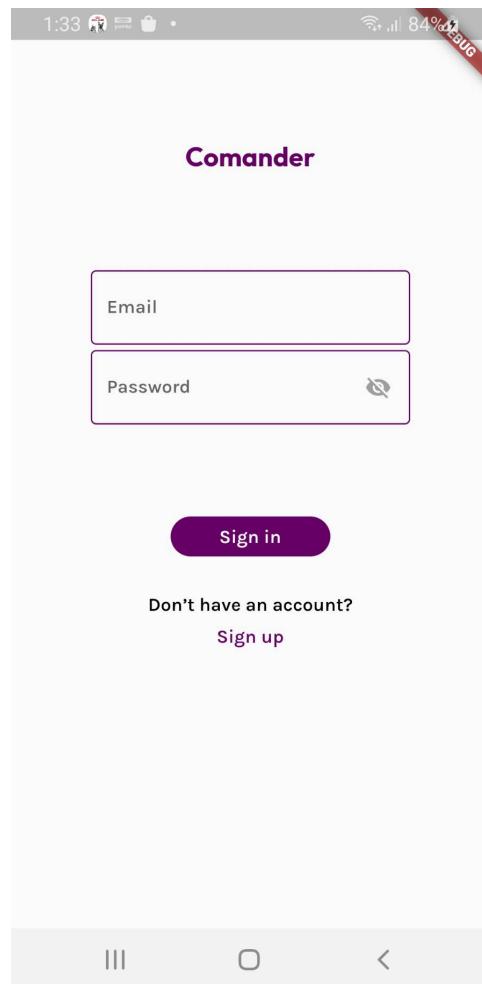


Gambar 4.12 Screenshot Halaman Edit Profile Customer

Pada halaman edit profil user dapat mengubah data diri dan menekan button edit untuk menyimpan data diri baru yang telah diubah kedalam database.

#### 4.2.2 Prosedur Untuk Merchant

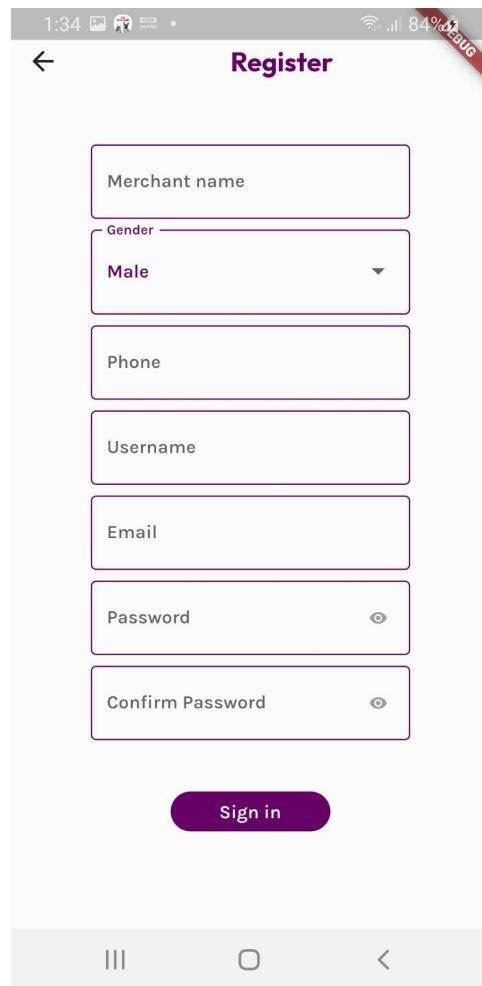
##### 4.2.2.1 Login



Gambar 4.13 Screenshoot Halaman Login

Merupakan halaman utama saat user menggunakan aplikasi. Halaman ini memiliki 2 tombol yaitu, “Sign in” dan “Sign up”. Tombol “Sign in” berguna untuk mengarahkan user menuju halaman utama setelah user berhasil memasukan akun yang sudah terverifikasi, sedangkan tombol “Sign up” berguna untuk mengarahkan user ke halaman registrasi.

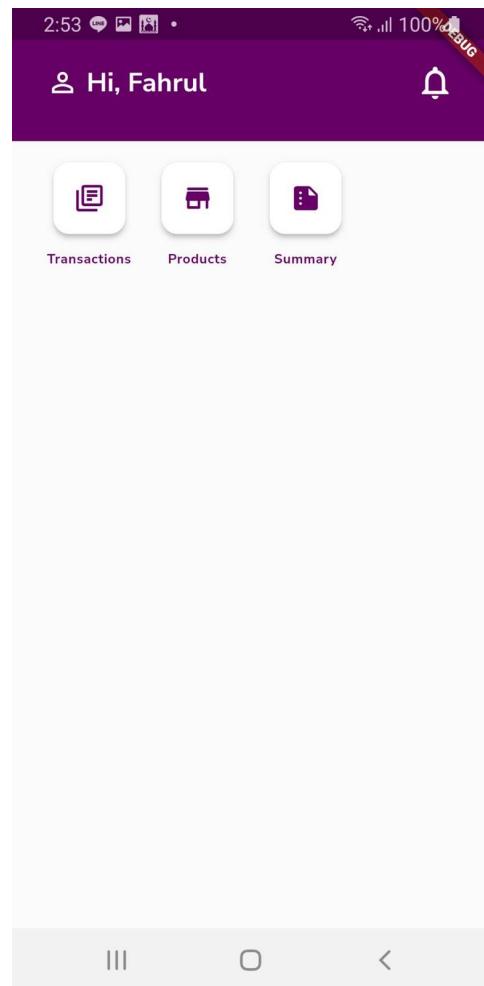
#### 4.2.2.2 Sign up



Gambar 4.14 Screenshot Halaman Register

User yang memilih untuk melakukan registrasi sebagai *merchant* akan diarahkan ke halaman register untuk *merchant* dengan mengisi data diri seperti nama, jenis kelamin, tanggal lahir, nomor telepon, username, email password dan konfirmasi password. Jika user sudah mengisi data diri dan menekan tombol “Sign in” maka user akan menerima email verifikasi dan akan diarahkan kembali ke halaman login untuk masuk ke dalam aplikasi setelah berhasil melakukan verifikasi email.

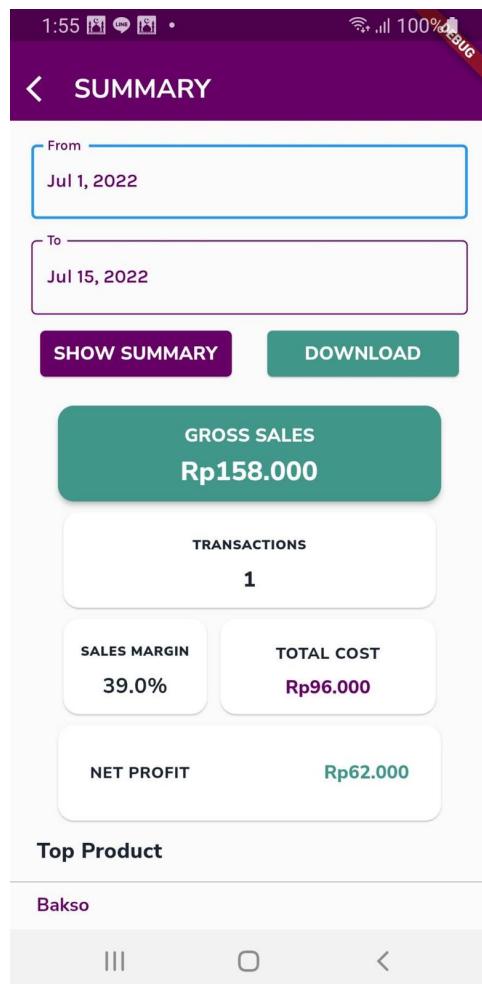
#### 4.2.2.3 Halaman Home



Gambar 4.15 Screenshot Halaman Merchant Dashboard

Halaman home terdiri dari header yang berisikan *button* akun dan nama pengguna. Apabila user menekan *button* akun akan diarahkan ke halaman profile. Selanjutnya pada halaman home *merchant* juga terdapat 3 *card button* yaitu; *card button transaction* yang akan mengarahkan *merchant* ke halaman transaksi, *card button product* yang akan mengarahkan *merchant* ke halaman produk dan *card button summary* yang akan mengarahkan merchant ke halaman *summary* penjualan

#### 4.2.2.4 Halaman Summary

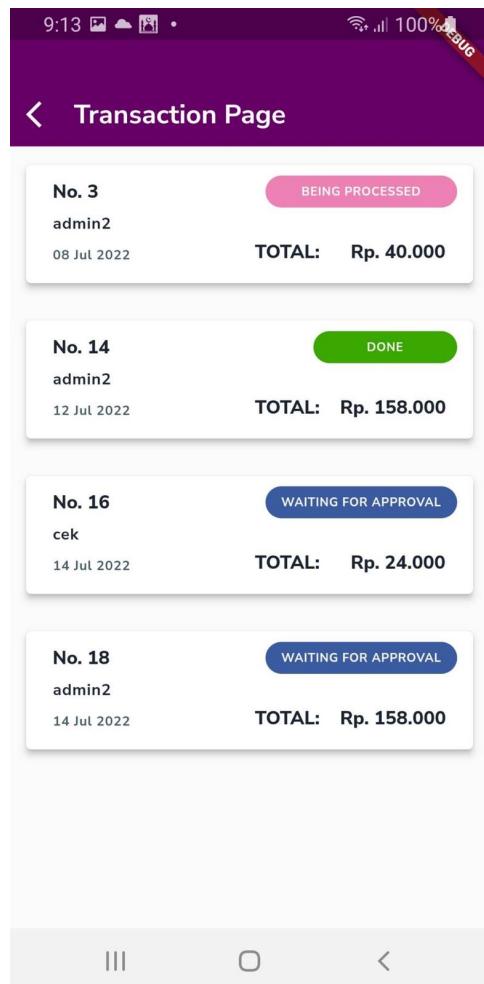


Gambar 4.16 Screenshot Halaman Summary

Halam *summary* merupakan halaman yang memberikan informasi laporan penjualan untuk jangka waktu tertentu yang ditentukan yaitu total transaksi, *gross sales*, *margin sales*, *total cost*, *net profit* dan produk yang terjual terbanyak. Pada awalnya, halaman ini akan menampilkan data sesuai dengan hari halaman tersebut dibuka. Namun merchant dapat menentukan dari dan sampai kapan ingin melihat informasi laporan dengan memasukkan tanggal pada form *from* yang merupakan tanggal awal, dan *to* yang merupakan tanggal akhir.

Lalu button *SHOW SUMMARY* untuk *request summary* berdasarkan tanggal yang telah di input. Lalu button *DOWNLOAD* untuk download data summary tersebut.

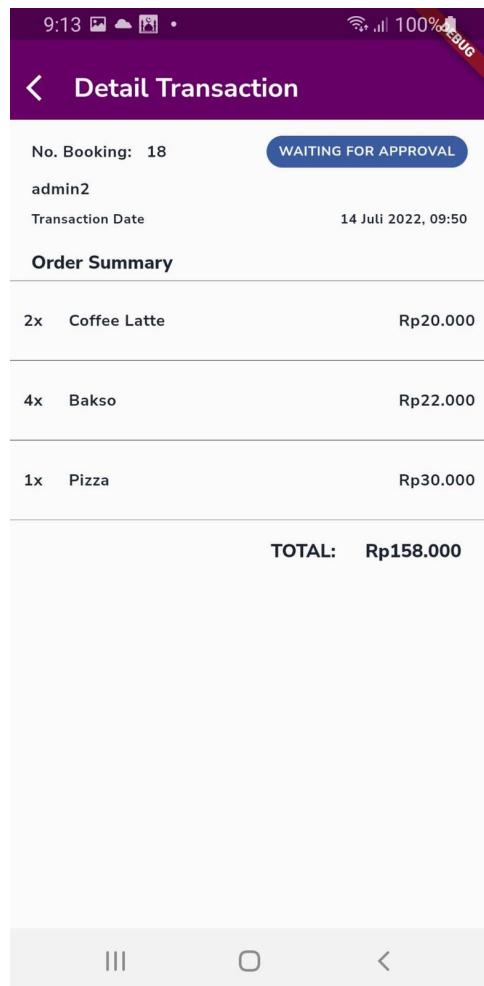
#### 4.2.2.5 Halaman Transaksi



Gambar 4.17 Screenshoot Halaman Transaksi Merchant

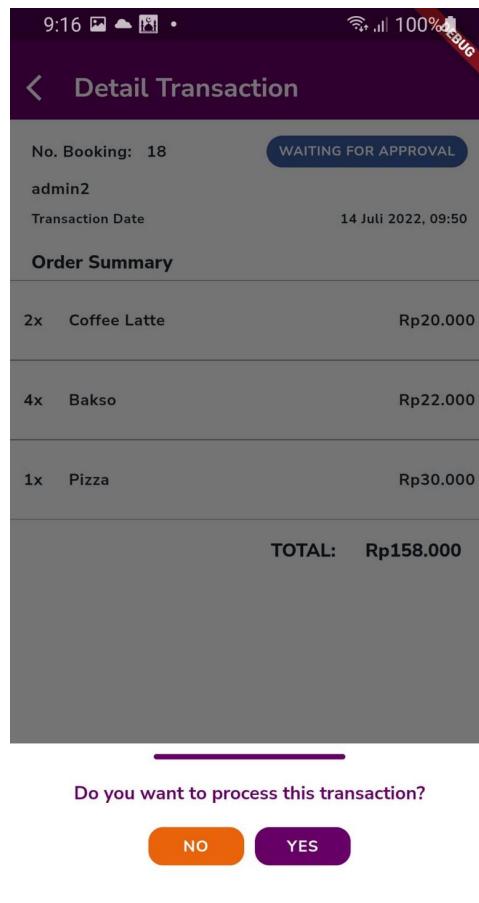
Halaman transaksi terdiri dari list daftar transaksi yang terjadi pada merchant tersebut. Setiap card terdiri dari informasi tentang status transaksi, total transaksi dan customer yang memesan.

#### 4.2.2.6 Halaman Detail Transaksi



Gambar 4.18 Screenshoot Halaman Detail transaksi

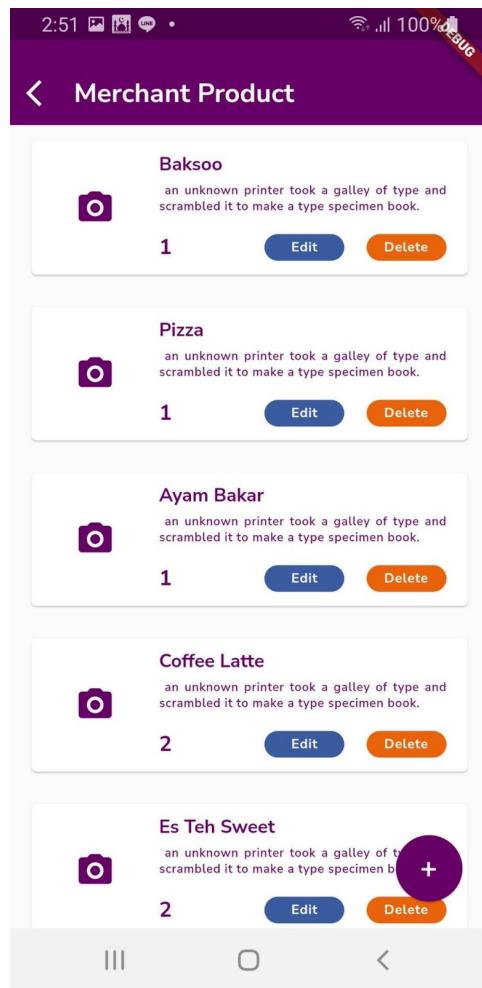
Halaman detail transaksi memiliki informasi detail transaksi customer yang berisikan status transaksi, nama transaksi, tanggal transaksi, list pesanan dan total. Status transaksi merupakan sebuah button yang berfungsi untuk merchant melakukan update status transaksi.



Gambar 4.19 Screenshot Bottom Sheet Update Status Transaksi

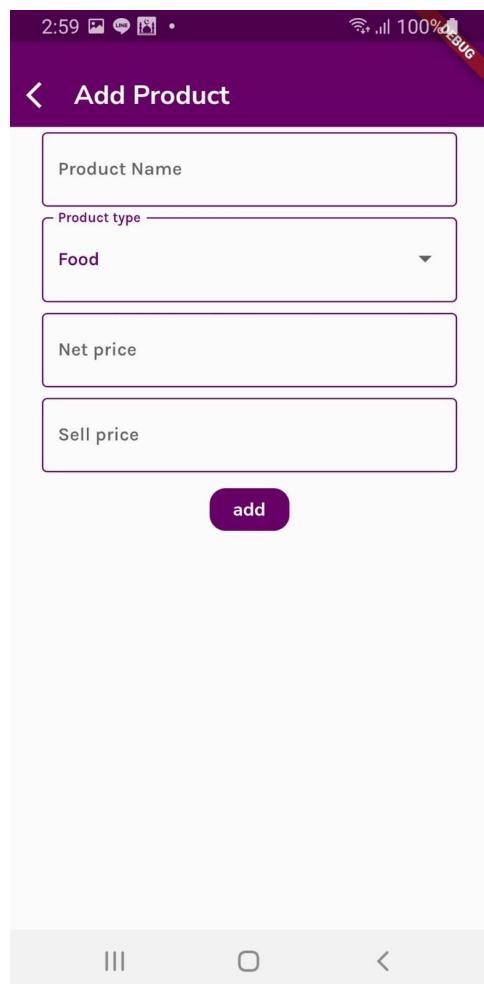
*Bottom sheet* keluar setelah user ingin melakukan check out sebagai validasi apakah user ingin melanjutkan proses ini.

#### 4.2.2.7 Halaman Produk



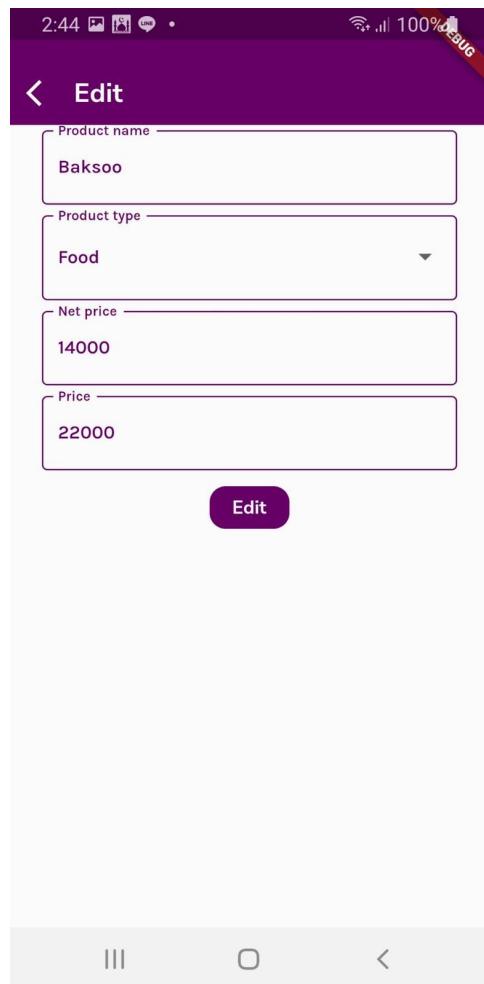
Gambar 4.20 Screenshot Halaman Product Merchant

Halaman produk berisikan daftar produk-produk yang dimiliki oleh merchant. *Merchant* dapat menambah produk yang akan di jual dengan menekan *floating button* dengan ikon “+”. *button edit* berfungsi apabila *merchant* ingin update produk yang dimiliki. *Button delete* berfungsi untuk *merchant* yang ingin menghapus produk



Gambar 4.21 Screenshoot Halaman Add Product Merchant

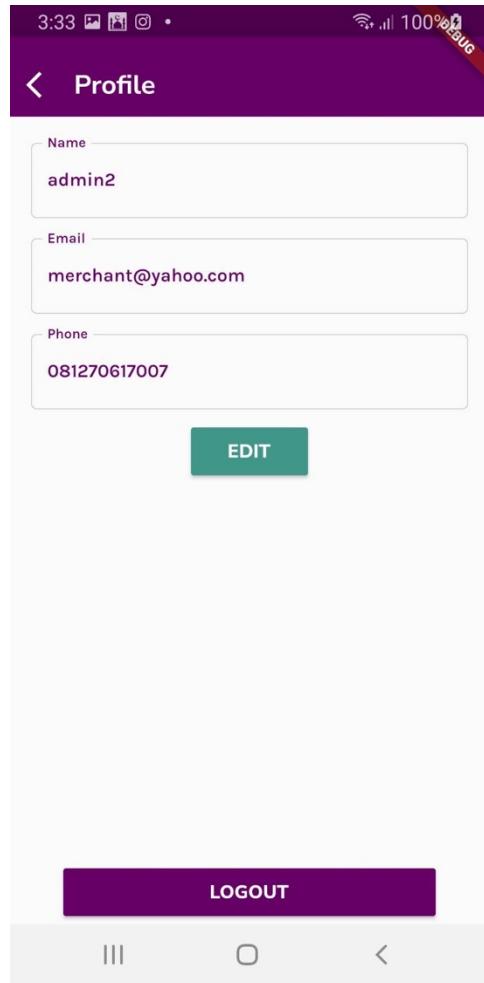
Halaman *add product* merupakan halaman yang dimiliki merchant yang ingin menambah produk. Merchant dapat mengisi pada form tersebut lalu menekan *button add*



Gambar 4.22 Screenshoot Halaman Edit Produk Merchant

Edit produk merupakan halaman merchant untuk melakukan edit produk yang dimiliki oleh merchant. Setelah melakukan edit, merchant menekan tombol edit untuk finalisasi data terbaru pada produk tersebut.

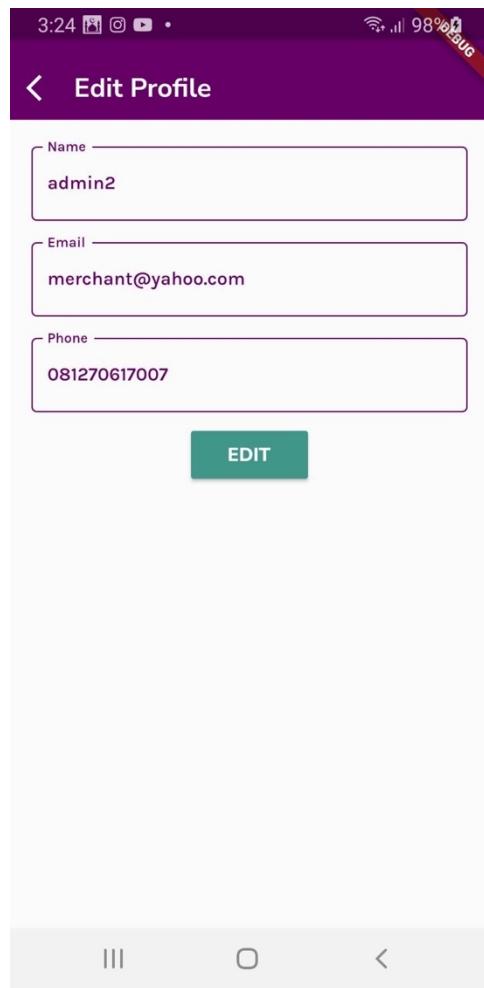
#### 4.2.2.8 Halaman Profil



Gambar 4.23 Screenshot Halaman Profile Merchant

Pada halaman profil merchant dapat melihat data diri yang tersimpan dalam database, selain itu merchant juga dapat mengubah data diri dengan menekan *button* edit dan merchant akan diarahkan ke halaman edit profil, merchant juga dapat mengeluarkan akun dari aplikasi dengan menekan *button* logout

#### 4.2.2.9 Halaman Edit Profile

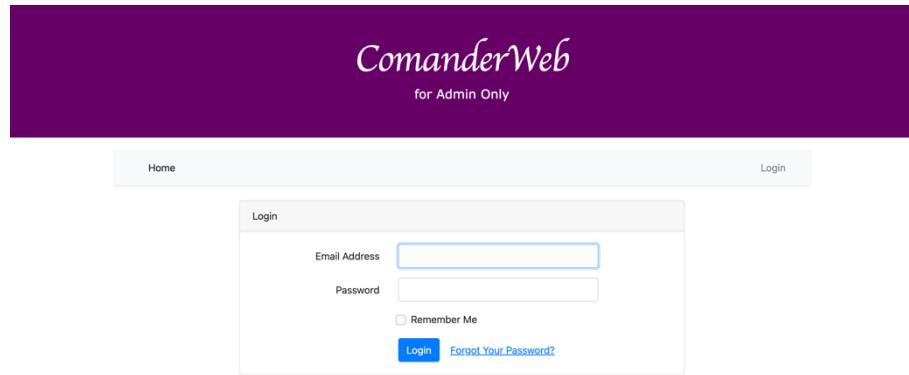


Gambar 4.24 Screenshoot Halaman Edit Profile Merchant

Pada halaman edit profil merchant dapat mengubah data diri dan menekan button edit untuk menyimpan data diri baru yang telah diubah kedalam database.

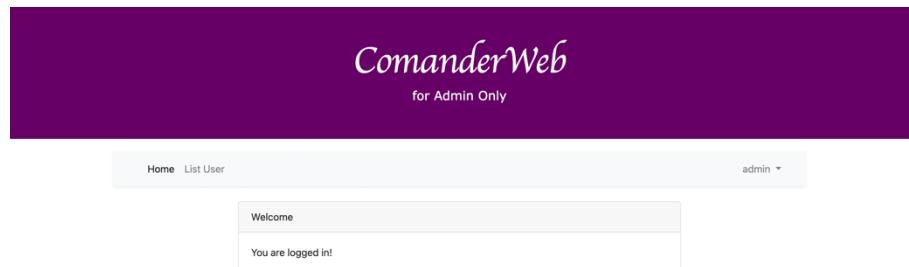
#### 4.2.3 Prosedur Untuk Admin

##### 4.2.3.1 Login



Gambar 4.25 Screenshot Halaman Login Admin

Admin harus melakukan login terlebih dahulu menggunakan email dan password yang sudah terdaftar dalam database.



Gambar 4.26 Screenshot Halaman Home

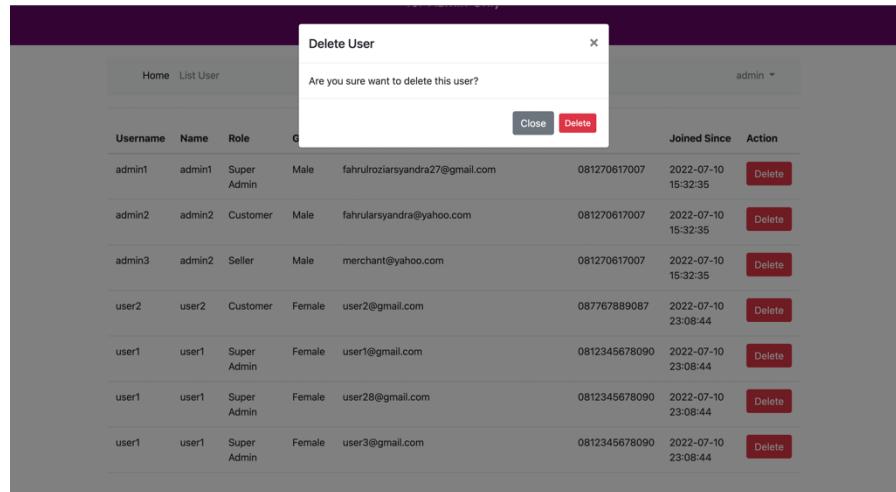
Admin yang berhasil melakukan login kedalam aplikasi akan masuk kehalaman home admin dan mendapat notifikasi “You are logged in!”.

#### 4.2.3.2 Delete User

Username	Name	Role	Gender	Email	Email Verified At	Phone	Joined Since	Action
admin1	admin1	Super Admin	Male	fahrulroziarsyandra27@gmail.com	081270617007	2022-07-10 15:32:35		<button>Delete</button>
admin2	admin2	Customer	Male	fahrulroziarsyandra@yahoo.com	081270617007	2022-07-10 15:32:35		<button>Delete</button>
admin3	admin2	Seller	Male	merchant@yahoo.com	081270617007	2022-07-10 15:32:35		<button>Delete</button>
user2	user2	Customer	Female	user2@gmail.com	087767889087	2022-07-10 23:08:44		<button>Delete</button>
user1	user1	Super Admin	Female	user1@gmail.com	0812345678090	2022-07-10 23:08:44		<button>Delete</button>

Gambar 4.27 Screenshot Halaman List User

Admin dapat melihat list user dengan menekan tombol “List User” yang terdapat pada navigation bar. Selanjutnya admin dapat menghapus user dengan menekan tombol “Delete” dan sistem akan menampilkan modal seperti gambar dibawah.



Gambar 4.28. Screenshot modal delete

## 4.3 Evaluasi

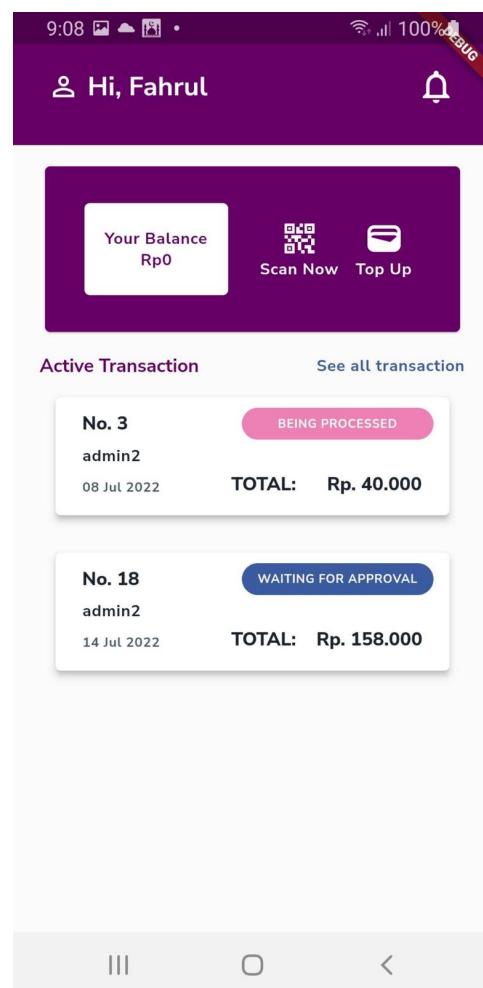
### 4.3.1 Evaluasi User Interface

#### 4.3.1.1 Evaluasi Berdasarkan 8 *Golden Rules*

Dalam pengembangan *user interface* aplikasi “ComanderApp” menggunakan 8 aturan emas sebagai acuan. Berikut evaluasi user interface “ComanderApp” berdasarkan 8 aturan emas:

##### *1. Strive for consistency*

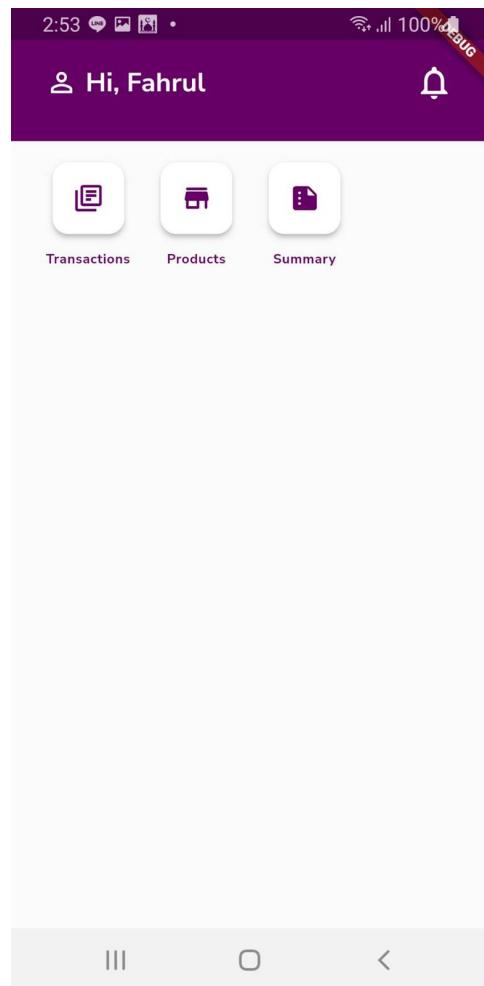
Pada aplikasi “ComanderApp” penerapan konsistensi *design* dapat dilihat dari penggunaan desain warna, *font*, *layout* yang konsisten pada setiap halamannya.



Gambar 4.29. Contoh penerapan Strive for Consistency

## 2. Seek universal usability

Tampilan aplikasi “ComanderApp” dirancang sehingga mudah dikenali baik oleh pemula yang baru saja belajar menggunakan aplikasi *mobile* hingga yang sudah terbiasa menggunakan aplikasi *mobile*. Seperti pada Gambar 4.30 dimana merchant dapat dengan mudah mengetahui fungsi dari masing icon-icon.

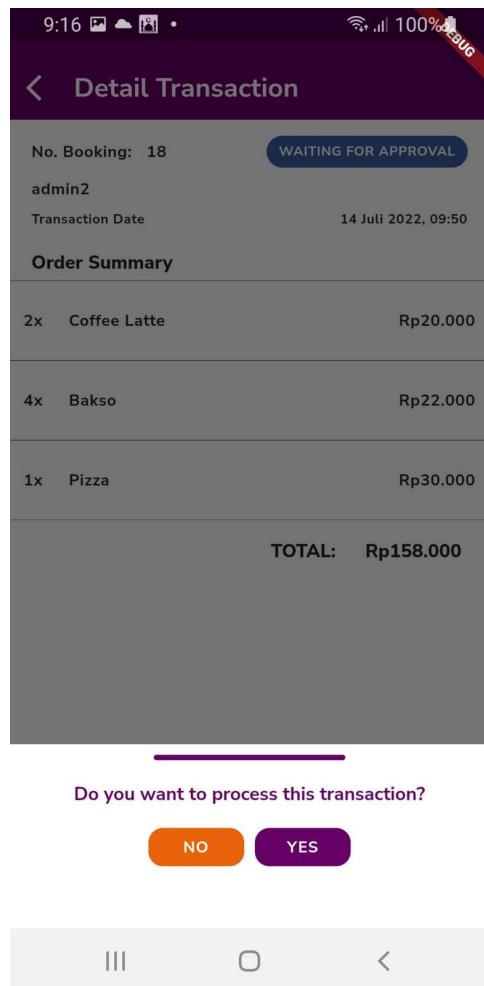


Gambar 4.30. Contoh penerapan Strive for Consistency

## 3. Offer informative feedback

Untuk setiap tindakan pengguna, harus ada yang menjadi umpan balik dari antarmuka, oleh karena itu aplikasi “ComanderApp” selalu memerikan umpan balik terhadap setiap perilaku yang dilakukan oleh pengguna. Contoh nya dapat dilihat

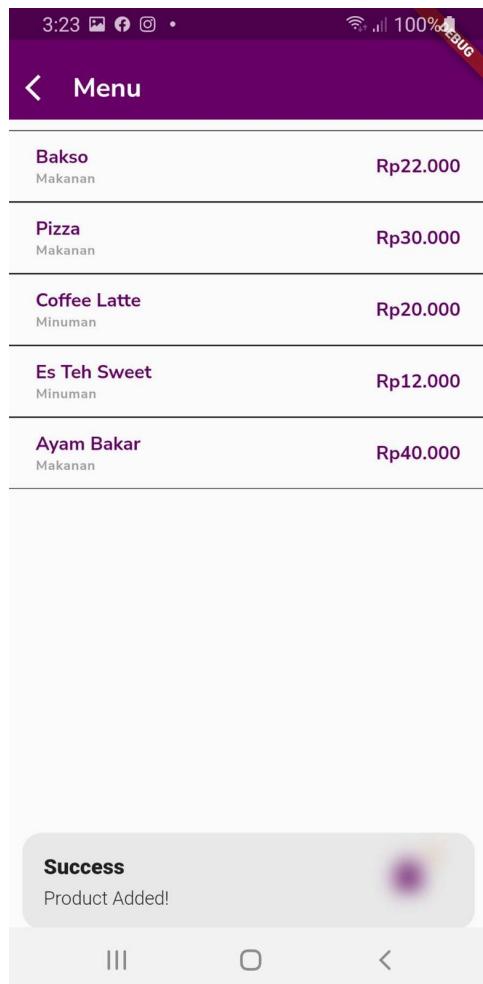
pada Gambar 4.31 dimana merchant mendapatkan notifikasi apakah ingin memproses transaksi tersebut.



Gambar 4.31. Contoh penerapan informative feedback

#### 4. Design dialog to yield closure

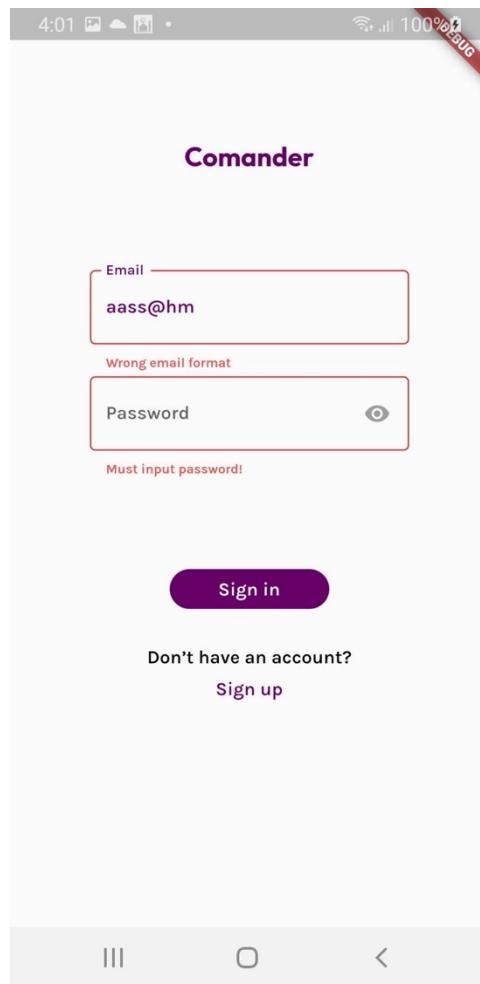
Umpulan yang informatif pada penyelesaian kelompok tindakan memberi pengguna kepuasan pencapaian, rasa lega. Oleh karena itu pada aplikasi “ComanderApp” akan menampilkan notifikasi untuk setiap aktifitas yang berhasil dilakukan pengguna, seperti pada Gambar 4.32 dimana notifikasi produk berhasil ditambahkan akan muncul apabila *customer* berhasil menambahkan produk kedalam *cart*.



Gambar 4.32. Contoh pesan pop up

##### 5. *Prevent error*

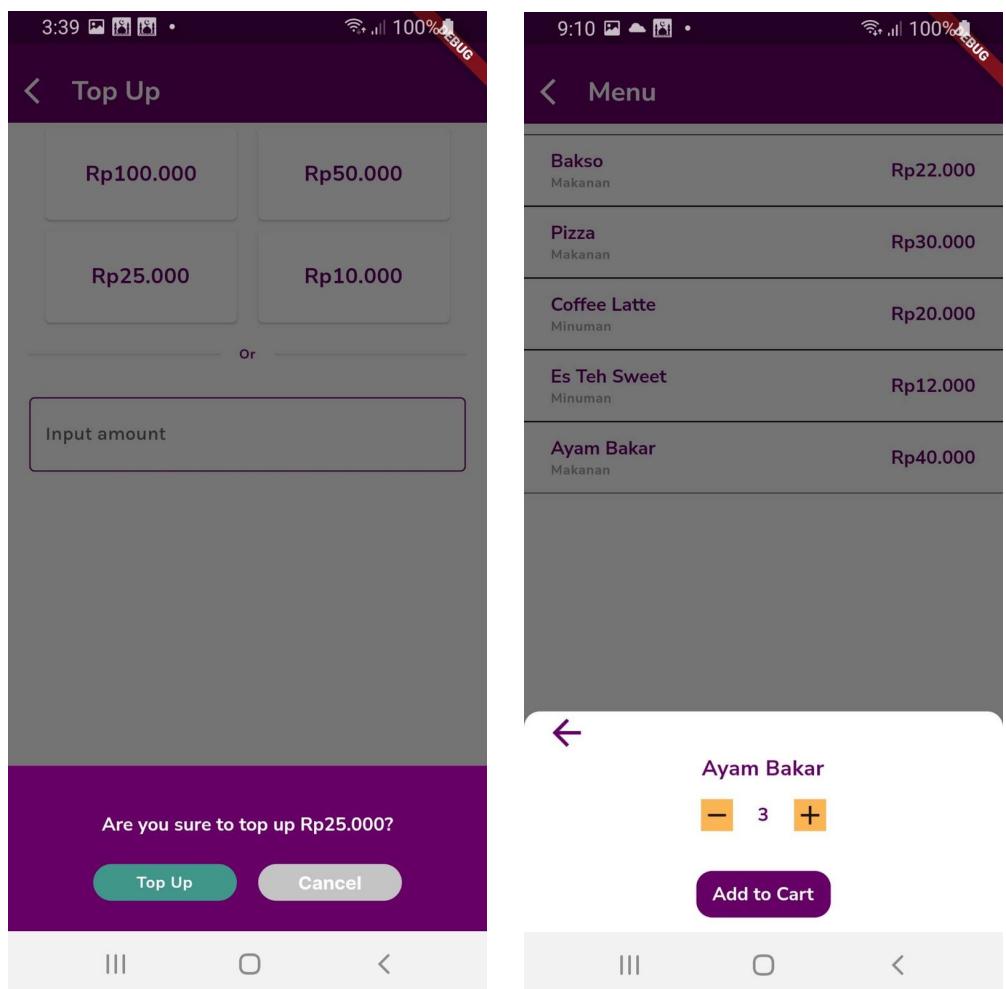
Antarmuka aplikasi “ComanderApp” dirancang sedemikian rupa untuk meminimalisir pengguna melakukan kesalahan yang serius dalam melakukan suatu aktifitas dengan menampilkan teks peringatan. Seperti yang terdapat pada Gambar 4.33 apabila user salah memberi inputan saat melakukan login maka aplikasi akan menampilkan teks peringatan.



Gambar 4.33. Contoh penerapan prevent error

6. *Permit easy reversal of action*

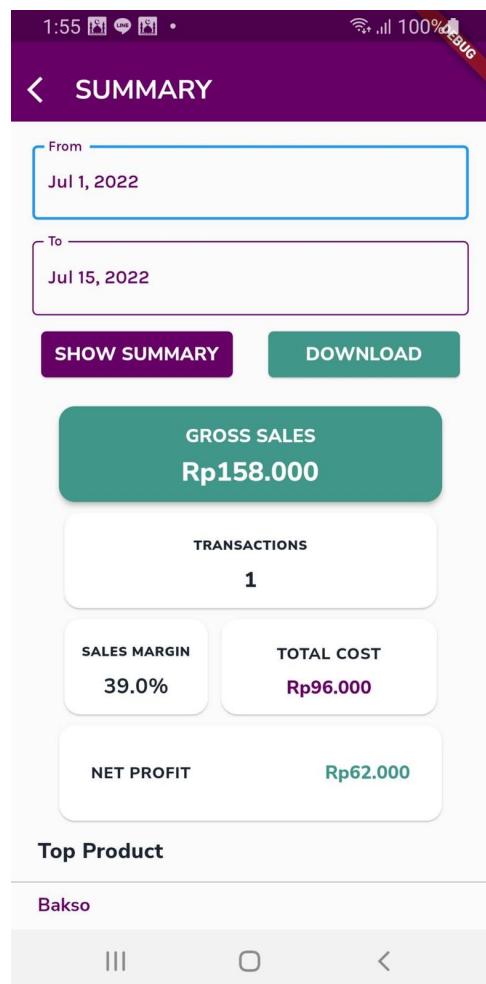
Sebisa mungkin tindakan harus dapat dibatalkan guna mengurangi kecemasan, karena pengguna tahu bahwa kesalahan dapat dibatalkan, dan mendorong eksplorasi pilihan yang tidak biasa. Aplikasi “ComanderApp” menyediakan tombol ‘cancel’ dan ‘<-’ untuk membatalkan sebuah Tindakan yang ingin dilakukan pengguna, seperti pada Gambar 4.34



Gambar 4.34. Contoh penerapan Strive for Consistency

#### 7. Keep users in control

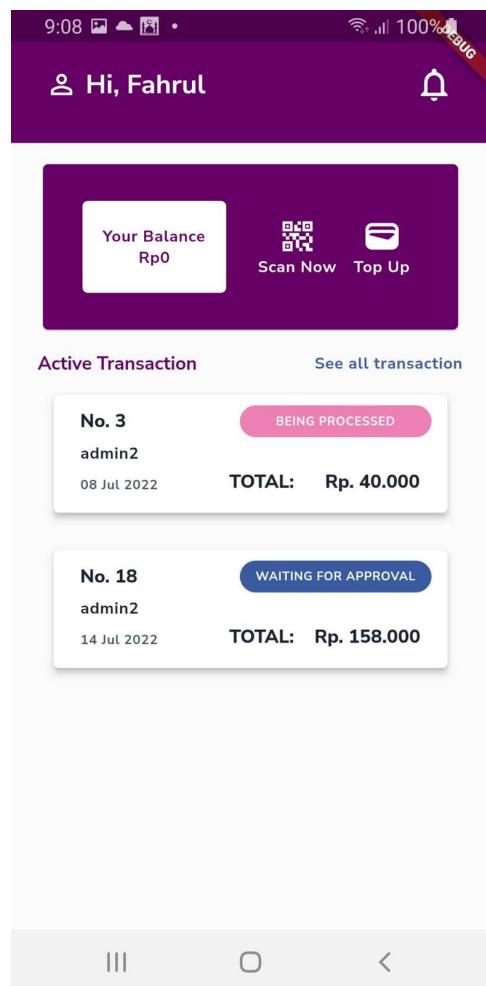
Aplikasi “ComanderApp” memberikan pengguna kebebasan dalam mengontrol aplikasi. Seperti pada Gambar 4.35 dimana user dapat memilih rentang waktu sendiri untuk menampilkan *summary* penjualan.



Gambar 4.35. Contoh keep user control

#### 8. Reduce short term memory load

Tampilan antarmuka aplikasi “ComanderApp” yang sederhana membuat pengguna mudah mengingat informasi yang diperlukan dalam menggunakan aplikasi. Seperti pada halaman home dimana *customer* dapat dengan mudah melihat daftar transaksi yang sedang berlangsung atau aktif.



Gambar 4.36. Contoh penerapan short term memory

#### 4.3.1.1 Evaluasi Berdasarkan 5 Faktor Manusia Terukur

Evaluasi yang dilakukan terhadap aplikasi “ComanderApp” yang mengacu pada prinsip 5 faktor manusia terukur yang dimana teorinya sudah dijelaskan pada bab sebelumnya seperti berikut;

##### 1. Time to learn

Waktu yang dibutuhkan untuk memahami sebuah fungsi-fungsi dari aplikasi memberikan dampak terhadap penilaian *user interface* sebuah aplikasi. Berdasarkan kuesioner yang dilakukan sebelumnya pada pertanyaan No. 2 dapat disimpulkan bahwa aplikasi “ComanderApp” memiliki waktu yang cukup singkat untuk dipahami pengguna yaitu sekitar 6 – 10 Menit.

2. *Speed of performance*

Kecepatan sebuah fungsi dijalankan dalam aplikasi dapat mempengaruhi kenyamanan user dalam menggunakan aplikasi. Berdasarkan kuesioner yang dilakukan sebelumnya pada pertanyaan No. 3 “ComanderApp” memiliki waktu yang singkat untuk memproses fungsi dari setiap fiturnya yaitu sekitar <30 detik.

3. *Rate of errors by user*

Meminimalisir terjadinya error selama penggunaan aplikasi oleh user juga menjadi point penting untuk kenyamanan user selama menggunakan aplikasi. Berdasarkan kuesioner yang dilakukan sebelumnya pada pertanyaan No. 4 dapat disimpulkan bahwa pengguna dapat dengan mudah menghindari kesalahan dalam menggunakan aplikasi ComanderApp.

4. *Retention over time*

Dengan user interface sederhana dapat mempermudah user dalam mengingat setiap fungsi dan fitur dari aplikasi. Berdasarkan kuesioner yang dilakukan sebelumnya pada pertanyaan No. 5 dapat disimpulkan dari jawaban pengguna yang lebih banyak memilih skala 4 dari skala 1 – 5 bahwa pengguna dapat dengan mudah mengetahui fungsi dari setiap fitur pada aplikasi “ComanderApp”.

5. *Subjective satisfaction*

Kepuasan user dalam berbagai aspek dari interface menjadi nilai penting agar aplikasi tetap digunakan untuk jangka waktu yang Panjang. Berdasarkan kuesioner yang dilakukan sebelumnya pada pertanyaan No. 6 dapat disimpulkan dari jawaban pengguna yang lebih banyak memilih skala 5 dari skala 1 – 5 bahwa pengguna merasa puas dengan aplikasi “ComanderApp”.

#### 4.3.2 Evaluasi Sistem

Pengujian aplikasi “ComanderApp” dilakukan menggunakan metode Black-box testing.

Table 4.1 Testing Aplikasi “ComanderApp”

No	Feature	Actor	Test Date	Input	Expected Result	Result
1	Melakukan Register	Customer	4 Juli 2022	Mengisi data Email, Password, Username, Nama, No Telepon, dan Gender	Sistem berhasil menyimpan data user yang di daftarkan ke database	Pass
		Merchant				
2	Melakukan Login	Customer	4 Juli 2022	Mengisi data Email yang sudah terdaftar dan Password kemudian menekan tombol Login	Sistem mengarahkan user ke halaman utama (homepage) aplikasi	Pass
		Merchant				
		Admin				
3	Melakukan Scan Barcode	Customer	5 Juli 2022	Menekan tombol Scan Barcode pada homepage	Sistem mengarahkan user ke halaman scan barcode	Pass
4	Melihat menu/list product	Customer	5 Juli 2022	Melakukan scan barcode	Sistem menampilkan data merchant dan list produk merchant	Pass
5	Memesan barang/produk	Customer	7 Juli 2022	Menekan card produk dan menekan tombol	Sistem berhasil menambahkan produk kedalam cart	Pass

				+ atau - pada card produk		
6	Melihat detail pemesanan	<i>Customer</i>	7 Juli 2022	Menekan tombol <i>cart</i>	Menampilkan list produk yang dipesan dan detail harga	Pass
7	Melakukan pembayaran	<i>Customer</i>	7 Juli 2022	Menekan tombol <i>checkout</i>	Sistem menampilkan pop up pembayaran sukses dan mengarahkan user ke halaman home	Pass
8	Melihat status transaksi	<i>Customer</i>	7 Juli 2022	Menekan tombol <i>See all Transaction</i> pada homepage	Menampilkan list transaksi beserta status transaksi	Pass
		<i>Merchant</i>				
9	Memproses pesanan	<i>Merchant</i>	7 Juli 2022	Menekan salah satu <i>card</i> transaksi dan mengubah status transaksi	Sistem berhasil mengubah status transaksi dan mengirim notifikasi ke <i>customer</i>	Pass
10	Melihat daftar produk	<i>Merchant</i>	7 Juli 2022	Menekan tombol <i>Manage</i> pada <i>homepage</i>	Sistem menampilkan data produk	Pass
11	Menambahkan produk baru	<i>Merchant</i>	7 Juli 2022	Menekan tombol + pada halaman <i>Manage</i>	Sistem menampilkan form untuk menambahkan produk baru dan menyimpan data	Pass

					produk ke <i>database</i>	
12	Mengubah data produk	<i>Merchant</i>	7 Juli 2022	Menekan tombol <i>edit</i> pada halaman <i>Manage</i>	Sistem menampilkan <i>form edit</i> produk dan mengubah data produk di <i>database</i>	Pass
13	Menghapus produk	<i>Merchant</i>	7 Juli 2022	Menekan tombol <i>delete</i> pada halaman <i>Manage</i>	Sistem berhasil menghapus produk dari <i>database</i> dan menampilkan pesan sukses	Pass
14	Melihat <i>reporting</i> penjualan	<i>Merchant</i>	11 Juli 2022	Menekan tombol <i>summary</i>	Sistem berhasil menampilkan halaman <i>summary</i> yang berisi laporan penjualan	Pass
15	Menampilkan halaman <i>Top-up</i>	<i>Customer</i>	11 Juli 2022	Menekan tombol <i>Top-up</i> pada <i>homepage</i>	Menampilkan halaman <i>top up</i>	Pass
16	Melakukan <i>Top-up</i>	<i>Customer</i>	11 Juli 2022	Memilih nominal angka <i>top-up</i> dan menekan tombol	Sistem mengupdate saldo customer	Pass
17	Melakukan Logout	<i>Customer</i> <i>Merchant</i>	11 Juli 2022	Menekan tombol Logout pada halaman profil	Sistem berhasil mengarahkan user keluar aplikasi yaitu ke halaman login	Pass

#### 4.3.3 Evaluasi User aplikasi “ComanderApp” Berdasarkan Kuesioner

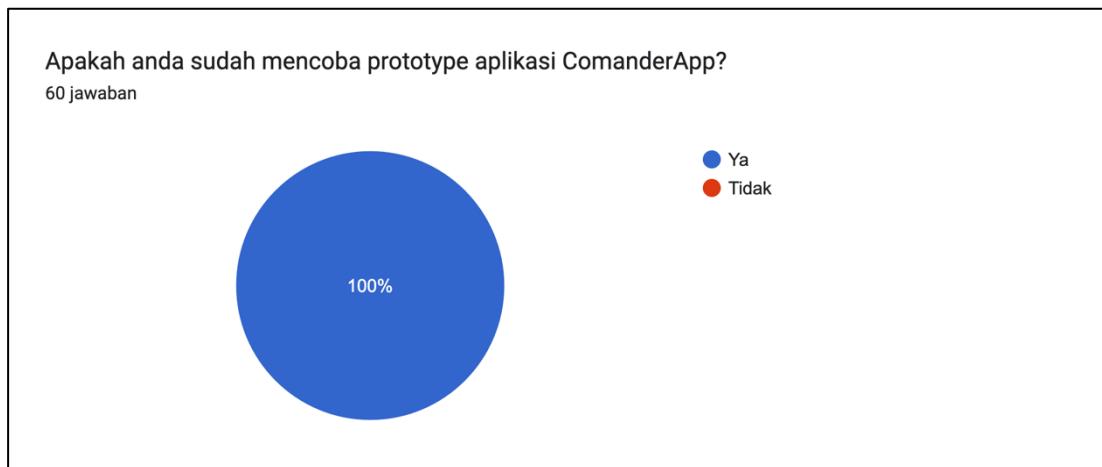
Proses evaluasi dilakukan dengan menerapkan metode kuesioner yang bertujuan untuk mengumpulkan data terkait evaluasi pengguna terhadap aplikasi “ComanderApp”. Berikut pertanyaan dan alas an pertanyaan yang digunakan untuk evaluasi aplikasi “ComanderApp”.

Tabel 4.2 Pertanyaan dan Alasan Pertanyaan Kuesioner

Pertanyaan	Alasan Pertanyaan
Apakah anda sudah mencoba prototype aplikasi “ComanderApp”?	Untuk mengetahui apakah responden sudah mencoba prototype aplikasi “ComanderApp”
Berapa lama waktu yang anda butuhkan untuk memahami fungsi dari setiap fitur yang ada pada aplikasi “ComanderApp”?	Untuk mengetahui berapa lama waktu yang responden butuhkan untuk memahami fungsi dari setiap fitur yang ada
Berapa lama waktu yang dibutuhkan aplikasi “ComanderApp” untuk memproses setiap fiturnya?	Untuk mengetahui berapa lama waktu yang dibutuhkan aplikasi “ComanderApp” untuk memproses setiap fitur
Apakah anda dapat dengan mudah menghindari kesalahan dalam menggunakan aplikasi “ComanderApp”?	Untuk mengetahui apakah responden dapat dengan mudah menghindari kesalahan dalam menggunakan aplikasi “ComanderApp”
Dari skala 1-5, seberapa mudah anda mengetahui fungsi dari setiap fitur?	Untuk mengetahui seberapa mudah responden mengetahui fungsi dari setiap fitur
Dari skala 1-5, seberapa puas anda dengan aplikasi “ComanderApp”?	Untuk mengetahui seberapa puas responden dengan aplikasi “ComanderApp”
Apakah dengan menggunakan aplikasi ini dapat membantu pengguna melakukan transaksi secara mandiri?	Untuk mengetahui apakah aplikasi “ComanderApp” dapat membantu pengguna melakukan transaksi secara mandiri

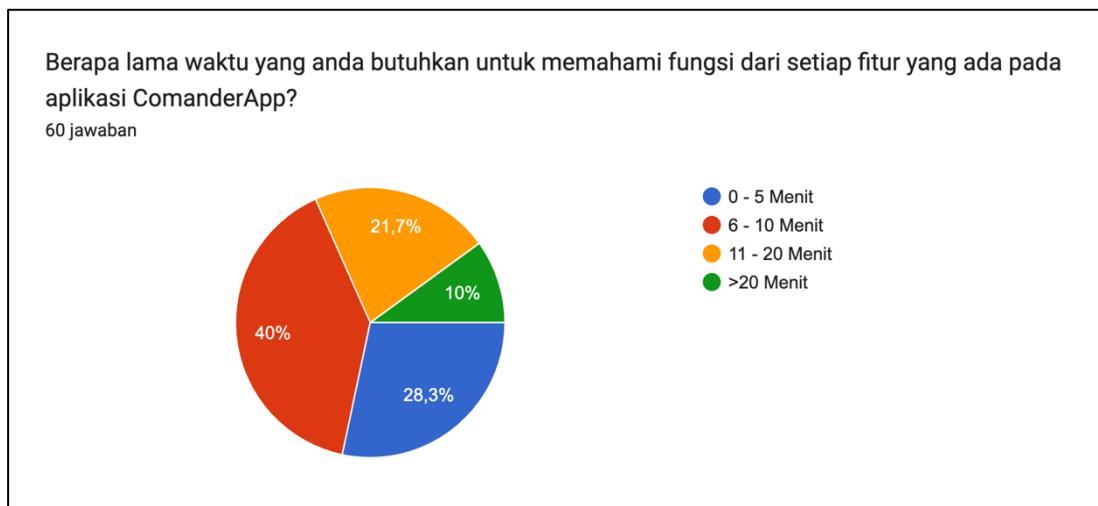
Apakah dengan menggunakan aplikasi ini dapat meminimalisir kerumunan demi mengoptimalkan social dan physical distancing selama pandemi?	Untuk mengetahui apakah aplikasi “ComanderApp” dapat meminimalisir kerumunan demi mengoptimalkan social dan physical distancing selama pandemi
Apakah dengan menggunakan aplikasi ini dapat membantu UMKM dalam menyelesaikan masalah reporting?	Untuk mengetahui apakah dengan menggunakan aplikasi “ComanderApp” dapat membantu UMKM dalam menyelesaikan masalah <i>reporting</i>

Berikut adalah hasil survey dari pertanyaan-pertanyaan diatas dan telah diisi oleh 60 responden:



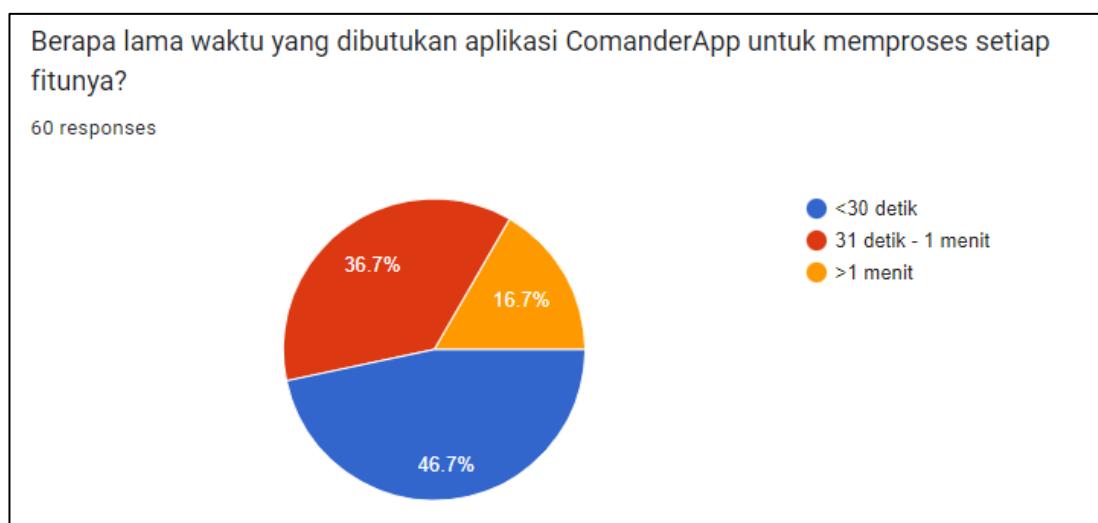
Gambar 4.37 Diagram Pertanyaan 1

Pertanyaan No. 1, “Apakah anda sudah mencoba prototype aplikasi ComanderApp?”. Dari 100% responden sebanyak 100% menjawab “Ya”. Kesimpulan, semua responden sudah mencoba prototype aplikasi ComanderApp.



Gambar 4.38 Diagram Pertanyaan 2

Pertanyaan No. 2, “Berapa lama waktu yang anda butuhkan untuk memahami fungsi dari setiap fitur yang ada pada aplikasi ComanderApp?”. Dari 100% responden sebanyak 40% menjawab “0 – 5 Menit”, sebanyak 28.3% menjawab “6 – 10 Menit” sebanyak 21.7% menjawab “11 – 20” dan sebanyak 10% menjawab “>20 Menit”. Kesimpulan, lebih banyak responden membutuhkan waktu 6 – 10 Menit untuk memahami fungsi dari setiap fitur yang ada pada aplikasi ComanderApp.



Gambar 4.39 Diagram Pertanyaan 3

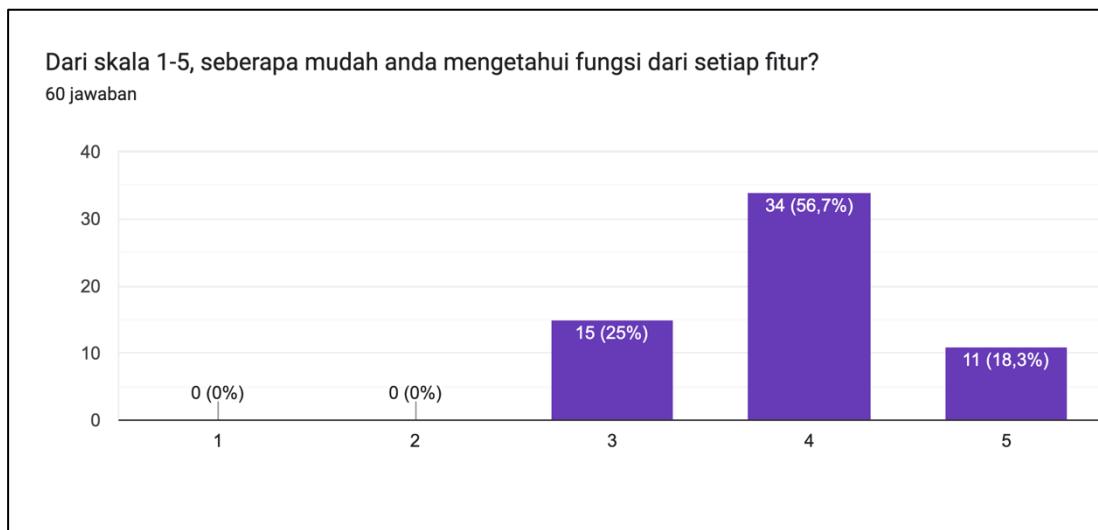
Pertanyaan No. 3, “Berapa lama waktu yang dibutuhkan aplikasi ComanderApp untuk memproses setiap fiturnya?”. Dari 100% responden, sebanyak 46,7% menjawab “< 30 detik”, sebanyak 36,7 % menjawab “31 detik – 1 Menit” dan sebanyak 16,7% menjawab “>1 Menit”. Kesimpulan, lebih

banyak responden menyatakan bahwa aplikasi “ComanderApp membutuhkan waktu kurang dari 30 detik untuk memproses setiap fiturnya.



Gambar 4.40 Diagram Pertanyaan 4

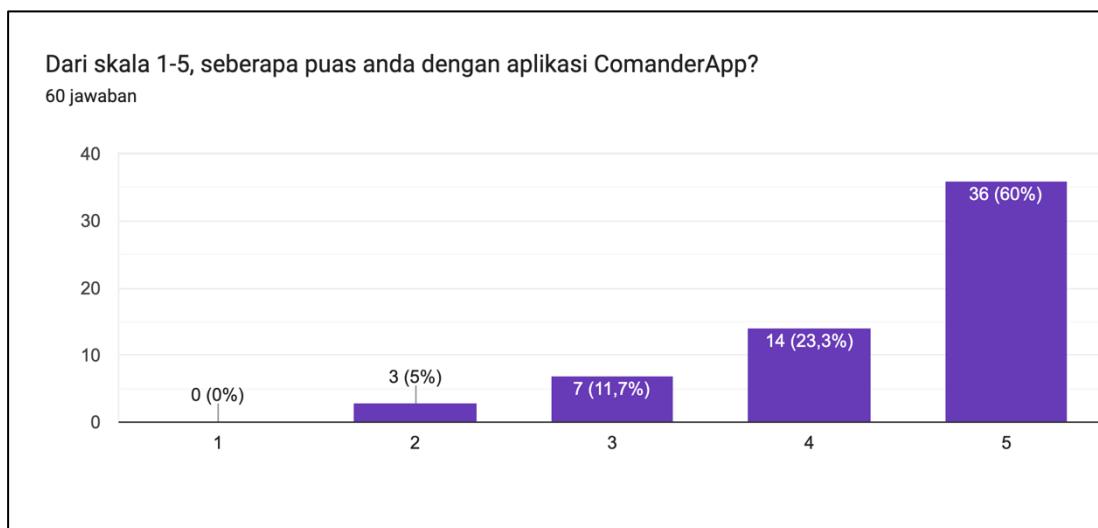
Pertanyaan No. 4, “Apakah anda dapat dengan mudah menghindari kesalahan dalam menggunakan aplikasi ComanderApp?”. Dari 100% responden sebanyak 100% menjawab “Ya”. Kesimpulan, lebih banyak responden menyatakan dapat dengan mudah menghindari kesalahan dalam menggunakan aplikasi ComanderApp.



Gambar 4.41 Diagram Pertanyaan 5

Pertanyaan No. 5, “Dari skala 1-5, seberapa mudah anda mengetahui fungsi dari setiap fitur?”. Dari 100% responden sebanyak 56.7% menjawab

skala 4, sebanyak 25% menjawab skala 3 dan sebanyak 18.3% menjawab skala 5. Kesimpulan, lebih banyak responden memilih skala 4.



Gambar 4.42 Diagram Pertanyaan 6

Pertanyaan No. 6, “Dari skala 1-5, seberapa puas anda dengan aplikasi ComanderApp?”. Dari 100% responden sebanyak 60% menjawab skala 5, sebanyak 23.3% menjawab skala 4, sebanyak 11.7% menjawab skala 3, sebanyak 5% menjawab skala 2. Kesimpulan, lebih banyak responden menjawab skala 5.



Gambar 4.43 Diagram Pertanyaan 7

Pertanyaan No. 7, “Apakah dengan menggunakan aplikasi ini dapat membantu pengguna melakukan transaksi secara mandiri?”. Dari 100% responden, sebanyak 91.7% menjawab “Ya”, sebanyak 8.3% menjawab

“Tidak. Kesimpulan, lebih banyak responden menyatakan dengan menggunakan aplikasi “ComanderApp” dapat membantu pengguna melakukan transaksi secara mandiri.



Gambar 4.44 Diagram Pertanyaan 8

Pertanyaan No. 8, “Apakah dengan menggunakan aplikasi ini dapat meminimalisir kerumunan demi mengoptimalkan social dan physical distancing selama pandemi?”. Dari 100% responden, sebanyak 80% menjawab “Ya” dan sebanyak 20% menjawab “Tidak”. Kesimpulan, lebih banyak responden menyatakan dengan menggunakan aplikasi “ComanderApp” dapat meminimalisir kerumunan demi mengoptimalkan social dan physical distancing selama pandemi.



Gambar 4.45 Diagram Pertanyaan 9

Pertanyaan No.9, “Apakah dengan menggunakan aplikasi ini dapat membantu UMKM dalam menyelesaikan masalah reporting?”. Dari 100% responden, sebanyak 88.3% menjawab “Ya” dan sebanyak 11.7% menjawab “Tidak”. Kesimpulan, lebih banyak responden menyatakan dengan menggunakan aplikasi ComanderApp dapat membantu UMKM dalam menyelesaikan masalah reporting.

#### 4.3.4 Evaluasi Aplikasi sejenis “ComanderApp”

Tabel 4.3 Evaluasi Aplikasi Sejenis

Perbandingan	Bakmi GM	Kopi Kenangan	ComanderApp
Jenis Platform	Web dan android	Android	Android
Menampilkan daftar produk merchant	v	v	v
Dapat melakukan transaksi	v	v	v
Menampilkan transaksi pembelian	v	v	v
Dapat melakukan pembayaran	x	v	v
Dapat melakukan top-up saldo	x	x	v
User Interface	Tampilan menarik dan mudah dimengerti	Tampilan menarik namun kompleks	Tampilan menarik, sederhana dan mudah dipahami

Kelebihan	Bisa melakukan order delivery	Bisa melakukan delivery dan penjadwalan self pickup	Customer dan Merchant dapat berinteraksi langsung melalui transaksi, Merchant dapat melakukan CRUD produk secara mandiri, dapat langsung menampilkan laporan penjualan berdasarkan jangka waktu tertentu
Kekurangan	Aplikasi dikhususkan untuk satu merchant dan tidak dapat digunakan untuk banyak merchant	Aplikasi dikhususkan untuk satu merchant dan tidak dapat digunakan untuk banyak merchant	Tidak memiliki fitur penjadwalan pickup dan delivery

## **BAB 5.**

### **SIMPULAN DAN SARAN**

#### 5.1 Simpulan

Kesimpulan dari pembuatan aplikasi “ComanderApp” adalah sebagai berikut:

1. Aplikasi “ComanderApp” dapat memungkinkan pengguna melakukan transaksi secara mandiri.
2. Aplikasi “ComanderApp” dapat membantu mengurangi kerumunan yang disebabkan oleh antrian.
3. Aplikasi “ComanderApp” dapat membantu UMKM dalam melakukan pembukuan.

#### 5.2 Saran

Beberapa saran yang diberikan untuk aplikasi “ComanderApp” untuk dimasa mendatang:

4. Terkait fitur *top-up* dapat dikembangkan menggunakan payment gateway
5. Menambahkan fitur chat agar *customer* dapat berinteraksi secara langsung dengan *merchant*
6. Menambahkan Halaman notifikasi

## REFERENSI

- Solmin, S. (2015). Aplikasi Rencana Anggaran Proyek pada Kontraktor PT. Hexa Mulia. *d'ComPutarE: Jurnal Ilmiah Information Technology*, 1(1), 44-50.
- Pratama, I. P. A. E. (2019). Sistem Informasi dan Implementasinya: Teori & Konsep Sistem Informasi Disertasi Berbagai Contoh Praktiknya Menggunakan Perangkat Lunak Open Source.
- Mujjati, H. (2013). Analisis dan perancangan sistem informasi stok obat pada apotek arjowinangun. *Speed-Sentra Penelitian Engineering dan Edukasi*, 12(1).
- Inukollu, V. N., Keshamoni, D. D., Kang, T., & Inukollu, M. (2014). Factors influencing quality of mobile apps: Role of mobile app development life cycle. *arXiv preprint arXiv:1410.4537*.
- Brahler, S. (2010). Analysis of the android architecture. *Karlsruhe institute for technology*, 7(8).
- Dennis, A., Wixom, B., & Tegarden, D. (2015). *Systems analysis and design: An object-oriented approach with UML*. John wiley & sons.
- Islam, R., Islam, R., & Mazumder, T. (2010). Mobile application and its global impact. *International Journal of Engineering & Technology (IJEST)*, 10(6), 72-78.
- Sillitto, H., Dori, D., Griego, R. M., Jackson, S., Krob, D., Godfrey, P., ... & McKinney, D. (2017, July). Defining “system”: A comprehensive approach. In *INCOSE International Symposium* (Vol. 27, No. 1, pp. 170-186).
- Hassan, A. M. (2020). JAVA and DART programming languages: Conceptual comparison. *vol*, 17, 845-849.

- Kumar, B. V. D., & Venkata, B. (2012). An Overview of Hypertext Transfer Protocol service Security on Business Domain [A]. In *Proceedings of 2012 International Conference on Management and Education Innovation*.
- Prajapati, P., Patel, N., & Shah, P. (2019). A review of recent detection methods for http ddos attacks. *International Journal of Scientific and Technology Research*, 8(12), 1693-1696.
- Vithani, T., & Kumar, A. (2014, March). Modeling the mobile application development lifecycle. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1, pp. 596-600).
- Moroney, L., Moroney, A., & Anglin. (2017). *Definitive Guide to Firebase* (pp. 51-71). California: Apress.
- Laudon, K. C., & Laudon, J. P. (2004). Managing the digital firm. *Managing Information Systems*, 34-56.
- Jacobson, L., & Booch, J. R. G. (2021). The unified modeling language reference manual.
- Baktha, K. (2017). Mobile Application Development: All the Steps and Guidelines for Successful Creation of Mobile App: Case Study.
- Weske (2007). Information System. *Introduction and Concepts*.
- Developers, A. (2011). What is android?. *Dosegljivo: <http://www.academia.edu/download/30551848/android--tech.pdf>*.
- Arianto, B. (2020). Pengembangan UMKM Digital di Masa Pandemi Covid-19. *ATRABIS: Jurnal Administrasi Bisnis (e-Journal)*, 6(2), 233-247.
- Raharja, S. U. J., & Natari, S. U. (2021). Pengembangan Usaha Umkm Di Masa Pandemi Melalui Optimalisasi Penggunaan Dan Pengelolaan Media Digital. *Kumawula: Jurnal Pengabdian Kepada Masyarakat*, 4(1), 108-123.

Teviningrum, S., & Urfa, F. (2021). Analisis Pengaruh Self Service Technology Terhadap Kualitas Pelayanan Restoran, Studi kasus di McDonald's TB Simatupang, Jakarta. *Jurnal Simki Economic*, 4(1), 11-22.

Wijaya, A. (2020). Implementasi Konsep Self Service Technology (SST) Pada Model Bisnis Restoran.

Tazkiyyaturrohmah, R. (2018). Eksistensi Uang Elektronik Sebagai Alat Transaksi Keuangan Modern. *Muslim Heritage*, 3(1), 23-44.

Fitri, R., Kom, S., & Kom, M. (2020). *Pemrograman Basis Data Menggunakan MySQL*. Deepublish.

Irawan, Y., Muzid, S., Susanti, N., & Setiawan, R. (2018). System Testing using Black Box Testing Equivalence Partitioning (Case Study at Garbage Bank Management Information System on Karya Sentosa). In *International Conference on Computer Science and Engineering Technology*.

Hayat, F., Rehman, A. U., Arif, K. S., Wahab, K., & Abbas, M. (2019, July). The influence of agile methodology (Scrum) on software project management. In *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (pp. 145-149). IEEE.

Shneiderman, B., Plaisant, C., Cohen, M. S., Jacobs, S., Elmquist, N., & Diakopoulos, N. (2016). *Designing the user interface: strategies for effective human-computer interaction*. Pearson.

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies*, 14(11).

Figma. (2022). Enterprise Overview. Retrieved from Figma:  
<https://www.figma.com/organization/>

Github. (2022). about. Retrieved from Github: <https://github.com>

Trello. (2022) tour . Retrived from Trello : <https://trello.com>

Flutter. (2022). Retrieved from Flutter : <https://flutter.dev>



## LAMPIRAN

### Lampiran 1 Kuisioner analisis aplikasi

# Kuesioner Analisis Aplikasi Self Service Customer- ComanderApp

Assalamualaikum warahmatullahi wabarakatuh,  
salam sejahtera untuk kita semua, Om  
Swastyastu, Namo Buddhaya, salam kebajikan.  
Perkenalkan kami mahasiswa Universitas Bina  
Nusantara Jakarta semester 8 jurusan  
Computer Science yang sedang menuntaskan  
tugas akhir kami dengan topik Perancangan dan  
Pengembangan sebuah Aplikasi Self Service  
berbasis Mobile Android. Kami ingin meminta  
sedikit waktu untuk mengisi kuesioner berikut  
ini.

Kuesioner yang telah dirangkai ini bertujuan  
untuk melakukan analisis aplikasi yang  
dikembangkan, dalam segi fungsionalitas dan  
kemudahan dalam penggunaan aplikasi.

Besar harapan kami bahwa teman-teman dapat  
bersedia untuk meluangkan waktu untuk  
mengisi kuesioner ini, data yang akan diterima  
akan sangat membantu kami untuk keperluan  
penuntasan tugas akhir.



Terima Kasih,

13.36 ↗  
◀ Telegram

Email \*

Email Anda

Nama \*

Jawaban Anda

Apakah anda pernah mengalami antrian yang panjang? \*

Ya

Tidak

Bagaimana situasi antrian yang pernah anda \* alami?

Ramai

Cukup ramai

Biasa saja

Cukup sepi

Sepi

docs.google.com — Private

13.36 ↗  
◀ Telegram

Apakah terdapat perbedaan antara antrian \* saat pandemic dengan normal?

Ya  
 Tidak

Apakah antrian lebih lama saat masa \* pandemi?

Ya  
 Tidak

Apakah anda pernah menggunakan layanan \* self service seperti di McDonald, bakmi GM dan kopi kenangan?

Ya  
 Tidak

Apakah layanan tersebut membantu anda \* mengurangi antrian?

Sangat mengurangi  
 Menurunkan

docs.google.com — Private

13.36 ↗  
◀ Telegram

Netral  
 Tidak mengurangi  
 Sangat tidak mengurangi

Bagaimana tingkat pelayanan yang diberikan? \*

Sangat memuaskan  
 Memuaskan  
 Biasa saja  
 Tidak memuaskan  
 Sangat tidak memuaskan

Apakah layanan tersebut dapat berguna apabila diterapkan di setiap rumah makan atau toko-toko lainnya? \*

Sangat berguna  
 Berguna  
 Netral  
 Tidak berguna  
 Sangat tidak berguna

docs.google.com — Private

13.37 ↗  
◀ Telegram

Apakah layanan tersebut dapat meningkatkan social-distancing selama masa ppkm? \*

Sangat meningkatkan  
 Meningkatkan  
 Netral  
 Tidak meningkatkan  
 Sangat tidak meningkatkan

Apakah layanan tersebut tetap efektif diterapkan apabila keadaan sudah kembali normal? \*

Sangat efektif  
 Efektif  
 Netral  
 Tidak efektif  
 Sangat tidak efektif

Bagaimana pandangan anda apabila terdapat suatu aplikasi yang memudahkan kalian dalam memesan hanya dengan

docs.google.com — Private

13.37 ↗  
◀ Telegram

Bagaimana pandangan anda apabila \*  
terdapat suatu aplikasi yang memudahkan  
kalian dalam memesan hanya dengan  
melakukan scan barcode dan melakukan  
transaksi dalam aplikasi tersebut sehingga  
kalian tidak perlu berdiri di depan kasir?

Sangat membantu  
 Membantu  
 Netral  
 Tidak membantu  
 Sangat tidak membantu

Apakah anda akan menggunakan aplikasi \*  
yang menerapkan sistem (self service)  
tersebut?

Ya  
 Tidak

**Kirim** **Kosongkan formulir**

Jangan pernah mengirimkan sandi melalui Google Formulir.

! Konten ini tidak dibuat atau didukung oleh Google. [Laporkan Penyalahgunaan](#) - [Persyaratan Layanan](#) - [Kebijakan Privasi](#)

docs.google.com — Private

## Lampiran 2 Kuisioner Evaluasi Aplikasi

The screenshot shows a Google Document with the following content:

**Kuesioner Evaluasi  
Aplikasi Self-service  
ComanderApp**

Assalamualaikum warahmatullahi wabarakatuh,  
salam sejahtera untuk kita semua, Om  
Swastyastu, Namo Buddhaya, salam kebajikan.  
Perkenalkan kami mahasiswa Universitas Bina  
Nusantara Jakarta semester 8 jurusan  
Computer Science yang sedang menuntaskan  
tugas akhir kami dengan topik Perancangan dan  
Pengembangan sebuah Aplikasi Self Service  
berbasis Mobile Android. Kami ingin meminta  
sedikit waktu untuk mengisi kuisioner berikut  
ini.

Kuesioner yang telah dirangkai ini bertujuan  
untuk melakukan evaluasi aplikasi yang  
dikembangkan, dalam segi fungsionalitas dan  
kemudahan dalam penggunaan aplikasi.

Berikut merupakan link untuk mengunduh  
aplikasi:  
[https://drive.google.com/drive/folders/1snZSn\\_hVyBVdasbLA4PiTB51h2p1HPdl?usp=sharing](https://drive.google.com/drive/folders/1snZSn_hVyBVdasbLA4PiTB51h2p1HPdl?usp=sharing)

Besar harapan kami bahwa teman-teman dapat  
bersedia untuk meluangkan waktu selama 5-10  
menit untuk mengisi kuisioner ini, data yang  
akan diterima akan sangat membantu kami  
untuk keperluan penuntasan tugas akhir.

Terima Kasih,

Annisa Octaviana Nurshanty ([2201836936](#)),  
Fahrul Rozi Arsyandra ([2201835883](#))

docs.google.com — Private

13.42 ↗

Email \*

Email Anda

Nama \*

Jawaban Anda

Apakah anda sudah mencoba prototype aplikasi ComanderApp? \*

Ya

Tidak

Berapa lama waktu yang anda butuhkan untuk memahami fungsi dari setiap fitur yang ada pada aplikasi ComanderApp? \*

0 - 5 Menit

6 - 10 Menit

11 - 20 Menit

>20 Menit

docs.google.com — Private

13.42 ↗



Berapa lama waktu yang dibutukan aplikasi \* ComanderApp untuk memproses setiap fitunya?

- <30 detik
- 31 detik - 1 menit
- >1 menit

Apakah anda dapat dengan mudah menghindari kesalahan dalam menggunakan aplikasi ComanderApp?

- Ya
- Tidak

Dari skala 1-5, seberapa mudah anda mengetahui fungsi dari setiap fitur?

1      2      3      4      5

- 
- 
- 
- 
- 

▢ Dari skala 1-5, seberapa puas anda dengan aplikasi ComanderApp?

13.42 ↗



Dari skala 1-5, seberapa puas anda dengan aplikasi ComanderApp?

1      2      3      4      5

Apakah dengan menggunakan aplikasi ini dapat membantu pengguna melakukan transaksi secara mandiri?

- Ya  
 Tidak

Apakah dengan menggunakan aplikasi ini dapat meminimalisir kerumunan demi mengoptimalkan social dan physical distancing dimasa pandemi?

- Ya  
 Tidak

Apakah dengan menggunakan aplikasi ini dapat membantu UMKM dalam

13.42 ↗

● Ya  
● Tidak

Apakah dengan menggunakan aplikasi ini dapat meminimalisir kerumunan demi mengoptimalkan social dan physical distancing dimasa pandemi?

● Ya  
● Tidak

Apakah dengan menggunakan aplikasi ini dapat membantu UMKM dalam menyelesaikan masalah reporting?

● Ya  
● Tidak

**Kirim** **Kosongkan formulir**

Jangan pernah mengirimkan sandi melalui Google Formulir.

Konten ini tidak dibuat atau didukung oleh Google. [Laporkan Penyalahgunaan](#) - [Persyaratan Layanan](#) - [Kebijakan Privasi](#)

**Google Formulir**

docs.google.com — Private

## Lampiran 2 Surat Survei

**Surat Survei**  
No : 400/HoP.CS/VII/2022

Yang bertandatangan di bawah ini :

Nama : **Irene Anindaputri Iswanto, S.Kom., M.Sc.Eng.**  
Jabatan : Head of Computer Science Program

Menyatakan bahwa mahasiswa berikut ini :

No.	N.I.M	Nama Mahasiswa
1.	2201835883	FAHRUL ROZI ARSYANDRA
2.	2201836936	ANNISA OCTAVIANA NURSHANTY
3.		

Sudah melakukan survei di Jurusan Computer Science Program Universitas Bina Nusantara dengan judul skripsi:

**“PENGEMBANGAN APLIKASI SELF-SERVICE BERBASIS MOBILE APPLICATION”**

di Lingkungan Universitas Bina Nusantara.

Demikian surat pernyataan ini dibuat untuk digunakan sebagaimana mestinya.

Jakarta, 07 Juli 2022



*Irene*  
(IRENE ANINDAPUTRI)

**Irene Anindaputri Iswanto, S.Kom., M.Sc.Eng.**  
Head of Computer Science Program

### Lampiran 3 Surat Pernyataan Telah Melakukan Survei



Specialist Offset Printing, Komp. Ruko Pasar Cipta Puri Blok DD No.22 Tiban  
Telp. (0778) 324388 HP. 0812 7001 1925,  
Email: [mediainsanimandiri@gmail.com](mailto:mediainsanimandiri@gmail.com)

No :  
Lampiran : -  
Perihal : Surat Peryataan Telah Melakukan Survei

Yang bertanda tangan di bawah ini:

Nama : Yoga Zanades

Perusahaan : Media Insani Mandiri

Dengan ini menyatakan bahwa:

No	Nim	Nama
1	2201835883	Fahrul Rozi Arsyandra
2	2201836936	Annisa Octaviana Nurshanty

Benar telah melakukan Survei Skripsi berupa kegiatan wawancara untuk mengumpulkan data, sejak tanggal 12 Juli 2022 di Media Insani Mandiri dengan judul "Pengembangan Aplikasi Self-Service Berbasis Mobile Application" sebagai syarat untuk kelulusan Skripsi pada Program Computer Science di Universitas Bina Nusantara.

Demikianlah surat pemberitahuan ini agar dapat dipergunakan sebaik-baiknya dan sebagaimana mestinya, atas perhatian dan kerjasamanya kami ucapan terima kasih.

Jakarta, 13 Juli 2022



Yoga Zanades



**NOISY TIME CLOTHING**

Jalan Kota Bambu Raya, RT.1/RW.5, Kota Bambu Utara, Kec. Palmerah,  
Kota Jakarta Barat, Jakarta 11420  
Telp : 082110047214

No : 06.01/NT/VII/2022

Lampiran : -

Perihal : Surat Peryataan Telah Melakukan Survei

Yang bertanda tangan di bawah ini:

Nama : Zikri Aufa

Perusahaan : Noisy Time Clothing

Dengan ini menyatakan bahwa:

No	Nim	Nama
1	<u>2201835883</u>	Fahrul Rozi Arsyandra
2	<u>2201836936</u>	Annisa Octaviana Nurshanty

Benar telah melakukan Survei Skripsi berupa kegiatan wawancara untuk mengumpulkan data, sejak tanggal 12 Juli 2022 di UMKM Noisy Time Clothing dengan judul "Pengembangan Aplikasi Self-Service Berbasis Mobile Application" sebagai syarat untuk kelulusan Skripsi pada Program Computer Science di Universitas Bina Nusantara.

Demikianlah surat pemberitahuan ini agar dapat dipergunakan sebaik-baiknya dan sebagaimana mestinya, atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Jakarta, 13 Juli 2022



Owner

Zikri Aufa

## RIWAYAT HIDUP

### Riwayat Hidup/Curriculum Annisa Octaviana Nurshanty



#### **PERSONAL INFORMATION**

**Binusian ID** 2201836936  
**Full Name** Annisa Octaviana Nurshanty  
**E-Mail** annisa.nurshanty@binus.ac.id

<b>Address</b>	<b>Permanent</b>
	Jalan Wuluh IV No. 2 RT 07/RW 07, Kota Bambu Utara, Palmerah, Jakarta Barat 11420, Daerah Khusus Ibukota Jakarta

**Phone Numbers** Mobile: 087782555016

<b>Gender</b>	Female
<b>Birth Place / Date</b>	Mataram/ 13 Oktober 2000
<b>Nationality</b>	Indonesia
<b>Marital Status</b>	Single
<b>Religion</b>	Islam

#### **FORMAL EDUCATION**

<b>Sept 2018– Present</b>	Bina Nusantara University, Jakarta, Indonesia Bachelor S1, Computer Science GPA: 3,28
<b>Jul 2015 – Apr 2018</b>	SMAN 78 Jakarta Barat, Jakarta, Indonesia Senior High, Not Applicable GPA: N/A

#### **ORGANIZATION EXPERIENCE**

<b>Jul 2018 – Aug 2020</b>	Himpunan Teknik Informatika Binus University (HIMTI)
----------------------------	--

#### **WORKING EXPERIENCE**

**Sept 2021 – Feb 2022**

Web Developer di PT.Telkom  
Indonesia (Telkom), Jakarta

**Job Description:**

Participate in the entire application lifecycle, focusing on coding and debugging, Build reusable code and libraries for future use, Follow emerging technologies, Doing test driven development, Within a cross-functional team, collaborate with other developers specializing in backend, frontend, quality assurance, product owner, scrum master, and etc, Apply design patterns and design principles to produce maintainable code, Learn multiple tech stacks to use the best tools for the job, Solve technical problems, Optimize applications for maximum speed

## Riwayat Hidup/Curriculum Vitae Fahrul Rozi Arsyandra



### PERSONAL INFORMATION

**Binusian ID** 2201835883  
**Full Name** Fahrul Rozi Arsyandra  
**E-Mail** fahrul.arsyandra@binus.ac.id

**Address** **Permanent**  
Tiban Koperasi Blok E No 47, Kel. Tiban Baru, Kec. Sekupang, Kota Batam, Kepulauan Riau, 29425

**Phone Numbers** Mobile: 081270617007

**Gender** Male  
**Birth Place / Date** Batam/ 27 April 2000  
**Nationality** Indonesia  
**Marital Status** Single  
**Religion** Islam

### FORMAL EDUCATION

<b>Sep 2018 – Present</b>	Bina Nusantara University, Jakarta, Indonesia Bachelor S1, Computer Science GPA : 3.18
<b>Jul 2015 – Apr 2018</b>	SMAN 1 Batam Senior High, Not Applicable GPA : N/A

### ORGANIZATION EXPERIENCE

#### WORKING EXPERIENCE

<b>Feb 2021 – Feb 2022</b>	Internship at BCA Multifinance as Mobile Programmer
	<b>Job Description:</b> Develop mobile application using flutter