

**PEMBUATAN APLIKASI PENJUALAN BINATANG TERNAK  
BERBASIS ANDROID DI PETERNAKAN BEBEK ALFALAH  
DESA PADAULUN**

**SKRIPSI**

Karya Tulis sebagai Syarat memperoleh  
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi  
Universitas Bale Bandung

Disusun oleh :

TINA LESTARI  
NPM. 301180013



**PROGRAM STRATA 1  
PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG  
BANDUNG  
2022**

**LEMBAR PERSETUJUAN PEMBIMBING**  
**PEMBUATAN APLIKASI PENJUALAN BINATANG TERNAK**  
**BERBASIS ANDROID DI PETERNAKAN BEBEK ALFALAH**  
**DESA PADAULUN**

Disusun oleh :

TINA LESTARI  
NPM. 301180013

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar  
**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2022

Disetujui oleh :

Pembimbing Utama

Pembimbing Pendamping

Yusuf Muharam, S.Kom., M.Kom.  
NIK. 04104820003

Rosmalina, S.T., M.Kom.  
NIK. 04104808122

**LEMBAR PERSETUJUAN PENGUJI**

**PEMBUATAN APLIKASI PENJUALAN BINATANG TERNAK  
BERBASIS ANDROID DI PETERNAKAN BEBEK ALFALAH  
DESA PADAULUN**

Disusun oleh :

TINA LESTARI

NPM. 301180013

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar  
**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2022

Disetujui oleh :

Penguji 1

Penguji 2

Yudi Herdiana, S.T., M.T.

NIK. 04104808008

Yaya Suharya, S.Kom., M.T.

NIK. 01043170007

**LEMBAR PENGESAHAN PROGRAM STUDI**

**PEMBUATAN APLIKASI PENJUALAN BINATANG TERNAK  
BERBASIS ANDROID DI PETERNAKAN BEBEK ALFALAH  
DESA PADAULUN**

Disusun oleh :

TINA LESTARI  
NPM. 301180013

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar  
**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2022

Disahkan oleh :

Mengetahui,  
Dekan

Mengesahkan,  
Ketua Program Studi

Yudi Herdiana, S.T., M.T.  
NIK. 04104808008

Yusuf Muharam, S.Kom., M.Kom.  
NIK. 04104820003

## HALAMAN PERNYATAAN

Penyusun yang bertanda tangan di bawah ini:

Nama : Tina Lestari

NPM : 301180013

Judul Skripsi : Pembuatan Aplikasi Penjualan Binatang Ternak Berbasis Android  
Di Peternakan Bebek Alfalah Desa Padaulun

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran, dan pemaparan asli dari penyusun sendiri, baik untuk naskah laporan maupun kegiatan *programming* yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, penyusun mencantumkan sumber yang jelas.

Pernyataan ini penyusun buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka penyusun bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di Fakultas Teknologi Informasi Universitas Bale Bandung. Demikian surat pernyataan ini penyusun buat dalam keadaan sadar tanpa paksaan dari pihak mana pun.

Baleendah, Agustus 2022

Yang membuat pernyataan

**TINA LESTARI**

NPM. 301180013

## ABSTRAK

*Sejalan dengan cepatnya perkembangan teknologi ini, para pelaku usaha semakin dipacu untuk menggunakan teknologi sebagai senjata untuk tetap bertahan dan memenangkan persaingan yang semakin hari semakin ketat dan keras. Salah satu bentuk sarana perdagangan di internet yang biasa kita temui disebut dengan e-commerce. Tidak seperti bisnis peternakan Bebek Alfalah yang masih belum efektif dan efisien dikarenakan dalam proses jual beli masih dilakukan secara langsung di tempat karena jumlah transaksi penjualan masih relatif rendah karena banyak usaha yang masih melakukan promosi dan penjualan masih menggunakan brosur ataupun baliho, maka penulis memutuskan untuk merancang sebuah aplikasi e-commerce untuk usaha pemasaran peternakan tersebut yang nantinya dapat memudahkan peternakan Bebek Alfalah dalam melakukan penjualan dan promosi ternak bebek dan daging yang dapat di harapkan memberikan calon pembeli memudahkan untuk mendapatkan informasi tentang peternakan dan daging bebek tersebut.*

*Proses rancang bangun aplikasi ini menggunakan metode System Development Life Cycle SDLC model waterfall dengan tahapan analisis, perancangan, implementasi dan pengujian. Pada tahap analisis dilakukan pengumpulan data dengan observasi ke tempat penjualan binatang ternak, kemudian wawancara kepada Mang Asep selaku pemilik penjualan bebek dan studi pustaka yang sesuai dengan penelitian. Perancangannya menggunakan software Microsoft Visio sebagai tool untuk perancangan sistem Unified Modeling Language UML dengan jenis perancangan use case diagram, activity diagram dan class diagram. Pada tahap desain tampilan menggunakan software Balsamiq Mockup. Kemudian perancangan diimplementasikan dengan menggunakan Laravel untuk membangun aplikasi untuk dikelola oleh admin dan menggunakan flutter untuk membangun aplikasi untuk digunakan oleh customer dalam memesan produk. Pengujian aplikasi ini hnaya diuji dari fungsional sistem input dan output sesuai dengan yang diharapkan.*

*Aplikasi penjualan binatang ternak yang dibuat dengan menggunakan Laravel dan flutter telah siap dijalankan sebagaimana fungsinya. Dengan adanya aplikasi ini diharapkan dapat memudahkan dalam menjalankan bisnis penjualan Bebek Alfalah. Tentunya nanti aplikasi bisa dikembangkan lagi menjadi lebih baik sebagaimana kebutuhan dalam proses penjualan menggunakan aplikasi.*

**Kata Kunci :** *Android, Aplikasi, Flutter, Laravel, Penjualan*

## **ABSTRACT**

*In line with the rapid development of this technology, business actors are increasingly encouraged to use technology as a weapon to stay afloat and win the competition that is getting tougher and harder every day. One form of means of trading on the internet that we usually encounter is called e-commerce. Unlike the duck farming business Bebek Alfalah duck is still not effective and efficient because the k an in the buying and selling process is still admitted directly on the spot because the number of sales transactions is still relatively low because many businesses that are still doing promotions and sales still use brochures or billboards, so the author decided to design an e-commerce application for the marketing efforts of the farm which can later facilitate the Bebek Alfalah farm in selling and promoting duck and meat livestock which can be expected to provide potential buyers with easy information about the farm and duck meat.*

*The design and build process of this application uses the System Development Life Cycle SDLC waterfall model method with stages of analysis, design, implementation and testing. At the analysis stage, data collection was carried out by observation to the place of sale of livestock, then an interview with Bebek Alfalah as the owner of duck sales and a literature study in accordance with the research. The design uses Microsoft Visio software as a tool for designing a Unified Modeling Language UML system with the type of designing use case diagrams, activity diagrams and class diagrams. At the design stage of the display using Balsamiq Mockup software. Then the design is implemented by using Laravel to build applications to be managed by admins and using flutter to build applications for use by customers in ordering products. Testing this application is hnaya tested from functional input and output systems as expected.*

*Farm animal sales applications created using Laravel and flutter are ready to run as they function. With this application, it is hoped that it can make it easier to run the Bebek Alfalah sales business. Of course, later the application can be developed again to be better as needed in the sales process using the application.*

*Keywords : Android, Apps, Flutter, Laravel, Sales*

## KATA PENGANTAR

Puji syukur penulis panjatkan terhadap kehadiran Allah SWT atas limpahan Rahmat, hidayah dan Karunia-Nya, sholawat serta salam semoga tetap tercurahkan kepada Rasulullah Muhammad Shalallahu ‘alaihi wa salam, beserta keluarga dan sahabatnya. Sehingga penulis dapat menulis dan meneliti proposal skripsi ini dengan judul : “Pembuatan Aplikasi Penjualan Binatang Ternak Berbasis Android Di Peternakan Mang Asep Desa Padaulun ”.

Oleh karena itu Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Kedua Orang Tua yang selalu memberikan dukungan baik dalam bentuk moral maupun materi
2. Bapak Yudi Herdiana, S.T., M.T. selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
3. Bapak Yusuf Muharam, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung sekaligus dosen pembimbing utama.
4. Ibu Rosmalina, S.T., M.Kom. selaku dosen pembimbing pendamping di Fakultas Teknologi Informasi Universitas Bale Bandung.
5. Mang Asep Ahsan selaku pemilik peternakan Bebek Alfalah
6. Seluruh pegawai yang ada di peternakan Mang Asep di Desa Padaulun.
7. Rekan–rekan mahasiswa angkatan 2018 Fakultas Teknologi Informasi yang telah memberikan semangat untuk menyusun laporan skripsi ini.
8. Semua pihak yang telah membantu dan memberikan dukungan kepada penyusun untuk menyelesaikan laporan tugas skripsi ini.

Penulis menyadari bahwa masih banyak kekurangan dalam laporan ini, segala kritik dan saran yang membangun akan penyusun terima dengan baik. Akhir kata semoga laporan skripsi ini bisa diterima dan bermanfaat bagi kita semua.

Bandung, Agustus 2022

Penulis



## DAFTAR ISI

<b>ABSTRAK .....</b>	<b>vi</b>
<b><i>ABSTRACT</i> .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiv</b>
<b>DAFTAR TABEL .....</b>	<b>xvi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Metodologi Penelitian.....	2
1.5.1 Lokasi Penelitian .....	3
1.5.2 Metode Pengumpulan Data .....	3
1.5.3 Metode Pengembangan Sistem.....	3
1.6 Sistematika Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1 Landasan Teori .....	5
2.2 Dasar Teori .....	8
2.2.1 Pengertian Aplikasi .....	8
2.2.2 Pengertian Penjualan .....	8
2.2.3 Pengertian Binatang Ternak .....	9
2.2.4 Sejarah Android.....	10
2.2.5 Pengertian Andorid.....	10
2.2.6 Perkembangan Android .....	10
2.2.7 Fitur Pada Android .....	13
2.2.8 <i>Waterfall</i> .....	14
2.2.9 Flutter .....	15
2.2.10 PHP.....	16
2.2.11 Laravel.....	16

2.2.12 Tailwind.....	17
2.2.13 MySQL .....	17
2.2.14 Balsamiq Mockup.....	18
2.2.15 UML ( <i>Unified Modelling Language</i> ) .....	18
2.2.16 Black Box Testing .....	22
<b>BAB III METODOLOGI .....</b>	<b>23</b>
3.1 Kerangka Pikir .....	23
3.2 Deskripsi .....	23
3.2.1 Pengumpulan Data.....	24
3.2.2 Analisis .....	24
3.2.3 Perancangan.....	25
3.2.4 Implementasi .....	25
3.2.5 Pengujian dan Evaluasi.....	25
<b>BAB IV ANALISIS DAN PERANCANGAN .....</b>	<b>26</b>
4.1 Analisis .....	26
4.1.1 Analisis Masalah .....	26
4.1.2 Analisis <i>Software</i> .....	26
4.1.3 Analisis Pengguna .....	27
4.1.4 <i>User Interface</i> .....	27
4.1.5 Fitur-Fitur .....	28
4.1.6 Analisis Data .....	29
4.1.7 Analisis Biaya.....	29
4.2 Perancangan.....	30
4.2.1 Pemodelan UML .....	30
4.2.2 Struktur Tabel .....	41
4.2.3 Desain .....	43
<b>BAB V IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>52</b>
5.1 Implementasi.....	52
5.1.1 <i>Listing</i> Program .....	52
5.1.2 Implementasi Sistem .....	69
5.1.3 Spesifikasi Sistem.....	69
5.2 Pengujian .....	78

<b>BAB VI KESIMPULAN DAN SARAN.....</b>	<b>80</b>
6.1 Kesimpulan .....	80
6.2 Saran .....	80
<b>DAFTAR PUSTAKA .....</b>	<b>82</b>
<b>LAMPIRAN.....</b>	<b>84</b>

## DAFTAR GAMBAR

Gambar 2.1 <i>Waterfall</i> .....	14
Gambar 3.1 Kerangka Pikir.....	23
Gambar 4.1 <i>Use Case</i> Admin .....	30
Gambar 4.2 <i>Use Case</i> Pelanggan .....	31
Gambar 4.3 <i>Activity Diagram</i> Admin - <i>Login</i> .....	32
Gambar 4.4 <i>Activity Diagram</i> Admin - Produk.....	33
Gambar 4.5 <i>Activity Diagram</i> Admin – Lupa <i>Password</i> .....	34
Gambar 4.6 <i>Activity Diagram</i> Admin – <i>User</i> .....	34
Gambar 4.7 <i>Activity Diagram</i> Admin - Transaksi.....	35
Gambar 4.8 <i>Activity Diagram</i> Admin – Galeri Produk .....	36
Gambar 4.9 <i>Activity Diagram</i> Pelanggan – Profil/Akun .....	37
Gambar 4.10 <i>Activity Diagram</i> Pelanggan – Edit Profil .....	38
Gambar 4.11 <i>Activity Diagram</i> Pelanggan – Home .....	38
Gambar 4.12 <i>Activity Diagram</i> Pelanggan – Keranjang .....	39
Gambar 4.13 <i>Activity Diagram</i> Pelanggan – <i>Login</i> .....	40
Gambar 4.14 <i>Activity Diagram</i> Pelanggan – Registrasi .....	40
Gambar 4.15 <i>Class Diagram</i> .....	41
Gambar 4.16 Desain Tampilan Antarmuka Admin - <i>Login</i> .....	44
Gambar 4.17 Desain Tampilan Antarmuka Admin – Lupa <i>Password</i> .....	44
Gambar 4.18 Desain Tampilan Antarmuka Admin - <i>Dashboard</i> .....	45
Gambar 4.19 Desain Tampilan Antarmuka Admin - Produk .....	45
Gambar 4.20 Desain Tampilan Antarmuka Admin – Tambah Produk.....	45
Gambar 4.21 Desain Tampilan Antarmuka Admin – Edit Produk .....	46
Gambar 4.22 Desain Tampilan Antarmuka Admin – Galeri Produk.....	46
Gambar 4.23 Desain Tampilan Antarmuka Admin - Transaksi .....	46
Gambar 4.24 Desain Tampilan Antarmuka Admin – Edit Transaksi .....	47
Gambar 4.25 Desain Tampilan Antarmuka Admin – Detail Transaksi .....	47
Gambar 4.26 Desain Tampilan Antarmuka Admin - <i>Users</i> .....	47
Gambar 4.27 Desain Tampilan Antarmuka Admin – Edit <i>Users</i> .....	48
Gambar 4.28 Desain Tampilan Antarmuka Pelanggan - <i>Login</i> .....	48

Gambar 4.29	Desain Tampilan Antarmuka Pelanggan - Registrasi .....	48
Gambar 4.30	Desain Tampilan Antarmuka Pelanggan - Home.....	49
Gambar 4.31	Desain Tampilan Antarmuka Pelanggan – Profil/Akun.....	49
Gambar 4.32	Desain Tampilan Antarmuka Pelanggan – Edit Profil .....	49
Gambar 4.33	Desain Tampilan Antarmuka Pelanggan – Riwayat Pesanan .....	50
Gambar 4.34	Desain Tampilan Antarmuka Pelanggan – Keranjang Kosong.....	50
Gambar 4.35	Desain Tampilan Antarmuka Pelanggan – Keranjang Isi .....	50
Gambar 4.36	Desain Tampilan Antarmuka Pelanggan – Detail Produk .....	51
Gambar 5.1	Halaman Admin - <i>Login</i> .....	70
Gambar 5.2	Halaman Admin – Lupa <i>Password</i> .....	70
Gambar 5.3	Halaman Admin - <i>Dashboard</i> .....	71
Gambar 5.4	Halaman Admin - Produk .....	71
Gambar 5.5	Halaman Admin – Tambah Produk.....	71
Gambar 5.6	Halaman Admin – Edit Produk .....	72
Gambar 5.7	Halaman Admin – Detail Produk .....	72
Gambar 5.8	Halaman Admin – <i>Users</i> .....	72
Gambar 5.9	Halaman Admin – Edit <i>User</i> .....	73
Gambar 5.10	Halaman Admin – Transaksi.....	73
Gambar 5.11	Halaman Admin – Transaksi Edit .....	73
Gambar 5.12	Halaman Admin – Detail Transaksi .....	74
Gambar 5.13	Halaman Pelanggan – <i>Login</i> .....	74
Gambar 5.14	Halaman Pelanggan – Registrasi.....	74
Gambar 5.15	Halaman Pelanggan – Home .....	75
Gambar 5.16	Halaman Pelanggan – Detail Produk .....	75
Gambar 5.17	Halaman Pelanggan – Keranjang Kosong.....	76
Gambar 5.18	Halaman Pelanggan – Keranjang Isi .....	76
Gambar 5.19	Halaman Pelanggan – Profil/Akun.....	77
Gambar 5.20	Halaman Pelanggan – Edit Profil .....	77
Gambar 5.21	Halaman Pelanggan – Riwayat Pesanan .....	78

## DAFTAR GAMBAR

Gambar 2.1 <i>Waterfall</i> .....	14
Gambar 3.1 Kerangka Pikir.....	23
Gambar 4.1 <i>Use Case</i> Admin .....	30
Gambar 4.2 <i>Use Case</i> Pelanggan .....	31
Gambar 4.3 <i>Activity Diagram</i> Admin - <i>Login</i> .....	32
Gambar 4.4 <i>Activity Diagram</i> Admin - Produk.....	33
Gambar 4.5 <i>Activity Diagram</i> Admin – Lupa <i>Password</i> .....	34
Gambar 4.6 <i>Activity Diagram</i> Admin – <i>User</i> .....	34
Gambar 4.7 <i>Activity Diagram</i> Admin - Transaksi.....	35
Gambar 4.8 <i>Activity Diagram</i> Admin – Galeri Produk .....	36
Gambar 4.9 <i>Activity Diagram</i> Pelanggan – Profil/Akun .....	37
Gambar 4.10 <i>Activity Diagram</i> Pelanggan – Edit Profil .....	38
Gambar 4.11 <i>Activity Diagram</i> Pelanggan – Home .....	38
Gambar 4.12 <i>Activity Diagram</i> Pelanggan – Keranjang .....	39
Gambar 4.13 <i>Activity Diagram</i> Pelanggan – <i>Login</i> .....	40
Gambar 4.14 <i>Activity Diagram</i> Pelanggan – Registrasi .....	40
Gambar 4.15 <i>Class Diagram</i> .....	41
Gambar 4.16 Desain Tampilan Antarmuka Admin - <i>Login</i> .....	44
Gambar 4.17 Desain Tampilan Antarmuka Admin – Lupa <i>Password</i> .....	44
Gambar 4.18 Desain Tampilan Antarmuka Admin - <i>Dashboard</i> .....	45
Gambar 4.19 Desain Tampilan Antarmuka Admin - Produk .....	45
Gambar 4.20 Desain Tampilan Antarmuka Admin – Tambah Produk.....	45
Gambar 4.21 Desain Tampilan Antarmuka Admin – Edit Produk .....	46
Gambar 4.22 Desain Tampilan Antarmuka Admin – Galeri Produk.....	46
Gambar 4.23 Desain Tampilan Antarmuka Admin - Transaksi .....	46
Gambar 4.24 Desain Tampilan Antarmuka Admin – Edit Transaksi .....	47
Gambar 4.25 Desain Tampilan Antarmuka Admin – Detail Transaksi .....	47
Gambar 4.26 Desain Tampilan Antarmuka Admin - <i>Users</i> .....	47
Gambar 4.27 Desain Tampilan Antarmuka Admin – Edit <i>Users</i> .....	48
Gambar 4.28 Desain Tampilan Antarmuka Pelanggan - <i>Login</i> .....	48

Gambar 4.29 Desain Tampilan Antarmuka Pelanggan - Registrasi .....	48
Gambar 4.30 Desain Tampilan Antarmuka Pelanggan - Home.....	49
Gambar 4.31 Desain Tampilan Antarmuka Pelanggan – Profil/Akun.....	49
Gambar 4.32 Desain Tampilan Antarmuka Pelanggan – Edit Profil .....	49
Gambar 4.33 Desain Tampilan Antarmuka Pelanggan – Riwayat Pesanan .....	50
Gambar 4.34 Desain Tampilan Antarmuka Pelanggan – Keranjang Kosong.....	50
Gambar 4.35 Desain Tampilan Antarmuka Pelanggan – Keranjang Isi .....	50
Gambar 4.36 Desain Tampilan Antarmuka Pelanggan – Detail Produk .....	51
Gambar 5.1 Halaman Admin - <i>Login</i> .....	70
Gambar 5.2 Halaman Admin – Lupa <i>Password</i> .....	70
Gambar 5.3 Halaman Admin - <i>Dashboard</i> .....	71
Gambar 5.4 Halaman Admin - Produk .....	71
Gambar 5.5 Halaman Admin – Tambah Produk.....	71
Gambar 5.6 Halaman Admin – Edit Produk .....	72
Gambar 5.7 Halaman Admin – Detail Produk .....	72
Gambar 5.8 Halaman Admin – <i>Users</i> .....	72
Gambar 5.9 Halaman Admin – Edit <i>User</i> .....	73
Gambar 5.10 Halaman Admin – Transaksi.....	73
Gambar 5.11 Halaman Admin – Transaksi Edit .....	73
Gambar 5.12 Halaman Admin – Detail Transaksi .....	74
Gambar 5.13 Halaman Pelanggan – <i>Login</i> .....	74
Gambar 5.14 Halaman Pelanggan – Registrasi.....	74
Gambar 5.15 Halaman Pelanggan – Home .....	75
Gambar 5.16 Halaman Pelanggan – Detail Produk .....	75
Gambar 5.17 Halaman Pelanggan – Keranjang Kosong.....	76
Gambar 5.18 Halaman Pelanggan – Keranjang Isi .....	76
Gambar 5.19 Halaman Pelanggan – Profil/Akun.....	77
Gambar 5.20 Halaman Pelanggan – Edit Profil .....	77
Gambar 5.21 Halaman Pelanggan – Riwayat Pesanan .....	78

## DAFTAR TABEL

Tabel 2.1 Tabel Ikhtisar .....	5
Tabel 4.1 Analisis <i>Software</i> .....	27
Tabel 4.2 Analisis Data .....	29
Tabel 4.3 Analisis Biaya .....	30
Tabel 4.4 Deskripsi Aktor Pada Aplikasi.....	31
Tabel 4.5 Deskripsi <i>Use Case</i> Admin .....	31
Tabel 4.5 Deskripsi <i>Use Case</i> Pelanggan .....	32
Tabel 4.6 Struktur Tabel <i>Product</i> .....	41
Tabel 4.7 Struktur Tabel <i>Product Galleries</i> .....	42
Tabel 4.8 Struktur Tabel <i>Product Categories</i> .....	42
Tabel 4.9 Struktur Tabel <i>Users</i> .....	42
Tabel 4.10 Struktur Tabel <i>Transactions</i> .....	43
Tabel 4.11 Struktur Tabel <i>Transaction item</i> .....	43
Tabel 5.1 Spesifikasi Perangkat Keras.....	69
Tabel 5.2 Spesifikasi Perangkat Lunak.....	69
Tabel 5.3 Pengujian.....	78



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sejalan dengan cepatnya perkembangan teknologi, para pelaku usaha semakin dipacu untuk menggunakan teknologi sebagai senjata untuk tetap bertahan dan memenangkan persaingan yang semakin hari semakin ketat dan keras. Salah satu bentuk sarana perdagangan di internet yang biasa kita temui disebut dengan *e-commerce*. Dengan memanfaatkan *e-commerce* maka akan memudahkan para calon pembeli untuk melihat, mendapatkan informasi penjualan, dan dapat melakukan transaksi dengan mudah. Dengan adanya fasilitas yang memudahkan ini maka para pembeli dapat menghemat waktu dan biaya, karena pembeli tidak perlu lagi datang ke tempat penjualan untuk mendapatkan apa yang pembeli inginkan. Selain itu para penjual juga dapat memperluas daerah pemasaran produknya dan diharapkan dapat meningkatkan omset penjualan tersebut.

Bisnis peternakan Bebek Alfalah yang berada di lokasi desa Padaulun yang sudah berjalan dari tahun 2015. Peternakan bebek Bebek Alfalah mempunyai usaha ternak dan penjualan daging bebek yang kurang lebih jumlah sangat banyak lebih dari 1000 bebek, usaha ternak bebek dan jual beli daging bebek ini mempunyai daya dukung tinggi di daerah tersebut meliputi ketersediaan bebek dan daging bebek, dalam pemasaran bebek dan daging bebek biasanya di kirim ke restoran dan pedagang kaki lima sehingga adanya beberapa relasi yang selalu memesan kepada Bebek Alfalah dan menjadi pelanggan setia.

Penjualan pada peternakan dan daging bebek masih belum efektif dan efisien dikarenakan dalam proses jual beli masih dilakukan secara langsung di tempat karena jumlah transaksi penjualan masih relatif rendah karena banyak usaha yang masih melakukan promosi dan penjualan masih menggunakan brosur ataupun baliho, maka penulis memutuskan untuk merancang sebuah aplikasi *e-commerce* untuk usaha pemasaran peternakan tersebut yang nantinya dapat memudahkan peternakan Bebek Alfalah dalam melakukan penjualan dan promosi ternak bebek dan daging yang dapat di harapkan memberikan calon pembeli memudahkan untuk mendapatkan informasi tentang peternakan dan daging bebek tersebut.

Berdasarkan permasalahan di atas, maka di buatlah sebuah “Aplikasi Penjualan Binatang Ternak Berbasis Android” sebagai media promosi. Adapun manfaat yang di rasakan oleh Pemilik dan Pelanggan yang membeli ternak dan daging bebek tinggal membuka aplikasi karena secara langsung aplikasi yang menyediakan ternak dan daging bebek, dengan adanya aplikasi tersebut pihak peternakan dapat melakukan dimana saja.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan diatas ,maka dapat diambil suatu rumusan masalah yaitu :

1. Bagaimana merancang dan membangun aplikasi penjualan binatang ternak berbasis android.
2. Bagaimana membuat laporan penjualan binatang ternak.
3. Bagaimana cara membuat media promosi untuk penjualan binatang ternak.

## **1.3 Batasan Masalah**

Adapun batasan masalah yakni sebagai berikut :

1. Aplikasi penjualan ternak berbasis android.
2. Pembuatan aplikasi menggunakan flutter dan laravel.
3. Menggunakan metodologi waterfall
4. Peternakan yang menjadi objek penelitian adalah bebek.

## **1.4 Tujuan Penelitian**

Berdasarkan rumusan masalah sebelumnya, maka tujuan penelitian yang ingin dicapai adalah :

1. Merancang dan membangun aplikasi penjualan binatang ternak untuk bisa membantu permasalahan penjualan binatang ternak.
2. Membuat laporan penjualan binatang ternak dengan berbasis android.
3. Membantu promosi penjualan binatang ternak dengan berbasis android.

## **1.5 Metodologi Penelitian**

Metodologi penelitian adalah cara melaksanakan penelitian berdasarkan fakta ilmiah. Pada penyusunan skripsi ini penulis menggunakan metode penelitian :

### 1.5.1 Lokasi Penelitian

Lokasi penelitian dilakukan di peternakan Bebek Alfalah Kp. Sukalilah RT 03 RW 14 Desa Padaulun Kec. Majalaya.

### 1.5.2 Metode Pengumpulan Data

Adapun teknik-teknik pengambilan data yang peneliti lakukan dalam penelitian adalah sebagai berikut :

a. Metode Observasi

Yaitu dilakukan dengan cara mengamati langsung sistem yang berjalan pada penjualan peternakan di Bebek Alfalah Kp. Sukalilah RT 03 RW 14 Desa Padaulun Kec. Majalaya. Untuk mendapatkan informasi yang dibutuhkan dalam membuat aplikasi penjualan ternak milik Mang Asep.

b. Metode Wawancara

Yaitu melakukan tanya jawab bersama Mang Asep dan pak Rizal selaku pengelola dan pegawai penjualan ternak bebek mengenai sistem yang berjalan di peternakan tersebut. Pengumpulan data secara wawancara adalah usaha untuk mengumpulkan informasi dengan mengajukan sejumlah pertanyaan secara lisan untuk dijawab secara lisan pula.

c. Metode Pustaka

Yaitu pengumpulan data dari beberapa buku, *e-jurnal*, internet dan sumber lainnya yang bersangkutan dengan masalah yang sedang dibahas

### 1.5.3 Metode Pengembangan Sistem

Dalam penelitian kali ini penulis menggunakan metode *waterfall* untuk metode pengembangan sistem aplikasinya. Metode ini merupakan salah satu model yang ada pada metode *System Development Life Cycle* (SDLC) yang memungkinkan untuk departementalisasi dan kontrol suatu langkah-langkah atau proses secara berurutan. Proses pengembangan model fase *one by one*, sehingga meminimalisir kesalahan yang mungkin akan terjadi. Sedangkan untuk kerangka kerja (*framework*) menggunakan *System Development Life Cycle* (SDLC) sehingga sistem yang dibuat dapat terstruktur yang disusun secara runtunan. Adapun tahapan – tahapan dalam metode *waterfall* yaitu : analisa, desain, koding, pengujian, dan pemeliharaan.

## **1.6 Sistematika Penulisan**

Sistematika penulisan pada skripsi ini terbagi menjadi beberapa bab yaitu :

### **BAB I : PENDAHULUAN**

Bab ini berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian dan sistematika penulisan skripsi.

### **BAB II : TINJAUAN PUSTAKA**

Bab ini berisi tentang landasan teori yang digunakan beserta dasar teori yang diperlukan dan bermanfaat sebagai dasar perancangan dan pembuatan aplikasi.

### **BAB III : METODOLOGI**

Bab ini berisi tentang kerangka pikir pada penelitian yang dilakukan beserta deskripsi penjelasan dari kerangka pikirnya.

### **BAB IV : ANALISIS DAN PERANCANGAN**

Bab ini membahas mengenai analisis yang dilakukan serta perancangan dan pengerjaan aplikasi yang akan dibuat.

### **BAB V : IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisi tentang bagaimana pengujian aplikasinya dan bagaimana pengimplementasian di tempat penelitian dilakukan.

### **BAB VI : KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dan saran dari hasil penelitian yang telah dilakukan.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Landasan Teori

Untuk melaksanakan penelitian, peserta skripsi akan menggunakan pengetahuan yang diperoleh baik dari perkuliahan maupun dari sumber-sumber lainnya yang relevan dengan penelitian yang dilakukan. Sebagai landasan teori Pembuatan Aplikasi Penjualan Binatang Ternak Berbasis Android Di Peternakan Bebek Alfalah Desa Padaulun, berdasarkan referensi yang relevan maka dibuatkanlah tabel sebagai berikut:

Tabel 2.1 Tabel Ikhtisar

No	Judul Penelitian	Masalah	Metode	Solusi
1	Judul : Perancangan Aplikasi Penjualan Hasil Panen di Indonesia Berbasis android Penulis : Anna Karenina Jusuf, Mei lestari, Ni Wayan Parwati Tahun : 2022	Penyampaian informasi Hanya lewat baliho, Spanduk	Metode yang digunakan adalah pengumpulan data sekunder, survei, dan wawancara	Pembuatan Aplikasi Penjualan Binatang Ternak Berbasis Android
2	Judul : Perancangan Aplikasi M-Commerce Berbasis Android sebagai media informasi dan penjualan kambing penulis : Retno Waluyu, Zuia Karini, Kharis Ardi Purnomo Tahun : 2018	Belum Memiliki aplikasi android mendukung kegiatan promosi dan sarana komunikasi dalam proses penjualan ternak bebek	Studi lapangan analisa permasalahan, perancangan sistem dan pengujian pelatihan publikasi	Pembuatan aplikasi jual ternak online berbasis android

3	Judul : Perancangan aplikasi Penjualan berbasis android sebagai media pemesanan pada disto <i>nine</i> Penulis : Eni Pudjiarti, Siti Faizah Tahun : 2021	Proses penjualan dan pembelian masih menggunakan cara manual yaitu dengan cara mencari penjual atau pemilik hanya dengan selebaran kertas di jalan terdekat maupun ke TKP	Properti	Pembuatan aplikasi penjualan binatang ternak berbasis android di butuhkan untuk mendapatkan informasi dari lokasi dan harga.
---	---	---	----------	--

Menurut penelitian pada Perancangan Aplikasi Penjualan Hasil Panen Di Indonesia Berbasis Android yang dituliskan oleh : *Anna Karenina Jusuf, Mei Lestari, Ni Wayan Parwati* pada jurnal (*Seminar Nasional Riset dan Inovasi Teknologi (SEMNAS RISTEK) 2022) Vol 6, No 1(2022)*”), Pada penelitiannya adapun masalah yang diangkat dalam penelitian mengatakan Indonesia merupakan negara agraris yang komoditas penduduknya memiliki profesi sebagai petani. Dengan letak negara yang strategis dan berada pada garis khatulistiwa, menjadikan Indonesia memiliki iklim tropis dalam suatu kondisi yang sangat mendukung untuk berkembangnya sektor pertanian. Sektor pertanian di Indonesia memiliki cakupan yang sangat luas, meliputi: Padi, palawijaya, hortikultura, perkebunan, peternakan, budi daya ikan, penangkapan ikan, tanaman kehutanan dan kehutanan lainnya. Hal tersebut menjadikan sektor pertanian sebagai salah satu pilar yang mendukung pergerakan roda ekonomi di Indonesia. Berdasarkan hal tersebut, penulis menemukan ide untuk mengembangkan suatu perancangan aplikasi penjualan hasil panen yang diharapkan dapat membantu petani di Indonesia. Metode pengembangan yang digunakan yaitu metode *Agile* karena metode ini telah terbukti dapat bekerja dengan baik untuk berbagai *project* dengan unsur *uncertainty*. Hasil dari penelitian ini sangat bermanfaat untuk membantu petani di Indonesia dalam beradaptasi dengan perekonomian digital dan meningkatkan penjualan hasil panen.

Menurut penelitian pada Perancang Aplikasi *M-Commerce* Berbasis Android Sebagai Media Informasi dan Penjualan Kambing” yang dituliskan oleh Retno Waluyo, Zulia Karini dan Kharis Ardi Purnomo pada jurnal *CoreIT*, Vol.4, No.1,

Juni (2018), Pada penelitian diatas adapun masalah yang diangkat dalam penelitian mengatakan permintaan hewan kambing meningkat pada saat menjelang hari raya idul adha untuk keperluan qurban. Untuk mendapatkan hewan kambing pembeli mencari di pasar hewan yang hanya buka 2 kali dalam satu minggu. Tujuan penelitian ini membangun aplikasi *m-commerce* berbasis android sebagai media informasi dan penjualan kambing yang dapat mempertemukan antara pembeli dan peternak kambing, sehingga pembeli mendapatkan harga yang murah, terhindar dari Calo dan pembeli mendapatkan kambing yang berkualitas. Metode pengembangan perangkat lunak menggunakan model sekuensial linier dengan tahapan analisis, desain, kode, dan pengujian. Hasil dari penelitian ini aplikasi m-commerce yang digunakan sebagai media informasi dan penjualan kambing, dengan pengujian *Blackbox*.

Menurut penelitian pada “Perancangan Aplikasi Penjualan Berbasis Android Sebagai Media Pemesanan Pada Distro *Online*” yang dituliskan oleh *Eni Pudjiarti, Siti Faizah* dalam jurnal *Bina Insani Ict Journal Vol. 8, No. 2, Desember 2021*” Pada penelitian yang disebutkan diatas, adapun masalah yang diangkat dalam penelitian tersebut mengatakan bahwa sekarang banyak yang ditargetkan untuk masyarakat pengguna *handphone*, dan dalam pemasarannya perusahaan dan bisnis startup membuat suatu media untuk memudahkan masyarakat dalam mengakses konten yang mereka jual yaitu dalam bentuk suatu aplikasi berbasis sistem android. Aplikasi startup ini sangat membantu baik bagi penjualan maupun pembelian dan dapat menjadi solusi bagi para pengusaha untuk memenuhi penjualan mereka di tengah masa pandemi, serta menyebarluaskan aplikasi Distro supaya lebih dikenal luas oleh masyarakat sehingga berminat untuk menggunakan aplikasi Distro. Beberapa perusahaan dan bisnis kecil yang masih berkembang atau biasa disebut usaha mikro ini sudah mulai mencoba cara baru dalam memasarkan apa yang mereka jual dalam pandemic yang masih terjadi saat ini dan salah satunya membuat aplikasi berbasis sistem android. Selain memudahkan penjual memasarkan yang mereka jual, masyarakat yang berperan sebagai pembeli pun dapat dengan mudah mengakses informasi dan melakukan transaksi jual-beli melalui aplikasi tersebut tanpa harus datang ke toko fisik secara langsung.

Berdasarkan uraian di atas, penulis mencoba membuat sebuah rancangan Aplikasi Distro *Online* Berbasis Android dan sistem Aplikasi Android ini dikembangkan dengan menggunakan metode pengembangan *waterfall*, dengan bahasa pemrograman *java* dan *database* yang digunakan adalah *cPanel*.

## **2.2 Dasar Teori**

Pada pembuatan aplikasi penjualan binatang ternak berbasis android di peternakan Bebek Alfalah Desa Padaulun ini penulis mempunyai bahan dan alat untuk pembuatan aplikasi maka penulis mempelajari teori-teori yang ada sebagai acuan dalam Implementasi aplikasi yang akan penulis buat yaitu sebagai berikut:

### **2.2.1 Pengertian Aplikasi**

Aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya (Syabania & Rosmawarni, 2021, hlm. 45).

Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan. (Andi, 2015).

Aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia.(Huda & Priyatna, 2019)

### **2.2.2 Pengertian Penjualan**

Penjualan adalah sebuah sistem yang melibatkan sumber daya di dalam suatu organisasi, prosedur, data, maupun sarana pendukung untuk mengoperasikan sistem penjualan, sehingga menghasilkan informasi yang berguna bagi pihak manajemen di dalam pengambilan suatu keputusan yang diinginkan (Anggraini dkk., 2020, hlm. 65).

Penjualan adalah proses dimana sang penjual memuaskan segala kebutuhan dan keinginan pembeli agar dicapai manfaatnya bagi yang penjual maupun sang



pembeli yang berkelanjutan dan yang menguntungkan bagi kedua belah pihak. Penjualan juga hasil yang dicapai sebagai imbalan jasa-jasa yang diselenggarakan perniagaan transaksi dunia usaha (Ahmad & Hasti, 2018, hlm. 69).

Penjualan merupakan kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut dan penjualan dapat diartikan sebagai pengalihan atau pemindahan hak kepemilikan atas barang atau jasa dari pihak penjual ke pembeli.(Arfianto & Nugrahanti, 2018).

### **2.2.3 Pengertian Binatang Ternak**

Menurut KBBI (2015) Binatang ternak adalah berbagai bentuk binatang yang sengaja untuk dipelihara dan di perkembangbiakan dengan tujuan untuk mendapatkan hasil melalui produksi dan pemeliharaan yang dilakukan. Hewan ternak atau *rajakaya* dalam bahasa Jawa adalah hewan yang dengan sengaja dipelihara sebagai sumber pangan, sumber bahan baku industri, atau sebagai pembantu pekerjaan manusia. Usaha pemeliharaan ternak disebut sebagai peternakan (atau perikanan) dan bagian dari kegiatan pertanian secara umum.

Ternak dapat berupa binatang apa pun (termasuk serangga dan vertebrata tingkat rendah seperti ikan dan katak). Namun, dalam percakapan sehari-hari biasanya merujuk kepada unggas dan mamalia seperti ayam, angsa, kalkun, atau itik untuk unggas, serta babi, sapi, kambing, domba, kuda, atau keledai untuk mamalia. Di beberapa daerah di dunia juga dikenal hewan ternak yang khas seperti unta, llama, bison, burung unta, dan tikus belanda. Jenis ternak bervariasi di seluruh dunia tergantung pada sejumlah faktor seperti iklim, permintaan Pelanggan, daerah asal, budaya lokal, dan topografi.

Kelompok hewan selain unggas dan mamalia yang dipelihara juga disebut hewan ternak, khususnya apabila dipelihara di tempat khusus. Penyebutan "ternak" biasanya dianggap "tepat" apabila hewan yang dipelihara sedikit banyak telah mengalami domestikasi, tidak sekadar diambil dari alam liar kemudian dipelihara. Dalam kelompok ini termasuk ngengat sutera, berbagai jenis ikan air tawar (seperti ikan mas, gurami, mujair, nila, atau lele), beberapa jenis katak (terutama bullfrog), buaya, dan beberapa jenis ular. Usaha pemeliharaan ikan umumnya disebut sebagai perikanan atau, lebih spesifik, budidaya ikan.

#### 2.2.4 Sejarah Android

Android adalah sistem operasi berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance terdiri dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. (Kusniyati & Pangondian Sitanggang, 2016).

#### 2.2.5 Pengertian Andorid

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc dengan dukungan finansial Google, yang kemudian dibeli pada tahun 2005 (Kusniyati & Pangondian Sitanggang, 2016).

Android merupakan *toolkit* perangkat lunak *open source* untuk perangkat *mobile*, android sendiri diciptakan oleh perusahaan *Google* yang bekerja sama dengan *Open Handset Alliance*. Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile device*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi. Sistem operasi Android diluncurkan bersamaan dengan dibentuknya organisasi Operasi *Handset Alliance* tahun 2007. Selain *Google*, beberapa nama-nama besar juga ikut serta dalam *Open Handset Alliance*, Antara lain Motorola, Samsung, LG, *Ericsson*, *T-Mobile*, Vodafone, Toshiba, dan Intel (Huda & Apriyanto, 2019).

#### 2.2.6 Perkembangan Android

Perkembangan Android Menurut (Kusniyati & Pangondian Sitanggang, 2016) dikemukakan yaitu :

1. Android 1.0 *Alpha* pertama kali dirilis pada September 2008 dan tidak ditujukan untuk kebutuhan komersial. Walaupun belum dirilis secara komersial, versi Android ini telah dilengkapi fitur dukungan akses web browser, *streaming* youtube, pemutar media, google map, dan sinkronisasi dengan aplikasi google lainnya.
2. Android Versi 1.1 Beta diluncurkan pada tanggal 9 Februari 2009. Sama seperti pendahulunya yakni *Alpha* versi Android belum dirilis secara komersial, hanya diperuntukkan untuk satu perangkat. Pembaruan ini dilakukan untuk memperbaiki *bugs* dan meningkatkan beberapa fitur seperti rincian lokasi pada aplikasi map serta fitur menyembunyikan dan menampilkan tombol panggilan.
3. Android Versi 1.5 (*Cupcake*) rilis pada bulan Mei 2009.
4. Perilisan Android ke 4 yaitu versi 1.6 (*Donut*) dirilis pada bulan September 2009.
5. Android versi 2.0/2.1 (*Eclair*) Masih ditahun yang sama, Android kembali merilis sistem operasi versi terbarunya, yaitu Android versi 2.0/2.1 *Eclair*. Android *Eclair* diluncurkan oleh Google 3 bulan setelah peluncuran *Donut*.
6. Android versi 2.2 (*Froyo: Frozen Yoghurt*) dirilis pada tanggal 20 Mei 2010.
7. Android versi 2.3 (*Gingerbread*) yang dirilis 7 bulan kemudian.
8. Android versi 3.0/3.1 (*Honeycomb*) Pada bulan Mei 2011 Android versi 3.0/3.1 atau Android *Honeycomb* dirilis. Android *Honeycomb* merupakan sebuah sistem operasi Android yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis Android.
9. Android versi 4.0 (ICS: *Ice Cream Sandwich*) dirilis pada tahun yang sama dengan *Honeycomb*, yaitu pada bulan Oktober 2011.
10. Android versi 4.1 (*Jelly Bean*) Android *Jelly Bean* merupakan versi Android lanjutan dari ICS. Salah satu gadget yang menggunakan sistem operasi *Jelly Bean* adalah Google Nexus 7 yang diprakarsai oleh ASUS, vendor asal Taiwan yang juga menjadi teman satu kampung halaman dengan Acer.

11. Android versi 4.4 (KitKat) diluncurkan pada 4 September 2013, sebelumnya banyak kabar beredar akan bernama *Key Lime Pie* namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga Namanya diganti dengan OS Android KitKat.
12. Android versi 5.0.2 (*Lollipop*) merupakan versi OS Android yang menjadi tren baru di industri *smartphone*, hal ini tak lepas dari banyaknya keunikan serta kelebihan yang terdapat pada versi tersebut. Kehadiran android versi ini sangat di nantikan oleh sekian banyak orang karena diharapkan sistem operasi *Lollipop* ini bisa lebih baik dibandingkan versi-versi yang sebelumnya.
13. Android 6.0 *Marshmallow* Pertama kali diperkenalkan Mei 2015 di Google I/O di bawah kode nama Android M, secara resmi dirilis pada Oktober 2015. Android *Marshmallow* memperkenalkan model izin aplikasi yang didesain ulang hanya ada delapan kategori izin, dan aplikasi tidak akan otomatis diberikan semua hak akses pada saat instalasi.
14. Android 7.0 *Nougat* Diperkenalkan pada tanggal 19 Oktober 2016 berfokus pada peningkatan performa *user interface* sehingga lebih intuitif dan penggunaan aplikasi secara bersamaan pada fitur *multi window*. Selain peningkatan fitur tadi, Android *Nougat* juga menambahkan beberapa fitur lain seperti dukungan cahaya malam atau mode malam, *keyboard default* yang dapat mengirim animasi GIF langsung dan panggilan *multi-endpoint*.
15. Android 8.0 *Oreo* diluncurkan pada bulan Agustus 2017. *User interface* pada Android *Oreo* lebih simpel agar memudahkan dalam mengakses aplikasi. Pembaruan pada versi *Oreo* membawa beberapa fitur seperti *Autofill* yang memberikan kemudahan dalam mengisi formulir, dukungan gambar dalam gambar dan pengoptimalan *booting* agar lebih cepat.
16. Android 9.0 *Pie* dirilis pada bulan Agustus 2018. Fitur unggulan pada versi ini adalah kemampuan AI atau kecerdasan buatan. Dengan fitur AI *smartphone* akan menganalisa dan mempelajari pola pemakaian penggunaan *smartphone* secara otomatis. Selain itu fitur lainnya seperti *Adaptive Brightness* yang akan menyesuaikan kecerahan layar secara otomatis dan dukungan pada ponsel *bezel less*.

17. Android 10 Q tidak diberi nama berkaitan dengan makanan manis. Versi Android diberi nama Android 10 Q untuk memperingati bahwa Android telah mencapai 1 dekade secara komersial. Versi Android 10 lebih berfokus pada penyempurnaan mode malam atau gelap serta peningkatan fitur *sound amplifier* untuk mengatur kualitas audio.
18. Android 11 *Red Velvet Cake*, Versi Android dengan nama makanan (versi *internal*) ini dirilis resmi pada 8 September 2020. Android 11 bisa mengelompokkan aplikasi di notifikasi. Misal semua aplikasi pesan digabung dalam notifikasi *Conversation*. Android 11 memiliki fitur Balon yang bisa membuat melanjutkan percakapan setelah mengakses aplikasi lain. Akses *chat* kapan saja dan di mana saja lebih mudah. Android 11 juga punya fitur perekam layar (*Screen Recording*) bawaan. Ada juga fitur izin satu kali ke aplikasi yang meminta akses.
19. Android 12 *Snow Cone* dirilis perdana pada 4 Oktober 2021. Android versi ini memiliki *update* besar pada Desain Material yang disebut “Material You”. Sistem operasi bisa secara otomatis menghasilkan tema warna untuk menu sistem dan aplikasi yang didukung menggunakan warna *wallpaper* pengguna. Android 12 juga dilengkapi fitur untuk ambil *screenshot* tampilan web secara utuh atau bergulir. Pengguna juga bisa mencegah aplikasi untuk menggunakan mikrofon melalui pengaturan cepat.
20. Android 13 *Tiramisu* diperkenalkan pada 10 Februari 2022 yang berfokus pada penyempurnaan fitur Android 12, meningkatkan privasi, keamanan, dan optimalisasi UI. Android 13 memiliki fitur pemilih foto baru yang memungkinkan bisa atur foto atau video tertentu untuk dibagikan dengan aplikasi berbeda. Selain itu, bisa kustomisasi ikon aplikasi dengan warna yang sama seperti tema atau *wallpaper*.

### 2.2.7 Fitur Pada Android

Android merupakan sistem operasi *open source* dan bebas untuk di kostumisasi. Oleh karena itu, tidak ada konfigurasi standar untuk *software* dan *hardware* nya, Android mendukung beberapa fitur dasar yaitu:

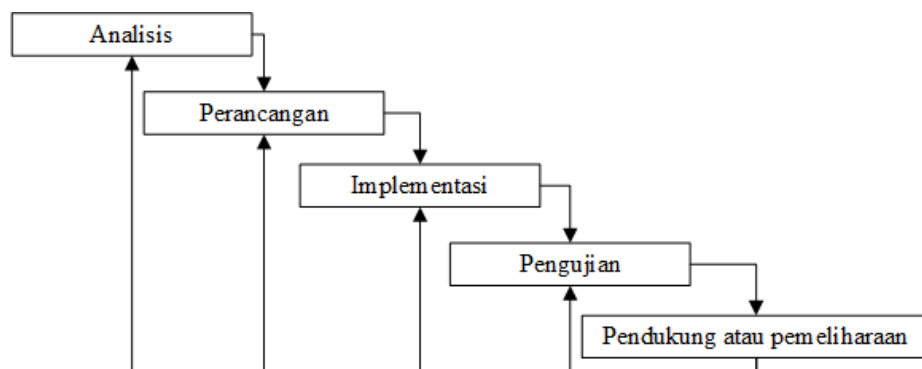
1. *Storage* menggunakan SQLite, karena SQLite merupakan *database* yang ringan untuk penyimpanan data.

2. *Connectivity* mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, WiFi, LTE, WiMAX.
3. *Messaging* mendukung SMS dan MMS
4. Menggunakan *open source webkit* dan *chrome v8 Javascript engine*
5. *Media Support* mendukung beberapa media seperti AAC, HE-AAC, MPEG-4 SP, AMR, AMR-WB, WAV, MP3, JPG, PNG *Hardware Support- accelerometer* sensor, kamera, digital kompas, *proximity* sensor, GPS mendukung *multi-touch screens*
6. Mendukung aplikasi *multi-tasking*
7. Android 2.3 mendukung *Flash* 10.1
8. Mendukung untuk *sharing* internet (*Tethering*)

### 2.2.8 Waterfall

*Waterfall* merupakan salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan yaitu setiap fase dalam *waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya (Tabrani & Aghniya, 2019).

Model *waterfall* adalah salah satu jenis pengembangan aplikasi dan termasuk dalam siklus hidup klasik, dimana kemajuan suatu proses dipandang terus mengalir ke bawah seperti air terjun. Dalam Model *Waterfall*, setiap tahap harus berurutan, dan tidak dapat melompat ke tahap berikutnya, *waterfall* harus menyelesaikan tahap pertama baru lanjut ke tahap ke dua dan seterusnya.



Gambar 2.1 *Waterfall*

#### a. Analisis

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

b. Perancangan

Proses yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

c. Implementasi

Proses translasi perancangan menjadi sebuah program komputer sesuai dengan desain yang telah dibuat.

d. Pengujian

Pengujian fokus pada perangkat lunak dari segi *logic* dan fungsional dan memastikan semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (eror) dan memastikan *output* dihasilkan sesuai.

e. Pendukung atau pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

### 2.2.9 Flutter

Flutter adalah platform yang digunakan para developer untuk membuat aplikasi *multiplatform* hanya dengan satu basis koding (*codebase*). Artinya, aplikasi yang dihasilkan dapat dipakai di berbagai platform, baik *mobile* Android, iOS, web, maupun desktop (Shinta, 2021, hlm. 2)

Flutter adalah *Framework* UI berbasis *open-source* yang dikembangkan Google, dan berjalan menggunakan bahasa pemrograman Dart. Flutter adalah jenis *framework* yang cocok digunakan untuk mengembangkan desain UI *website* maupun aplikasi. Karena menggunakan lisensi *open-source* Flutter dapat bebas digunakan oleh siapa saja di berbagai platform. Sejak dirilis tahun 2017, Google menargetkan Flutter sebagai solusi cepat dan mudah bagi developer dalam

mendesain UI yang kompatibel dengan berbagai perangkat, misalnya *mobile*, web, dan desktop. *Framework* ini sudah digunakan oleh berbagai perusahaan besar, misalnya Alibaba, Ebay, Baidu, Google Ads dan lainnya. Flutter menawarkan beberapa kemudahan, seperti adanya *library* untuk menyimpan elemen *widget* seperti tombol, teks, *slider*, dan lainnya. Flutter juga memiliki fasilitas SDK alat untuk meng-*compile* kode ke dalam perangkat yang dikenal dengan istilah ‘*one-block-code*’, untuk kemudian dapat dijalankan pada *multi-platform*.

#### 2.2.10 PHP

PHP berasal dari kata “*Hypertext Preprocessor*”, yaitu bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. Saat ini, PHP banyak dipakai untuk membuat program situs web dinamis. Contoh Aplikasi PHP adalah forum(*phpBB*). Sedangkan, *mambo*, Joomla!, *Postnuke*, *Xaraya*, dan lain – lain merupakan contoh aplikasi yang lebih kompleks (Mundzir, 2018, hlm. 3).

PHP sebagai sekumpulan *script* atau bahasa program memiliki fungsi utama yaitu mampu mengumpulkan dan mengevaluasi hasil survei atau bentuk apa pun ke server *database* dan pada tahap selanjutnya akan menciptakan efek beruntun. Efek beruntun PHP ini berupa tindakan dari *script* lain yang akan melakukan komunikasi dengan *database*, mengumpulkan dan mengelompokkan informasi, kemudian menampilkannya pada saat diperlukan oleh *user*.

PHP diciptakan oleh Rasmus Lerdorf di tahun 1995. Pada masa itu, PHP merupakan kependekan dari *Personal Home Page* atau biasa dikenal sebagai situs personal bernama *Form Interpreted* (FI) memiliki wujud berupa sekumpulan *script* yang digunakan untuk mengolah data formulir dari web.

#### 2.2.11 Laravel

Laravel adalah sebuah *framework* PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintak yang ekspresif, jelas dan menghemat waktu (Hermanto, 2019, hlm. 19).



Menurut situs web resimnya yaitu *laravel.com*, Laravel adalah kerangka kerja aplikasi web dengan sintak yang ekspresif dan elegan. Laravel telah memberikan kebebasan kepada pengguna untuk berkreasi tanpa melakukannya dari hal-hal kecil atau awal. Sintak kode pada laravel dibuat dengan serapi mungkin, itu karena laravel berfokus pada kerapian sintak agar memudahkan para pengembang untuk membaca sintak kode pada laravel. Selain itu, laravel juga memiliki integritas dengan berbagai API yang tersedia agar memudahkan penggunaan API dengan kerangka kerja laravel. Sama seperti halnya kerangka kerja web yang lain, laravel juga memiliki kemampuan untuk mengelola *database*, *websocket*, autentifikasi, dan sebagainya.

### 2.2.12 Tailwind

Tailwind CSS, yaitu framework yang bersifat *utility-first* untuk membangun desain antarmuka khusus dengan cepat. Alasan pemilihan *framework* ini dikarenakan Tailwind memiliki kelebihan seperti, lebih sederhana jika dibandingkan dengan *framework* CSS lainnya, responsif, *component friendly*, dan didesain untuk disesuaikan (Listiyah, 2022, hlm. 44).

Tailwind CSS merupakan salah satu *framework* populer di kalangan para *programmer*. Tailwind dikemas dengan *classes* seperti *flex*, *pt-4*, *text-center*, dan *rotate-90* yang dapat dikomposisikan untuk membuat berbagai desain langsung pada *markup*. *Framework* ini diciptakan oleh Adam Wathan dengan desain visual komponennya dibuat oleh Steve Schoger serta diluncurkan sekitar pertengahan 2020 lalu. Pada Tailwind CSS tidak akan menemukan *pra*-desain komponen seperti *framework* kebanyakan. Hal ini sesuai dengan klaim mereka “*utility-first*” yang di mana Tailwind tidak menyediakan komponen utuh. Tailwind menyediakan *class-class* kecil yang bisa digunakan dengan sangat fleksibel untuk membuat *interface* sesuai keinginan tanpa perlu *override styles* seperti *framework* kebanyakan. Jadi, ketika ingin membuat komponen, maka perlu menggunakan beberapa *class* kecil hingga tercipta komponen yang di inginkan.

### 2.2.13 MySQL

MySQL disebut juga SQL yang merupakan singkatan dari *Structured Query Language*. bahasa terstruktur yang khusus digunakan untuk mengolah *database*.

SQL pertama kali didefinisikan oleh American National Standards Institute (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen *database* yang bersifat *open source* dan *relational* artinya data yang dikelola dalam *database* yang akan diletakkan pada tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat.

SQL juga dapat diartikan sebagai antar muka standar untuk sistem manajemen relasional. SQL lebih mudah digunakan dibandingkan dengan bahasa pemrograman, tetapi rumit dibandingkan *software* lembar kerja dan pengolahan data. Sebuah pernyataan SQL yang sederhana dapat menghasilkan set permintaan untuk informasi yang tersimpan pada komputer yang berbeda di berbagai lokasi yang tersebar, sehingga membutuhkan waktu dan sumber daya komputasi yang banyak. SQL dapat digunakan untuk investigasi interaktif, atau pembuatan laporan atau disisipkan dalam program aplikasi.(Novendri, 2019).

#### **2.2.14 Balsamiq Mockup**

Balsamiq merupakan salah satu *tool* yang biasa digunakan oleh para UI *designer* ataupun UX *designer* untuk merancang desain tampilan aplikasi yang akan dibuat. Menurut website resmi Balsamiq yaitu <https://balsamiq.com/>. “Balsamiq Mockups adalah alat *wireframing* cepat yang membantu Anda bekerja lebih cepat & lebih pintar”. Balsamiq Mockups menciptakan pengalaman sketsa di papan tulis, tetapi menggunakan komputer yang akan membuat mockups menjadi cepat dan juga mudah.

#### **2.2.15 UML (*Unified Modelling Language*)**

*Unified Modelling Language* (UML) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Informasinya dapat berupa model, deskripsi atau perangkat lunak dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan kesatuan dari bahasa permodelan yang dikembangkan oleh booch, object modeling technique (omt) dan object oriented software engineering (oose). metode booch dari grady boochsss sangat terkenal dengan nama metode





design object oriented. metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu: identifikasi kelas-kelas dan objek-objek, identifikasi semantik dari hubungan objek dan kelas tersebut, perincian interface dan implementasi. keunggulan metode booch adalah pada detail dan kayanya dengan notasi dan elemen. permodelan omt yang dikembangkan oleh rumbaugh didasarkan pada analisis terstruktur dan pemodelan entity-relationship. tahapan utama dalam metodologi ini adalah analisis, design sistem, design objek dan implementasi.



Ada empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu use case diagram, sequence diagram, class diagram, dan activity diagram.

#### 1. *Use case*

*Use Case* rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah laku benda atau *thing* dalam sebuah mode serta direalisasikan oleh sebuah *collaborator*, umumnya *use case* digambarkan dengan sebuah *elips* dengan garis yang *solid*.

Tabel 2.1 Simbol - Simbol dalam Use Case Diagram



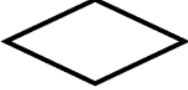


No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem, menghasilkan suatu hasil terukur bagi suatu <i>Actor</i> .
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Association</i>	Penghubung antara objek satu dengan objek lainnya.

5		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
6		<i>Extend</i>	Menspesifikasikan bahwa <i>use case target</i> memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan

## 2. Activity Diagram

*Activity* diagram yaitu menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity* diagram dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram.

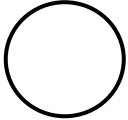

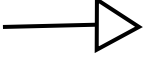

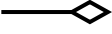

Tabel 2.2 Simbol - Simbol dalam Activity Diagram

No	Simbol	Nama	Keterangan
1		Status Awal	Aktivitas yang dilakukan sistem, aktivitas diawali dengan kata kerja
2		Aktivitas	Deskripsi dari urutan aksi - aksi yang ditampilkan sistem,
3		Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status Akhir	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

### 3. Class Diagram

*Class* diagram menggambarkan struktur dan deskripsi *class* dan objek hubungan antara lain pewarisan, asosiasi dan lain-lain. *Class* diagram berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek lain. Objek adalah nilai tertentu dari setiap atribut *class entity*.

Tabel 2.3 Simbol - Simbol dalam Class Diagram

No	Simbol	Nama	Keterangan
1	Nama_kelas	Kelas	Kelas pada struktur sistem.
	+atribut		
	+operasi()		
2		Antarmuka / <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3		Asosiasi berarah / <i>directed association</i>	Relasi antar kelas dengan makna kelas yang digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
5		Kebergantungan / <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
6		Agresi / <i>aggretrion</i>	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )
7		Asosiasi / <i>association</i>	Relasi antar class dengan arti umum

#### 2.2.16 Black Box Testing

Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Metode *Black box Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan, Estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. (Cholifah dkk., 2018) .

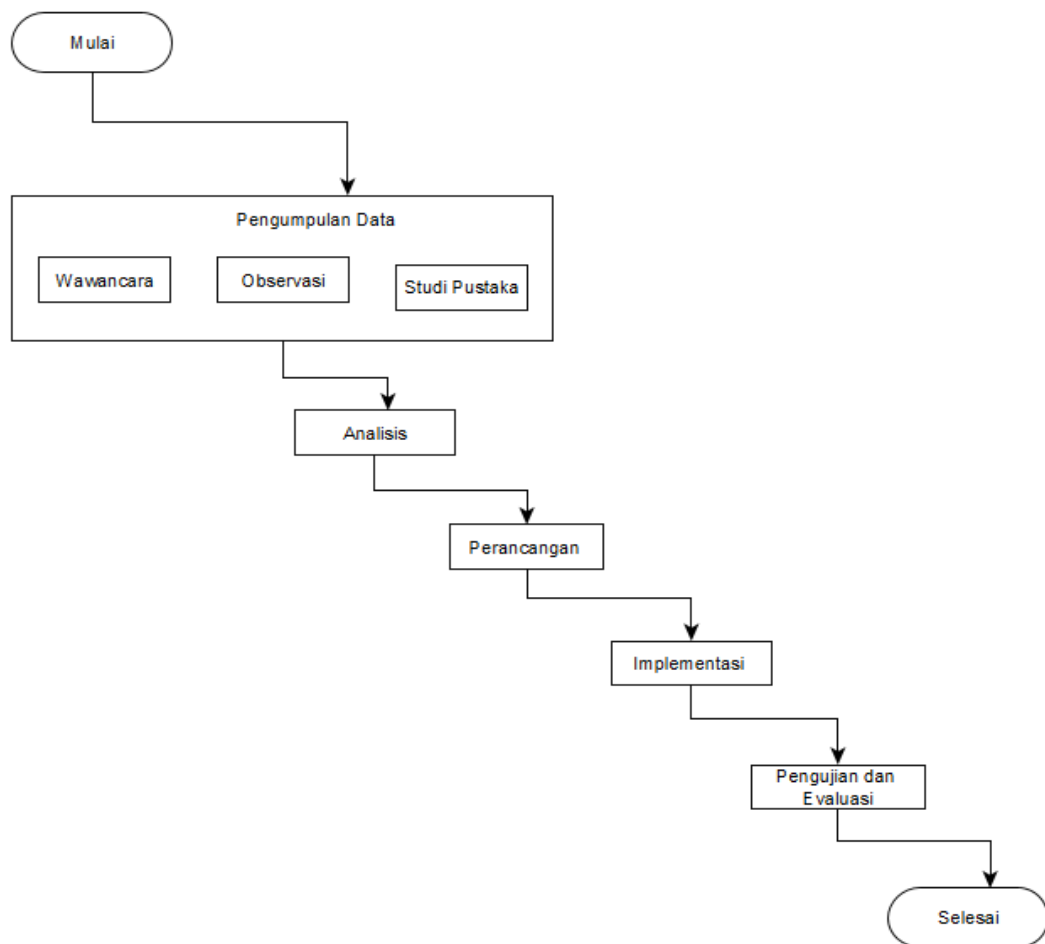
Black Box testing adalah uji coba terhadap fungsionalitas sebuah aplikasi atau program apa pun yang sedang dikembangkan. Dengan dilakukannya pengujian *black box* maka akan diketahui apakah semua fungsi yang ada pada perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Aldiansyah dkk., 2022).

## BAB III

### METODOLOGI

#### 3.1 Kerangka Pikir

Kerangka pikir merupakan jalur pemikiran yang dirancang berdasarkan kegiatan peneliti yang dilakukan. Berikut adalah kerangka pikir yang merupakan langkah-langkah yang akan dilakukan dalam penyelesaian penelitian ini. Adapun kerangka kerja penelitian dengan mengadaptasi metode SDLC model *waterfall* yaitu :



Gambar 3.1 Kerangka Pikir

#### 3.2 Deskripsi

Deskripsi merupakan sebuah kaidah yang mempunyai hubungan dengan adanya upaya pengolahan data menjadi sebuah hal yang dapat dikemukakan dan diutarakan dengan cara yang jelas serta tepat guna mencapai suatu tujuan tertentu

sehingga dimengerti dan dipahami oleh orang yang memang tidak langsung mengalami hal yang dideskripsikan tersebut.

### **3.2.1 Pengumpulan Data**

Pada tahapan ini penulis melakukan pengumpulan data menggunakan beberapa metode sekaligus yaitu metode wawancara, observasi dan studi pustaka. Untuk lebih jelasnya mengenai metode yang dilakukan, berikut penjelasannya :

#### **1. Observasi**

Metode Observasi ini adalah metode pengumpulan data dengan mengamati langsung di lokasi atau tempat penelitian dilakukan. Pada tahap ini akan dilakukan pengumpulan data dengan mengambil bukti beberapa foto gambar serta video dengan mengamati langsung keadaan di lokasi seperti apa agar bisa dijadikan sampel nanti pada tahap uji coba.

#### **2. Wawancara**

Pada tahap ini wawancara dilakukan secara langsung dengan proses tanya jawab terhadap pihak-pihak yang bersangkutan di lokasi peternakan tempat penelitian dilakukan.

#### **3. Studi Pustaka**

Metode Studi Pustaka adalah metode untuk melakukan pengumpulan data dengan membaca buku Pustaka dan sumber ilmu dalam bentuk tulisan lainnya yang merupakan penunjang dalam memperoleh data untuk melengkapi dalam pembangunan pendataan penduduk maupun dalam penyusunan laporan yang berhubungan dengan masalah yang diteliti.

### **3.2.2 Analisis**

Pada tahap ini penulis melakukan pengumpulan data yang dimana pada tahapan analisis data dilakukan agar dapat mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan- hambatan, yang terjadi dan kebutuhan-kebutuhan yang diperlukan sehingga diharapkan dapat melakukan penelitian secara lancar.

#### **1. Identifikasi Masalah**

Pada tahap ini, penulis dapat mengidentifikasi permasalahan yang ada di lokasi penelitian yaitu di peternakan Bebek Alfalah yang ada di Desa Padaulun



sistem penjualan yang dilakukannya masih secara langsung dilakukan di tempat (peternakan). Selain itu, media promosi yang dilakukan juga masih menggunakan brosur ataupun baliho.

## **2. Analisis Sistem**

Pada tahapan ini juga penulis perlu melakukan analisis kebutuhan-kebutuhan apa saja untuk penunjang pembuatan aplikasi penjualannya. Selain menentukan kebutuhan-kebutuhan yang diperlukan, penulis juga perlu menentukan target pengguna, analisis data yang akan digunakan, analisis biaya yang diperlukan, analisis perangkat lunak apa saja yang diperlukan untuk melakukan pembuatan aplikasi penjualannya, antarmuka serta fitur-fiturnya agar aplikasi yang dibuat sesuai dengan yang di harapkan dan di rancang.

### **3.2.3 Perancangan**

Dalam tahap perancangan ini yaitu dilakukannya perancangan berdasarkan data yang telah di analisis. Sistem yang akan dibentuk oleh penulis berupa gambaran desain, pembuatan sketsa, perencanaan dari sistem yang akan dibangun sesuai analisis yang dilakukan yaitu membuat pembuatan aplikasi penjualan binatang ternak berbasis android di peternakan Bebek Alfalah desa Padaulun.

### **3.2.4 Implementasi**

Pada tahap ini dilakukan implementasi koding berdasarkan perancangan yang telah dibuat dan diterjemahkan ke dalam bahasa pemrograman dan dengan dilakukan menggunakan flutter yang digunakan untuk memproses data dan informasi dari hasil analisis dan perancangan.

### **3.2.5 Pengujian dan Evaluasi**

Pada tahap terakhir dilakukan pengujian atau testing terhadap sistem yang telah selesai dibangun. Tahap ini dilakukan setelah sistem yang dibuat telah selesai dalam memenuhi kebutuhan sesuai data, analisis, dan juga perancangan. Tahap ini diperlukan evaluasi untuk mengetahui apakah masih terdapat kesalahan dalam rancangan logika atau program apabila masih terdapat kesalahan akan dilakukan perbaikan-perbaikan sehingga program yang dibangun siap dan layak untuk digunakan.

## **BAB IV**

### **ANALISIS DAN PERANCANGAN**

#### **4.1 Analisis**

Analisis dalam pembuatan aplikasi penjualan binatang ternak menggunakan metode *System Development Life Cycle (SDLC)* model *waterfall*. Dalam analisis ini, penulis menggunakan beberapa tahapan yaitu analisis masalah, analisis *software*, analisis pengguna, *user interface*, fitur-fitur, analisis data dan analisis biaya.

##### **4.1.1 Analisis Masalah**

Bisnis peternakan Bebek Alfalah yang berada di lokasi desa Padaulun yang sudah berjalan dari tahun 2015. Peternakan Bebek Alfalah mempunyai usaha ternak dan penjualan daging bebek yang kurang lebih jumlah sangat banyak lebih dari 1000 bebek, usaha ternak bebek dan jual beli daging bebek ini mempunyai daya dukung tinggi di daerah tersebut meliputi ketersediaan bebek dan daging bebek, dalam pemasaran bebek dan daging bebek biasanya di kirim ke restoran dan pedagang kaki lima sehingga adanya beberapa relasi yang selalu memesan kepada Bebek Alfalah dan menjadi pelanggan setia.

Penjualan pada peternakan dan daging bebek masih belum efektif dan efisien dikarenakan dalam proses jual beli masih dilakukan secara langsung di tempat karena jumlah transaksi penjualan masih relatif rendah karena banyak usaha yang masih melakukan promosi dan penjualan masih menggunakan brosur ataupun baliho, maka penulis memutuskan untuk merancang sebuah aplikasi *e-commerce* untuk usaha pemasaran peternakan tersebut yang nantinya dapat memudahkan peternakan Bebek Alfalah dalam melakukan penjualan dan promosi ternak bebek dan daging yang dapat di harapkan memberikan calon pembeli memudahkan untuk mendapatkan informasi tentang peternakan dan daging bebek tersebut.

##### **4.1.2 Analisis Software**

Analisis *software* dilakukan untuk mengetahui *software* apa saja yang digunakan. *Software* atau perangkat lunak yang digunakan pada penelitian ini yaitu :

Tabel 4.1 Analisis *Software*

Sistem Operasi	Windows 8.1
Kode Editor	Visual Studio Code
Browser	Google Chrome
<i>Database</i>	MySQL
Bahasa Pemrograman	PHP
Diagram-Diagram	Microsoft Visio
<i>Mockup Antarmuka</i>	Balsamiq Mockup
<i>Framework-Web</i>	Laravel
<i>Framework-Android</i>	Flutter

#### 4.1.3 Analisis Pengguna

Analisis pengguna dilakukan untuk mengetahui siapa saja yang akan menggunakan aplikasinya, diantaranya :

1. Admin yaitu pengguna yang bisa mengakses semua menu dan fitur serta dapat melakukan pengelolaan data pada aplikasi penjualan binatang ternak yang dapat diakses berbasis web.
2. Pelanggan yaitu pengguna yang bisa mengakses menu dan fitur aplikasi penjualan binatang ternak berbasis android yang dapat melakukan transaksi pembelian.

#### 4.1.4 User Interface

*User Interface* pada aplikasi yang dibuat berbeda antara pengguna admin dengan Pelanggan. *User interface* untuk admin berbasiskan web yang artinya dapat diakses menggunakan platform apa pun. Sedangkan untuk pengguna Pelanggan aplikasinya berbasiskan android yang artinya hanya dapat digunakan melalui platform android saja.

*User Interface* dari aplikasi untuk Pelanggan sangat berpengaruh terhadap minat Pelanggan dalam menentukan transaksi pembelian. Hal ini mencakup kesederhanaan antarmuka dan kedetailan data yang ditampilkan di layar perangkat Pelanggan guna tampilan masukan dan keluaran dari aplikasi. Karena aplikasi penjualan yang dibuat ditujukan untuk Pelanggan, maka *user interface* aplikasinya juga harus dibuat sesederhana mungkin agar tidak membingungkan Pelanggan tetapi tetap tidak menghilangkan kedetailan data yang dijual. Sedangkan *user*

*interface* aplikasi untuk admin juga haruslah dibuat sesederhana mungkin guna memudahkan admin dalam melakukan pengelolaan data produk yang akan dijual atau ditampilkan di layar aplikasi Pelanggan.

#### 4.1.5 Fitur-Fitur

Fitur-fitur yang digunakan dalam aplikasi penjualan binatang ternak ini dimaksudkan agar Pelanggan dapat memudahkan dalam melakukan transaksi pembelian produk. Selain itu, pada aplikasi penjualan binatang ternak ini juga dimaksudkan agar memudahkan penjual dalam melakukan promosi binatang ternak yang dijualnya.

Pada aplikasi penjualan binatang ternak ini menyediakan fitur registrasi untuk Pelanggan yang akan menggunakan aplikasi ini tetapi belum memiliki akunnya. Aplikasi penjualan binatang ternak ini memiliki fitur keranjang produk yang dibeli atau akan dibeli, riwayat pemesanan yang telah dilakukan, mengedit informasi akun Pelanggan, detail produk, daftar produk dan sebagainya.

Sedangkan untuk pengguna admin, aplikasi penjualan binatang ternak ini menyediakan fitur yang difokuskan untuk melakukan pengelolaan data produknya. Lebih lengkapnya fitur-fitur yang ada pada aplikasi penjualan binatang ternak ini adalah :

##### Pelanggan

- Registrasi, *login*, *logout*, serta edit profil atau akun.
- *Dashboard* yang menampilkan daftar produk yang tersedia untuk dibeli.
- Riwayat pemesanan produk yang telah dilakukan.
- Detail produk
- Keranjang produk

##### Admin

- *Login*, *logout*, serta lupa *password* untuk akun admin.
- *Dashboard* yang menampilkan informasi mengenai total transaksi, total Pelanggan, total pembayaran dan total produk.
- Pengelolaan data produk.
- Pengelolaan data *user*.
- Data transaksi yang disertai fitur laporannya.

#### 4.1.6 Analisis Data

Berikut adalah analisis data berupa data masukan, proses dan keluaran sebagai penunjang pada aplikasi ini yaitu :

Tabel 4.2 Analisis Data

Masukan	Proses	Keluaran
Pelanggan maupun admin menginput data <i>login</i>	Aplikasi memvalidasi data <i>login</i> apakah valid atau tidak	Aplikasi menampilkan halaman <i>dashbboard</i> atau <i>home</i> apabila data <i>login</i> valid.
Admin menginput atau edit data produk, <i>user</i> , transaksi	Memvalidasi data, jika <i>inputan</i> sesuai maka disimpan atau diedit.	Data berhasil disimpan atau diedit
Admin menghapus data produk, <i>user</i> , transaksi	Aplikasi menghapus data yang dipilih	Aplikasi memperbarui data
Admin melakukan lupa <i>password</i> akun dan menginput email akun	Aplikasi mengirimkan <i>link</i> reset <i>password</i> ke email yang di <i>inputkan</i>	Admin dapat mengatur <i>password</i> baru
Pelanggan melakukan registrasi akun dan menginputkan data akunya	Aplikasi menyimpan data akun	Pelanggan memiliki akun untuk mengakses aplikasi
Pelanggan berhasil <i>login</i>	Aplikasi membaca data dari <i>database</i>	Aplikasi menampilkan daftar produk yang tersedia
Pelanggan menginput atau edit data akun	Memvalidasi data, jika sesuai maka disimpan atau diedit datanya	Data akun berhasil diubah
Pelanggan memasukkan atau melakukan transaksi pembelian produk ke keranjang	Mengakses informasi dari produk seperti harga, detail, dan sebagainya	Pelanggan berhasil melakukan transaksi pembelian produk
Pelanggan maupun admin melakukan <i>logout</i>	Aplikasi menghapus sesi <i>login</i>	Aplikasi kembali ke halaman awal aplikasi diakses

#### 4.1.7 Analisis Biaya

Pada penelitian pembuatan aplikasi penjualan binatang ternak kali ini ada beberapa rincian biaya yang dibutuhkan dalam proses pengerjaan penelitian dan pembuatan aplikasinya, berikut diantaranya rincian yang dibutuhkannya adalah sebagai berikut :

Tabel 4.3 Analisis Biaya

Jenis Kebutuhan	Biaya
<i>Programmer</i>	Rp. 1.000.000
<i>Biaya Print</i>	Rp. 1.000.000
<i>Biaya Fotocopy</i>	Rp. 300.000
Akses Internet	Rp. 100.000
Bensin Kendaraan	Rp. 500.000
Konsumsi	Rp. 300.000
Laptop	Rp. 5.000.000
<b>Total Biaya</b>	<b>Rp. 8.200.000</b>

## 4.2 Perancangan

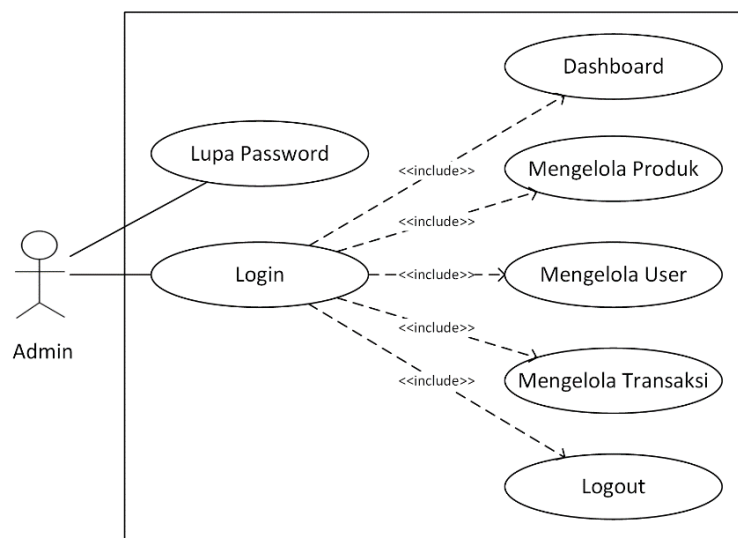
Perancangan aplikasi pada penelitian ini menggunakan *Unified Modelling Language* (UML) sebagai bahasa pemodelan yang merupakan metode pemodelan berorientasi objek. Perancangan yang dilakukan berupa pembuatan rancangan diagram serta pembuatan *mockup* untuk antarmuka aplikasinya.

### 4.2.1 Pemodelan UML

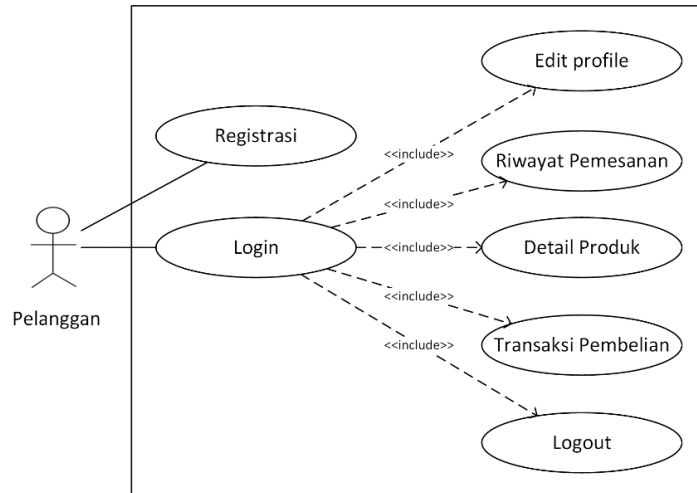
Pemodelan diagram UML yang digunakan untuk perancangan pada aplikasi adalah *use case diagram*, *activity diagram* dan *class diagram*.

#### 4.2.1.1 Use Case Diagram

Penggambaran fungsi aplikasi berdasarkan interaksi antar aktor dan objek pada sistem yang digambarkan dengan menggunakan *use case diagram*. Berikut gambar diagram *use case* nya :



Gambar 4.1 Use Case Admin



Gambar 4.2 Use Case Pelanggan

Tabel 4.4 Deskripsi Aktor Pada Aplikasi

No	Aktor	Deskripsi
1	Admin	Admin dapat mengelola data produk, <i>user</i> serta transaksi yang ada pada aplikasi.
2	Pelanggan	Pelanggan dapat melakukan transaksi pembelian produk serta dapat melakukan registrasi akun dan pengeditan akun Pelanggan mereka sendiri.

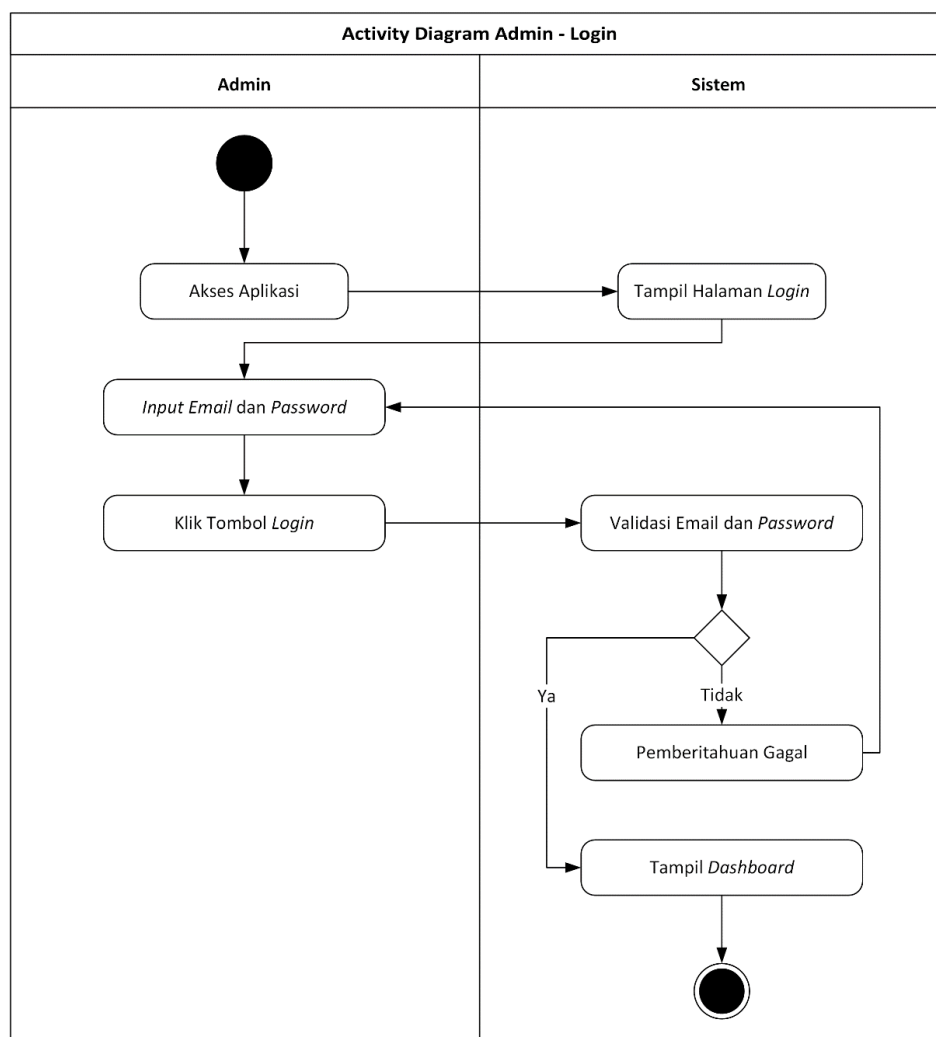
Tabel 4.5 Deskripsi Use Case Admin

No	Use Case	Deskripsi
1	<i>Login</i>	Proses melakukan <i>login</i> (masuk) ke menu aplikasi
2	Lupa Password	Untuk melakukan reset <i>password</i> akun admin berupa tautan reset <i>password</i> yang dikirimkan ke email akun yang dicantumkan
3	<i>Dashboard</i>	Menampilkan informasi mengenai data yang terjadi atau ada di aplikasi
4	Mengelola Produk	Menu aplikasi yang menampilkan data produk serta untuk melakukan pengelolaan data produk ( <i>input</i> , edit, dan hapus)
5	Mengelola User	Menu aplikasi yang menampilkan data <i>user</i> serta untuk melakukan pengelolaan data <i>user</i> (edit dan hapus)
6	Mengelola Transaksi	Menu aplikasi yang menampilkan data laboran serta melakukan pengelolaan data laboran ( <i>input</i> , edit, print dan hapus)
7	<i>Logout</i>	Proses melakukan <i>logout</i> (keluar) dari menu aplikasi

Tabel 4.6 Deskripsi *Use Case* Pelanggan

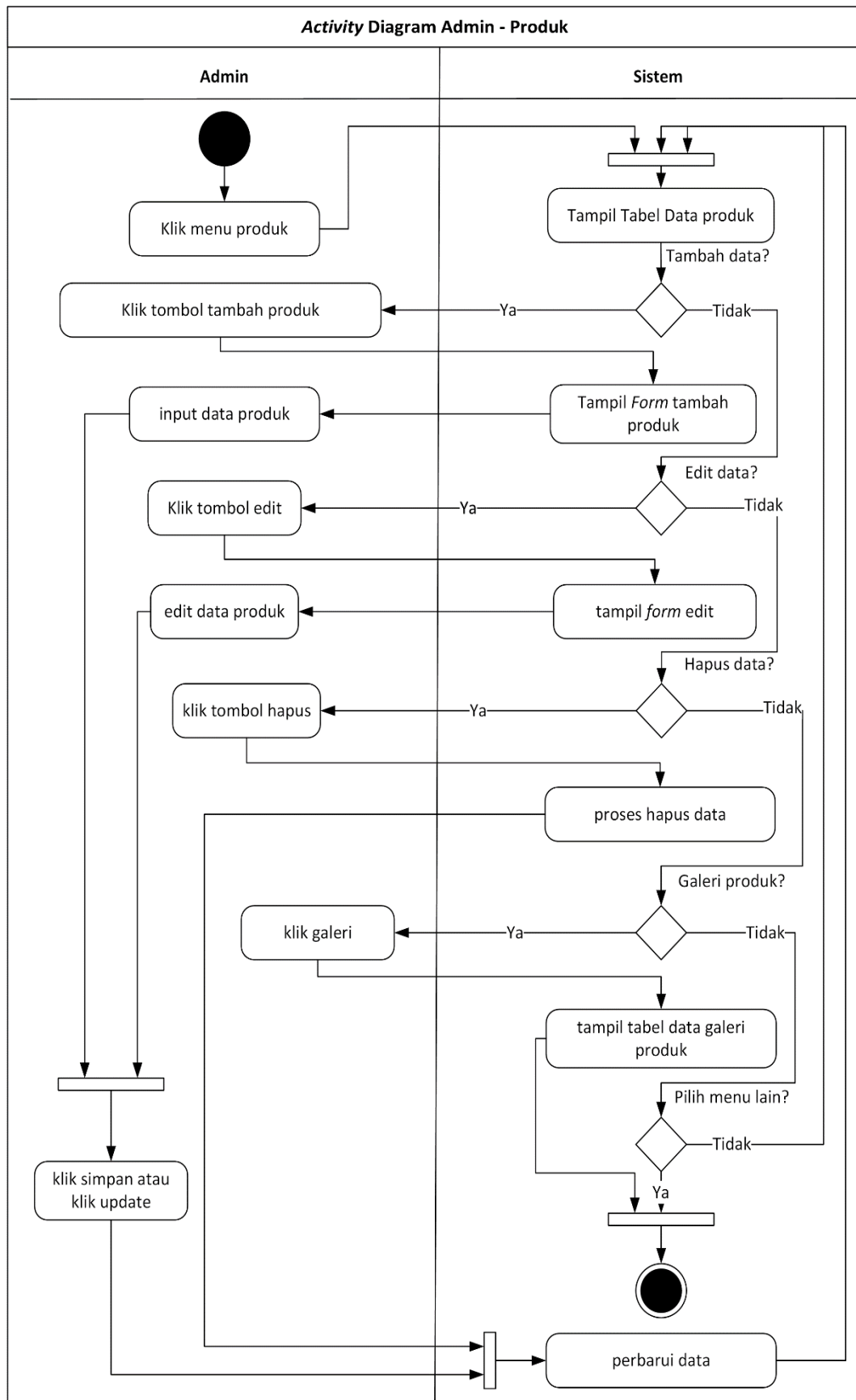
No	Use Case	Deskripsi
1	Login	Proses melakukan <i>login</i> (masuk) ke menu aplikasi
2	Registrasi	Untuk melakukan registrasi atau daftar akun Pelanggan
3	Edit Profil	Menu aplikasi untuk mengedit informasi akun Pelanggan
4	Riwayat Pemesanan	Menu aplikasi untuk melihat riwayat pemesanan produk yang dilakukan oleh Pelanggan
5	Detail Produk	Untuk melihat detail produk berupa informasi mengenai gambar, harga, dan deskripsi produk
6	Transaksi Pembelian	Berupa menu untuk melakukan transaksi pembelian produk yang ditentukan berapa banyak produk yang akan dibeli
7	Logout	Proses melakukan <i>logout</i> (keluar) dari menu aplikasi

#### 4.2.1.2 Activity Diagram

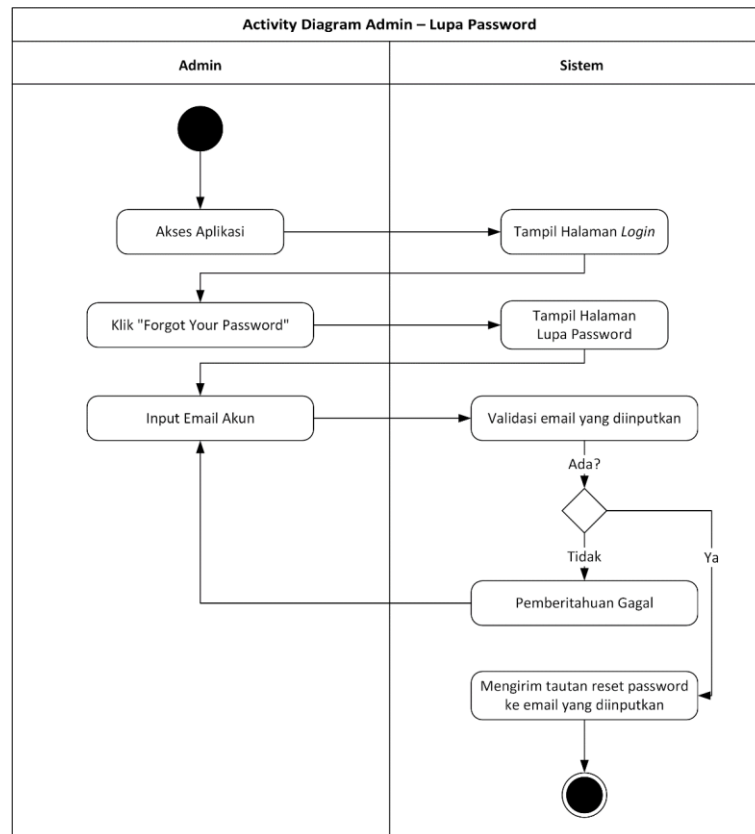


Gambar 4.3 Activity Diagram Admin - Login

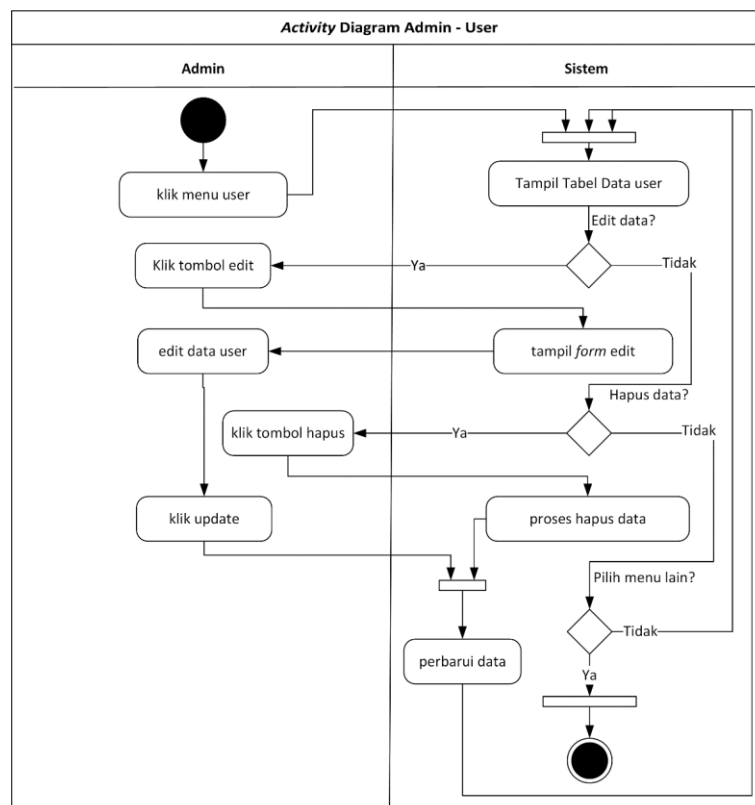




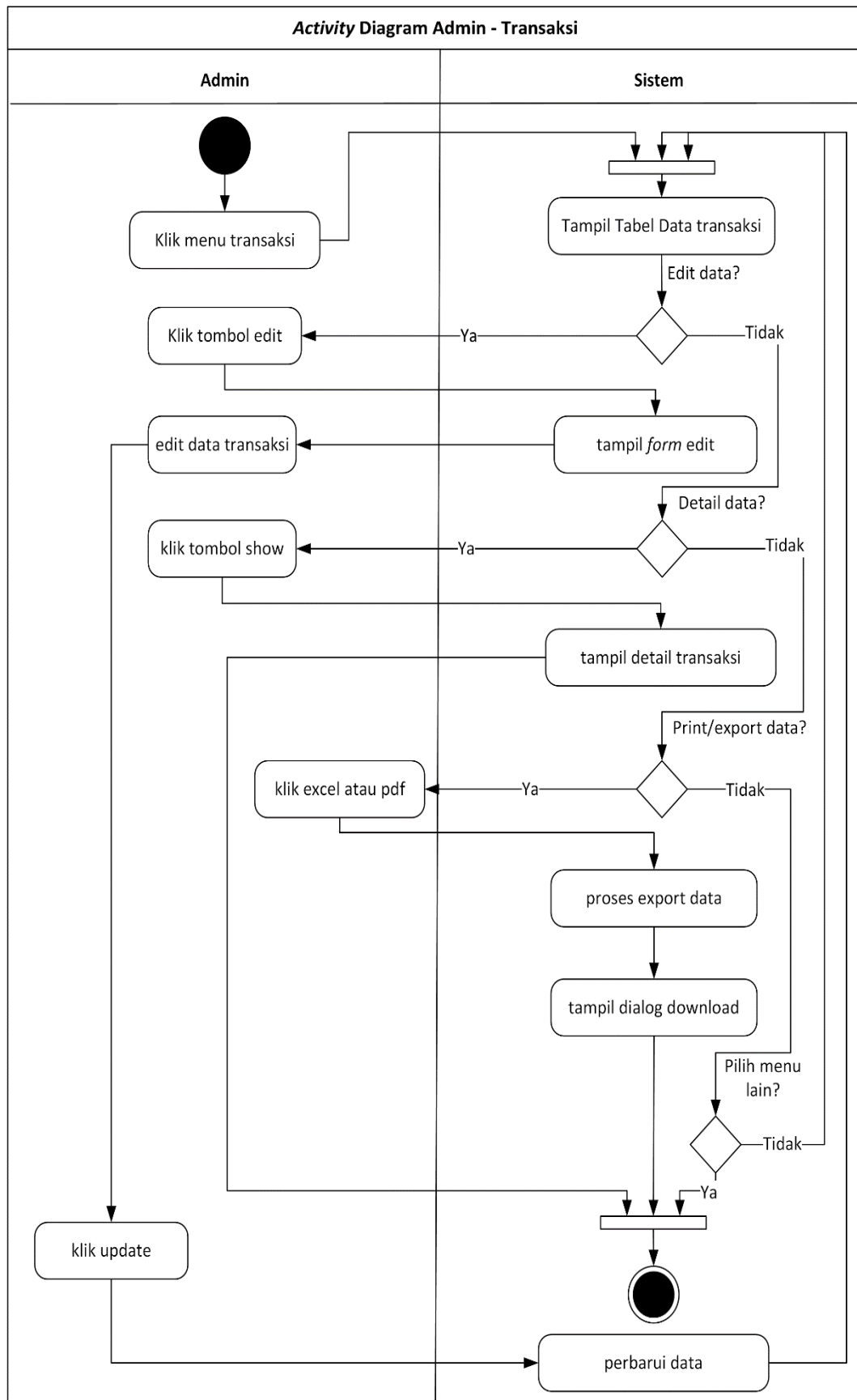
Gambar 4.4 Activity Diagram Admin - Produk



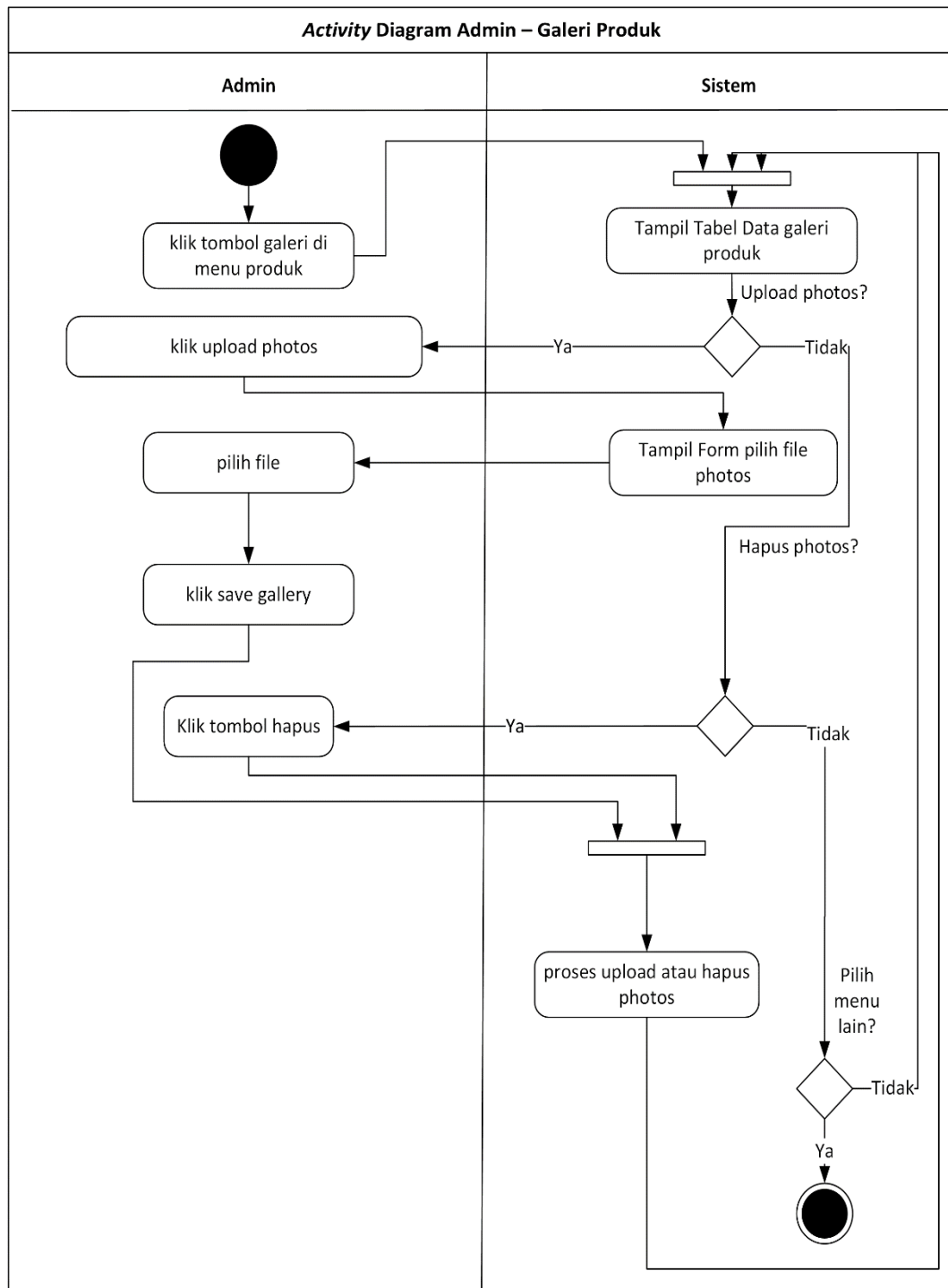
Gambar 4.5 Activity Diagram Admin – Lupa Password



Gambar 4.6 Activity Diagram Admin – User

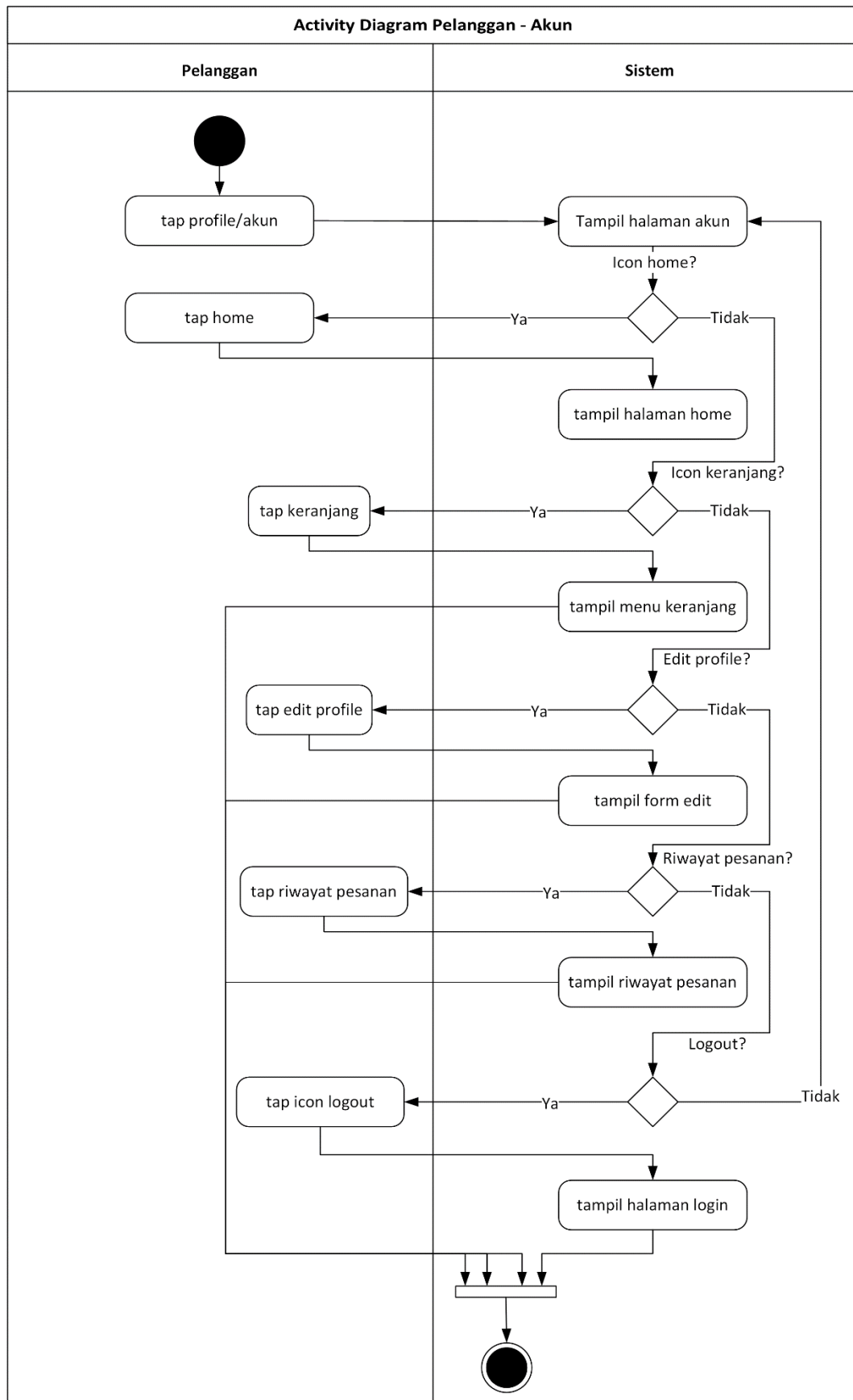


Gambar 4.7 Activity Diagram Admin - Transaksi

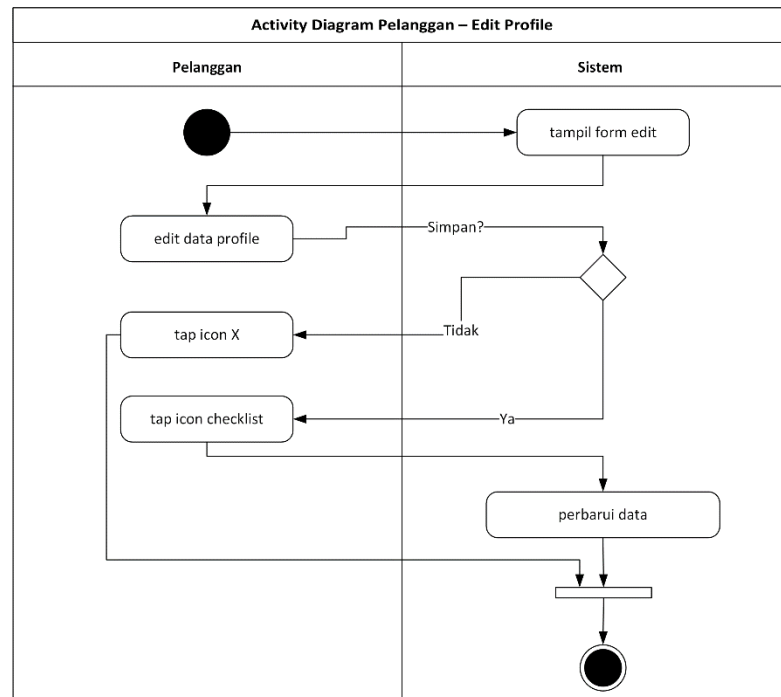


Gambar 4.8 Activity Diagram Admin – Galeri Produk

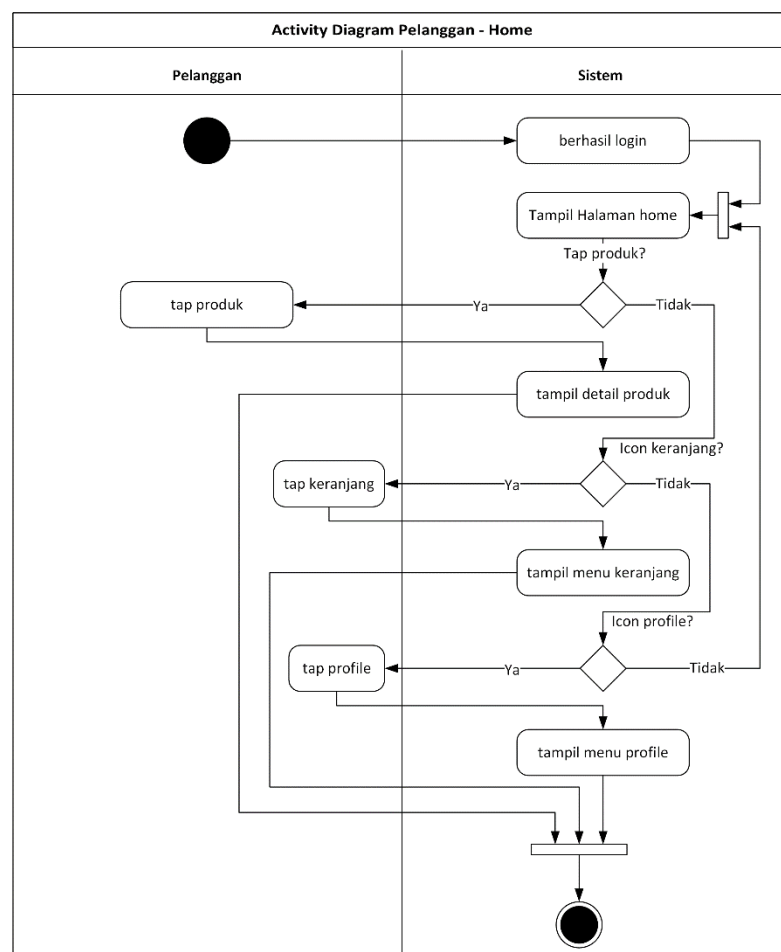
Dari *activity* diagram diatas dapat diketahui bahwa menu galeri produk ini hanya bisa diakses melalui produk yang terdaftar saja yang artinya menu ini tidaklah dapat diakses secara langsung dikarenakan diperlukannya produk yang terdaftar di menu produk. Pada menu galeri produk ini memiliki *method* tambah (*upload* gambar) dan hapus saja.



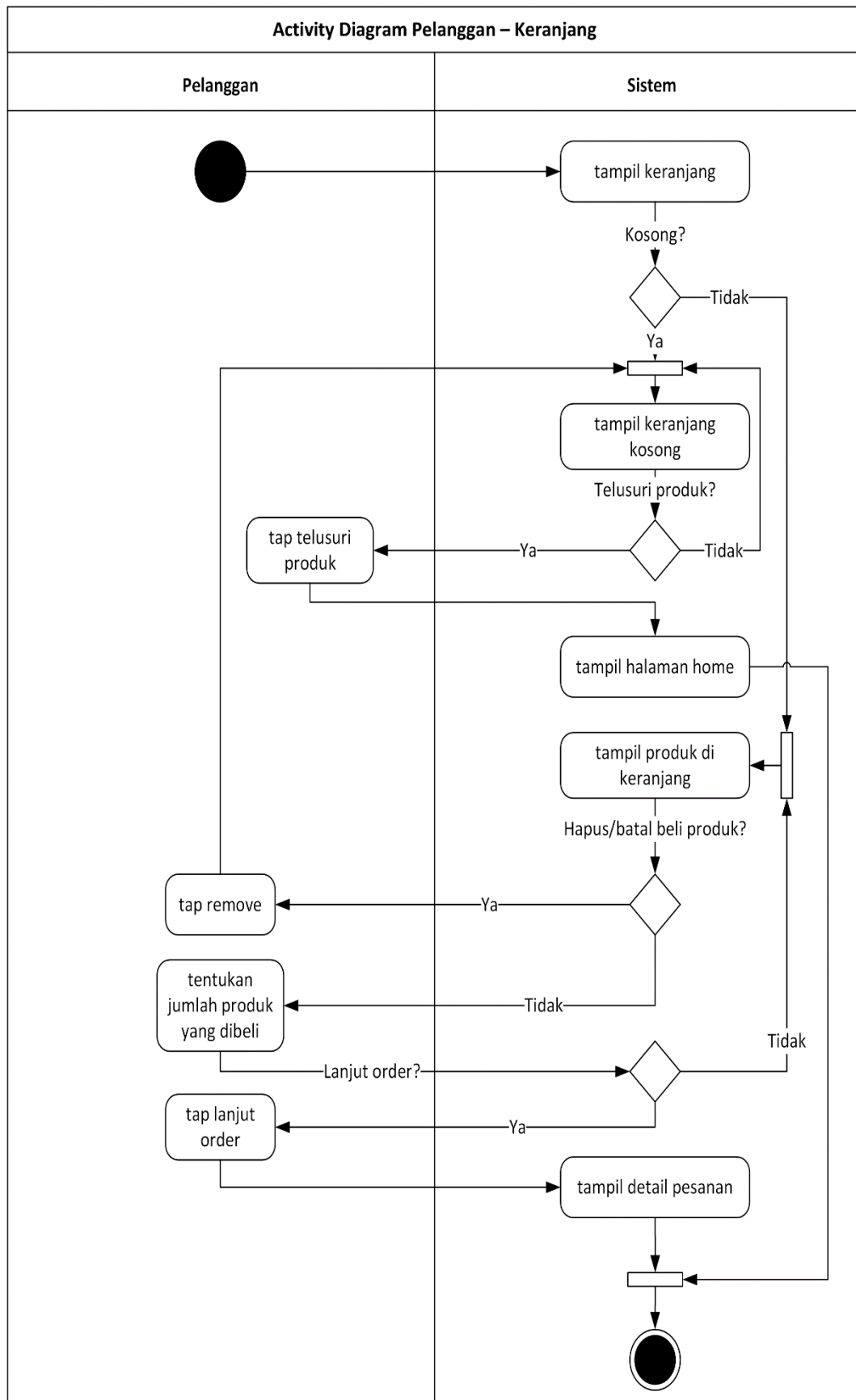
Gambar 4.9 Activity Diagram Pelanggan – Profil/Akun



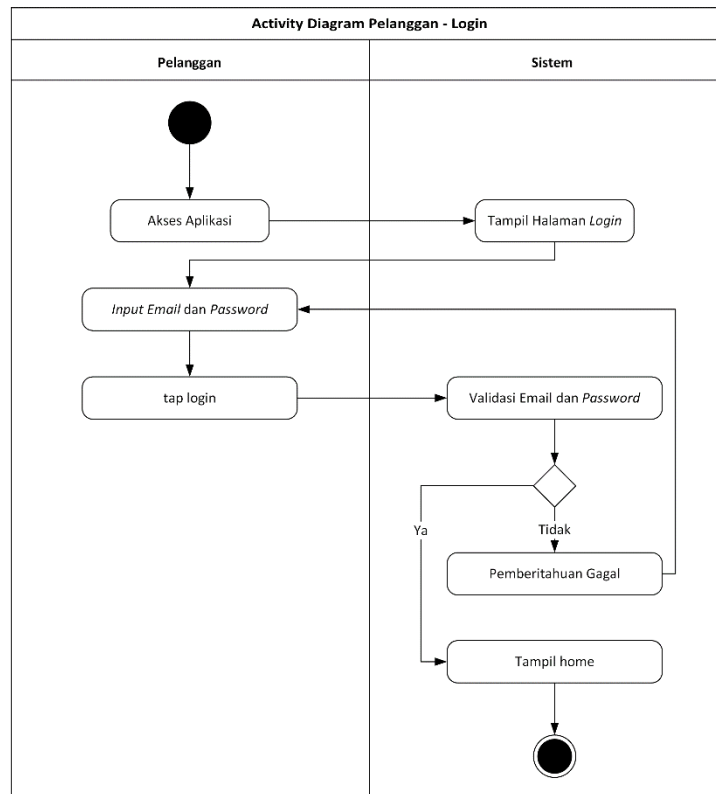
Gambar 4.10 Activity Diagram Pelanggan – Edit Profil



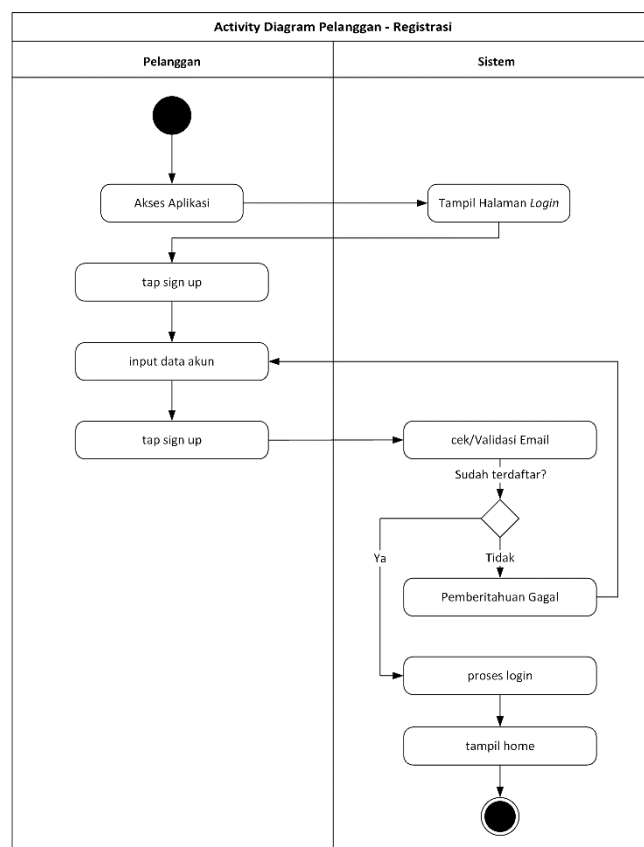
Gambar 4.11 Activity Diagram Pelanggan – Home



Gambar 4.12 Activity Diagram Pelanggan – Keranjang



Gambar 4.13 Activity Diagram Pelanggan – Login

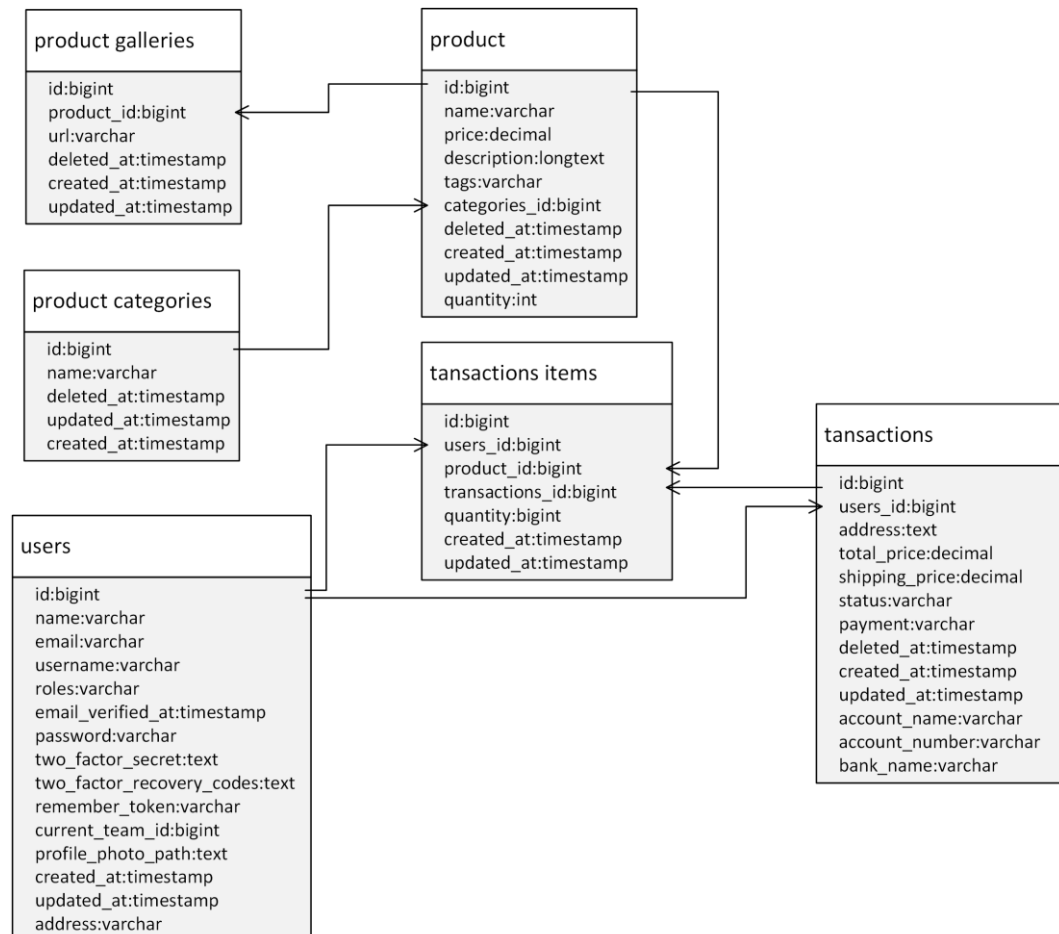


Gambar 4.14 Activity Diagram Pelanggan – Registrasi



#### 4.2.1.3 Class Diagram

Pada perancangan *class* diagram menjelaskan mengenai tabel *database* yang dibuat untuk aplikasi. Pada *class* diagram ini akan menunjukkan atribut, dan relasi penghubung pada setiap tabel yang ada.



Gambar 4.15 Class Diagram

#### 4.2.2 Struktur Tabel

Struktur tabel dilakukan untuk mengetahui struktur tabel pada basis data yang sudah dirancang pada *class* diagram sebagai media penyimpanan data. Adapun struktur rancangan basis data menunjukkan *field*, tipe, *size*, indeks dan deskripsi. Untuk rancangan basis datanya yaitu sebagai berikut:

Tabel 4.7 Struktur Tabel *Product*

Field	Tipe	Size	Indeks	Deskripsi
id	Bigint	20	PK	Id produk
name	Varchar	255		Nama produk

price	Decimal	16,0		Harga produk
description	Longtext			Deskripsi produk
tags	Varchar	255		Tak produk
categories_id	Bigint	20	FK	Id kategori
deleted_at	Timestamp			Waktu data di hapus
created_at	Timestamp			Waktu data di buat
updated_at	Timestamp			Waktu data di update
quantity	int	11		Banyak produk

Tabel 4.8 Struktur Tabel *Product Galleries*

Field	Tipe	Size	Indeks	Deskripsi
id	Bigint	20	PK	Id galeri
product__id	Bigint	20	FK	Id produk
url	varchar	255		url galeri
deleted_at	Timestamp			Waktu data di hapus
created_at	Timestamp			Waktu data di buat
updated_at	Timestamp			Waktu data di update

Tabel 4.9 Struktur Tabel *Product Categories*

Field	Tipe	Size	Indeks	Deskripsi
id	Bigint	20	PK	Id kategori
name	varchar	255		Nama kategori
deleted_at	Timestamp			Waktu data di hapus
created_at	Timestamp			Waktu data di buat
updated_at	Timestamp			Waktu data di update

Tabel 4.10 Struktur Tabel *Users*

Field	Tipe	Size	Indeks	Deskripsi
id	Bigint	20	PK	Id <i>users</i>
name	Varchar	255		Nama <i>user</i>
email	Varchar	255		Email
username	Varchar	255		Username
roles	Varchar	255		Role
email_verified_at	Timestamp			Waktu email verifikasi
password	Varchar	255		Password
two_factor_secret	Text			Pengaman akun
two_factor_recovery_codes	Text			Kode pemulihan akun

remember_token	Varchar	100		Simpan sesi token
curent_team_id	Bigint	20		Id team
profile_photo_path	Text			Lokasi foto
created_at	Timestamp			Waktu dibuat
updated_at	Timestamp			Waktu diupdate
address	varchar	255		alamat

Tabel 4.11 Struktur Tabel *Transactions*

Field	Tipe	Size	Indeks	Deskripsi
id	Bigint	20	PK	Id transaksi
users_id	Bigint	20	FK	id <i>user</i>
address	Text			Alamat
total_price	Decimal	16,0		Total harga
shipping_price	Decimal	16,0		Harga pengiriman
status	Varchar	255		Status transaksi
payment	Varchar	255		Pembayaran
deleted_at	Timestamp			Waktu dihapus
created_at	Timestamp			Waktu dibuat
updated_at	Timestamp			Waktu diupdate
account_name	Varchar	255		Nama akun
account_number	Varchar	255		Nomor akun
bank_name	varchar	255		Nama bank

Tabel 4.12 Struktur Tabel *Transaction item*

Field	Tipe	Size	Indeks	Deskripsi
id	Bigint	20	PK	Id transaksi
users_id	Bigint	20	FK	id <i>user</i>
products_id	Bigint	20	FK	Id produk
transactions_id	Bigint	20	FK	Id transaksi
quantity	Bigint	20		Banyak produk
created_at	Timestamp			Waktu dibuat
updated_at	Timestamp			Waktu diupdate

### 4.2.3 Desain

Perancangan atau desain aplikasi bertujuan untuk melakukan rancangan tampilan pada aplikasi. Sehingga saat proses implementasi aplikasinya akan lebih mudah karena hanya perlu mengikuti rancangan yang telah dibuat.

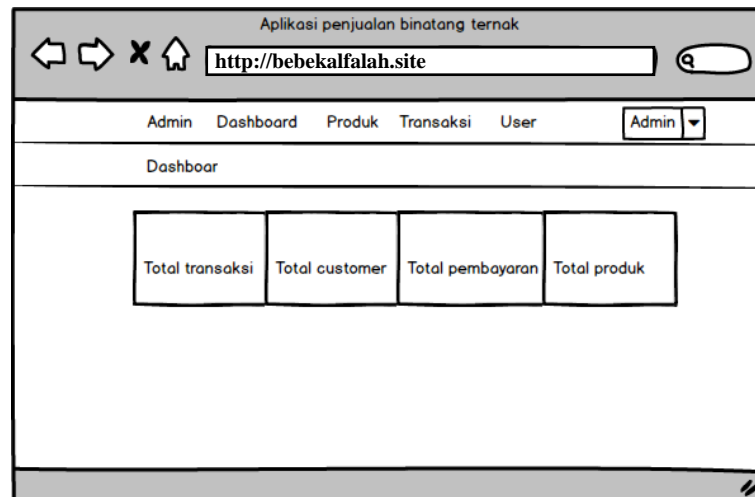
Pada tahap perancangan atau desain tampilan antarmuka aplikasi dibuat untuk memudahkan dalam tahap pengembangan yang dilakukan dengan memanfaatkan *tools* atau *software* Balsamiq Mockups. Perancangan desain tampilan antarmuka ditujukan agar tampilan aplikasi yang akan dibuat sudah terancang dengan baik. Ketika akan membuat tampilan aplikasi hanya perlu mengikuti desain yang sudah dibuat. Berikut adalah desain antarmuka aplikasinya:

The wireframe shows a web browser window titled "Aplikasi penjualan binatang ternak". The address bar contains "http://bebekalfalah.site". The main content area features a login form with two input fields labeled "Email" and "Password". Below the "Password" field is a link that says "forgot your password" and a "Login" button. The browser window includes standard navigation icons (back, forward, stop, home) and a search icon.

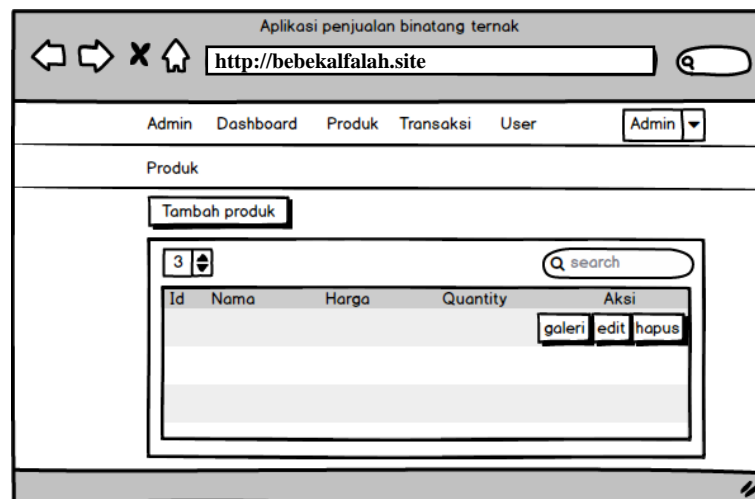
Gambar 4.16 Desain Tampilan Antarmuka Admin - *Login*

The wireframe shows a web browser window titled "Aplikasi penjualan binatang ternak". The address bar contains "http://bebekalfalah.site". The main content area features a form for password reset. It starts with a text area labeled "text penjelasan". Below this is an "Email" input field and a button labeled "Email password reset link". The browser window includes standard navigation icons (back, forward, stop, home) and a search icon.

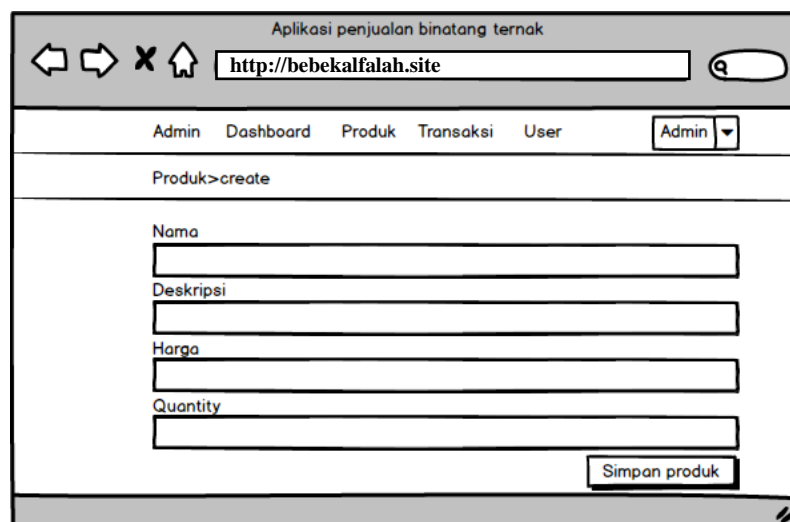
Gambar 4.17 Desain Tampilan Antarmuka Admin – *Lupa Password*



Gambar 4.18 Desain Tampilan Antarmuka Admin - *Dashboard*



Gambar 4.19 Desain Tampilan Antarmuka Admin - Produk



Gambar 4.20 Desain Tampilan Antarmuka Admin – Tambah Produk

Aplikasi penjualan binatang ternak

http://bebekalfalah.site

Admin Dashboard Produk Transaksi User Admin

Produk>edit

Nama

Deskripsi

Harga

Quantity

Update produk

Gambar 4.21 Desain Tampilan Antarmuka Admin – Edit Produk

Aplikasi penjualan binatang ternak

http://bebekalfalah.site

Admin Dashboard Produk Transaksi User Admin

Produk>galeri

Upload photos

3

search

Id	Foto	Featured	Action
			hapus

Gambar 4.22 Desain Tampilan Antarmuka Admin – Galeri Produk

Aplikasi penjualan binatang ternak

http://bebekalfalah.site

Admin Dashboard Produk Transaksi User Admin

Transaksi

Dari :

Sampai :

Excel pdf show 10 refresh

search

Id	Nama	Email	Roles	Aksi
				show edit

Gambar 4.23 Desain Tampilan Antarmuka Admin - Transaksi

Aplikasi penjualan binatang ternak

http://bebekalfalah.site

Admin Dashboard Produk Transaksi User Admin

Transaksi>Edit

Status

Update transaction

Gambar 4.24 Desain Tampilan Antarmuka Admin – Edit Transaksi

Aplikasi penjualan binatang ternak

http://bebekalfalah.site

Admin Dashboard Produk Transaksi User Admin

Transaksi

Detail transaksi

Id	
Email	
Alamat	
Pembayaran	
Total harga	
Harga pengiriman	
Status	

Gambar 4.25 Desain Tampilan Antarmuka Admin – Detail Transaksi

Aplikasi penjualan binatang ternak

http://bebekalfalah.site

Admin Dashboard Produk Transaksi User Admin

User

3

search

Id	Nama	Email	Roles	Aksi
				edit hapus

Gambar 4.26 Desain Tampilan Antarmuka Admin - Users

A screenshot of a web browser window titled 'Aplikasi penjualan binatang ternak'. The address bar shows 'http://bebekalfalah.site'. The navigation menu includes 'Admin', 'Dashboard', 'Produk', 'Transaksi', 'User', and a dropdown menu for 'Admin'. The page title is 'User>Edit'. The form contains four input fields labeled 'Name', 'Email', 'Username', and 'Roles'. An 'Update user' button is located at the bottom right of the form.

Gambar 4.27 Desain Tampilan Antarmuka Admin – Edit *Users*

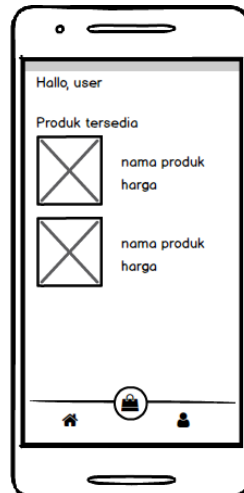
A mobile app login screen design. It features a 'Login' title at the top. Below the title are two input fields labeled 'Email Address' and 'Password'. A 'Sign in' button is positioned below the password field. At the bottom, there is a link that says 'Belum punya akun? Sign up'.

Gambar 4.28 Desain Tampilan Antarmuka Pelanggan - *Login*

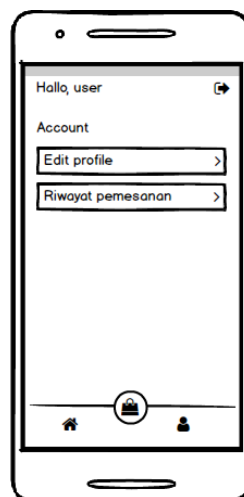
A mobile app sign-up screen design. It features a 'Sign Up' title at the top. Below the title are five input fields labeled 'Full Name', 'Username', 'Email Address', 'Password', and 'Address'. A 'Sign up' button is positioned below the address field. At the bottom, there is a link that says 'Sudah punya akun? Sign in'.

Gambar 4.29 Desain Tampilan Antarmuka Pelanggan - Registrasi

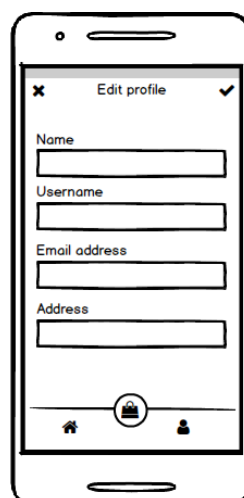




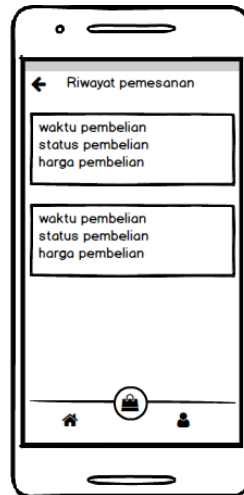
Gambar 4.30 Desain Tampilan Antarmuka Pelanggan - Home



Gambar 4.31 Desain Tampilan Antarmuka Pelanggan – Profil/Akun



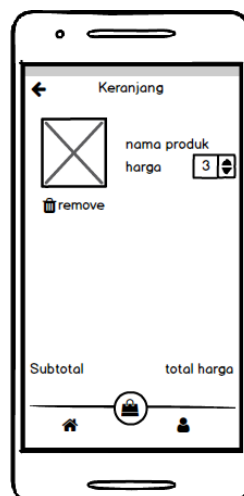
Gambar 4.32 Desain Tampilan Antarmuka Pelanggan – Edit Profil



Gambar 4.33 Desain Tampilan Antarmuka Pelanggan – Riwayat Pesanan



Gambar 4.34 Desain Tampilan Antarmuka Pelanggan – Keranjang Kosong



Gambar 4.35 Desain Tampilan Antarmuka Pelanggan – Keranjang Isi



Gambar 4.36 Desain Tampilan Antarmuka Pelanggan – Detail Produk

## BAB V

### IMPLEMENTASI DAN PENGUJIAN

#### 5.1 Implementasi

Tahap Implementasi ini dilakukan sesuai dengan perancangan yang dibuat pada bab sebelumnya. Tahap ini terdiri dari implementasi *coding* (*listing* program), tampilan aplikasi, implementasi sistem berupa tempat dan waktu pengimplementasian sistem serta spesifikasi sistem berupa perangkat keras, perangkat lunak, instalasi sistem dan menjalankan sistem.

##### 5.1.1 Listing Program

*Listing* program yaitu menampilkan *coding* program yang digunakan dalam pengembangan aplikasi, terutama *coding* solusi untuk mengatasi masalah yang diteliti. Berikut *list* program yang digunakan dalam mengatasi masalah :

###### a. Admin - Menu Produk

```
<x-app-layout>
  <x-slot name="header">
    <h2 class="font-semibold text-xl text-gray-800 leading-tight">
      {{ __( 'Product' ) }}
    </h2>
  </x-slot>

  <x-slot name="script">
    <script>
      // AJAX DataTable
      var datatable = $('#crudTable').DataTable({
        ajax: {
          url: '{!! url()->current() !!}',
        },
        columns: [
          { data: 'id', name: 'id', width: '5%' },
          { data: 'name', name: 'name' },
          { data: 'price', name: 'price' },
          { data: 'quantity', name: 'quantity' },
          {
            data: 'action',
            name: 'action',
            orderable: false,
            searchable: false,
            width: '25%'
          }
        ],
      });
    </script>
  </x-slot>
```

```

<div class="py-12">
  <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
    <div class="mb-10">
      <a href="{{ route('dashboard.product.create') }}" class="bg-green-500 hover:bg-green-700 text-white font-bold py-2 px-4 rounded shadow-lg">
        + Tambah Product
      </a>
    </div>
    <div class="shadow overflow-hidden sm:rounded-md">
      <div class="px-4 py-5 bg-white sm:p-6">
        <table id="crudTable">
          <thead>
            <tr>
              <th>ID</th>
              <th>Nama</th>
              <th>Harga</th>
              <th>Quantity</th>
              <th>Aksi</th>
            </tr>
          </thead>
          <tbody></tbody>
        </table>
      </div>
    </div>
  </div>
</div>
</x-app-layout>

```

#### b. Admin – Tambah Produk

```

<x-app-layout>
  <x-slot name="header">
    <h2 class="font-semibold text-xl text-gray-800 leading-tight">
      {!! __('Product &raquo; Create') !!}
    </h2>
  </x-slot>

  <div class="py-12">
    <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
      <div>
        @if ($errors->any())
          <div class="mb-5" role="alert">
            <div class="bg-red-500 text-white font-bold rounded-t px-4 py-2">
              There's something wrong!
            </div>
            <div class="border border-t-0 border-red-400 rounded-b bg-red-100 px-4 py-3 text-red-700">
              <p>
                <ul>
                  @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                  @endforeach
                </ul>
              </p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</x-app-layout>

```

```

        </p>
    </div>
</div>
@endif
<form class="w-full" action="{{
route('dashboard.product.store') }}" method="post"
enctype="multipart/form-data">
    @csrf
    <div class="flex flex-wrap -mx-3 mb-6">
        <div class="w-full px-3">
            <label class="block uppercase
tracking-wide text-gray-700 text-xs font-bold mb-2" for="grid-
last-name">
                Nama
            </label>
            <input value="{{ old('name') }}"
name="name" class="appearance-none block w-full bg-gray-200
text-gray-700 border border-gray-200 rounded py-3 px-4 leading-
tight focus:outline-none focus:bg-white focus:border-gray-500"
id="grid-last-name" type="text" placeholder="Product Name">
        </div>
    </div>
    <div class="flex flex-wrap -mx-3 mb-6">
        <div class="w-full px-3">
            <label class="block uppercase
tracking-wide text-gray-700 text-xs font-bold mb-2" for="grid-
last-name">
                Deskripsi
            </label>
            <textarea name="description"
class="appearance-none block w-full bg-gray-200 text-gray-700
border border-gray-200 rounded py-3 px-4 leading-tight
focus:outline-none focus:bg-white focus:border-gray-500"
id="grid-last-name" type="text" placeholder="Product
Description">{{ old('description') }}</textarea>
        </div>
    </div>
    <div class="flex flex-wrap -mx-3 mb-6">
        <div class="w-full px-3">
            <label class="block uppercase
tracking-wide text-gray-700 text-xs font-bold mb-2" for="grid-
last-name">
                Harga
            </label>
            <input value="{{ old('price') }}"
name="price" class="appearance-none block w-full bg-gray-200
text-gray-700 border border-gray-200 rounded py-3 px-4 leading-
tight focus:outline-none focus:bg-white focus:border-gray-500"
id="grid-last-name" type="number" placeholder="Product Price">
        </div>
    </div>
    <div class="flex flex-wrap -mx-3 mb-6">
        <div class="w-full px-3">
            <label class="block uppercase
tracking-wide text-gray-700 text-xs font-bold mb-2" for="grid-
last-name">
                Quantity

```

```

        </label>
        <input value="{{ old('quantity') }}"
name="quantity" class="appearance-none block w-full bg-gray-200
text-gray-700 border border-gray-200 rounded py-3 px-4 leading-
tight focus:outline-none focus:bg-white focus:border-gray-500"
id="grid-last-name" type="number" placeholder="Quantity">
    </div>
</div>
<div class="flex flex-wrap -mx-3 mb-6">
    <div class="w-full px-3 text-right">
        <button type="submit" class="
shadow-lg bg-green-500 hover:bg-green-700 text-white font-bold
py-2 px-4 rounded">
            Simpan Produk
        </button>
    </div>
</div>
</form>
</div>
</div>
</div>
</x-app-layout>

```

### c. Admin – Edit Produk

```

<x-app-layout>
    <x-slot name="header">
        <h2 class="font-semibold text-xl text-gray-800 leading-
tight">
            Product &raquo; {{ $item->name }} &raquo; Edit
        </h2>
    </x-slot>

    <div class="py-12">
        <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
            <div>
                @if ($errors->any())
                    <div class="mb-5" role="alert">
                        <div class="bg-red-500 text-white font-
bold rounded-t px-4 py-2">
                            There's something wrong!
                        </div>
                        <div class="border border-t-0 border-
red-400 rounded-b bg-red-100 px-4 py-3 text-red-700">
                            <p>
                                <ul>
                                    @foreach ($errors->all() as
$error)
                                        <li>{{ $error }}</li>
                                    @endforeach
                                </ul>
                            </p>
                        </div>
                    </div>
                @endif
                <form class="w-full" action="{{
route('dashboard.product.update', $item->id) }}" method="post"

```

```

enctype="multipart/form-data">
    @csrf
    @method('put')
    <div class="flex flex-wrap -mx-3 mb-6">
        <div class="w-full px-3">
            <label class="block uppercase
tracking-wide text-gray-700 text-xs font-bold mb-2" for="grid-
last-name">
                Nama
            </label>
            <input value="{{ old('name') }}"
$item->name }}" name="name" class="appearance-none block w-full
bg-gray-200 text-gray-700 border border-gray-200 rounded py-3
px-4 leading-tight focus:outline-none focus:bg-white
focus:border-gray-500" id="grid-last-name" type="text"
placeholder="Product Name">
        </div>
    </div>
    <div class="flex flex-wrap -mx-3 mb-6">
        <div class="w-full px-3">
            <label class="block uppercase
tracking-wide text-gray-700 text-xs font-bold mb-2" for="grid-
last-name">
                Deskripsi
            </label>
            <textarea name="description"
class="appearance-none block w-full bg-gray-200 text-gray-700
border border-gray-200 rounded py-3 px-4 leading-tight
focus:outline-none focus:bg-white focus:border-gray-500"
id="grid-last-name" type="text" placeholder="Product
Description">{{ old('description') }} $item->description
</textarea>
        </div>
    </div>

    <div class="flex flex-wrap -mx-3 mb-6">
        <div class="w-full px-3">
            <label class="block uppercase
tracking-wide text-gray-700 text-xs font-bold mb-2" for="grid-
last-name">
                Harga
            </label>
            <input value="{{ old('price') }}"
$item->price }}" name="price" class="appearance-none block w-
full bg-gray-200 text-gray-700 border border-gray-200 rounded
py-3 px-4 leading-tight focus:outline-none focus:bg-white
focus:border-gray-500" id="grid-last-name" type="number"
placeholder="Product Price">
        </div>
    </div>
    <div class="flex flex-wrap -mx-3 mb-6">
        <div class="w-full px-3">
            <label class="block uppercase
tracking-wide text-gray-700 text-xs font-bold mb-2" for="grid-
last-name">
                Quantity
            </label>

```



```

                <input value="{{ old('quantity') ??
$item->quantity }}" name="quantity" class="appearance-none block
w-full bg-gray-200 text-gray-700 border border-gray-200 rounded
py-3 px-4 leading-tight focus:outline-none focus:bg-white
focus:border-gray-500" id="grid-last-name" type="number"
placeholder="Quantity">
            </div>
        </div>
        <div class="flex flex-wrap -mx-3 mb-6">
            <div class="w-full px-3 text-right">
                <button type="submit" class="bg-
green-500 hover:bg-green-700 text-white font-bold py-2 px-4
rounded">
                    Update Produk
                </button>
            </div>
        </div>
    </form>
</div>
</div>
</div>
</x-app-layout>

```

d. Pelanggan – Home atau Produk Tersedia

```

import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:flutter_application_1/models/user_model.dart';
import
'package:flutter_application_1/providers/auth_provider.dart';
import
'package:flutter_application_1/providers/product_provider.dart';
import 'package:flutter_application_1/theme.dart';
import
'package:flutter_application_1/widgets/product_card.dart';
import
'package:flutter_application_1/widgets/product_tile.dart';

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    AuthProvider authProvider =
Provider.of<AuthProvider>(context);
    UserModel user = authProvider.user;
    ProductProvider productProvider =
Provider.of<ProductProvider>(context);

    Widget header() {
      return Container(
        margin: EdgeInsets.only(
          top: defaultMargin,
          left: defaultMargin,
          right: defaultMargin,
        ),
        child: Row(
          children: [
            Expanded(

```

```

        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              'Hallo, ${user.name}',
              style: primaryTextStyle.copyWith(
                fontSize: 24,
                fontWeight: semiBold,
              ),
            ),
            Text(
              '@${user.username}',
              style: subtitleTextStyle.copyWith(
                fontSize: 16,
              ),
            ),
          ],
        ),
      ),
      Container(
        width: 54,
        height: 54,
        decoration: BoxDecoration(
          shape: BoxShape.circle,
          image: DecorationImage(
            image: NetworkImage(
              user.profilePhotoUrl,
            ),
          ),
        ),
      ),
    ),
  ],
),
);
}

Widget categories() {
  return Container(
    margin: EdgeInsets.only(
      top: defaultMargin,
    ),
    child: SingleChildScrollView(
      scrollDirection: Axis.horizontal,
      child: Row(
        children: [
          SizedBox(
            width: defaultMargin,
          ),
          Container(
            padding: EdgeInsets.symmetric(
              horizontal: 12,
              vertical: 10,
            ),
          ),
          margin: EdgeInsets.only(right: 16),
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(12),
            color: primaryColor,

```

```

    ),
    child: Text(
      'All Shoes',
      style: primaryTextStyle.copyWith(
        fontSize: 13,
        fontWeight: medium,
      ),
    ),
  ),
),
Container(
  padding: EdgeInsets.symmetric(
    horizontal: 12,
    vertical: 10,
  ),
),
margin: EdgeInsets.only(right: 16),
decoration: BoxDecoration(
  borderRadius: BorderRadius.circular(12),
  border: Border.all(
    color: subtitleColor,
  ),
  color: transparentColor,
),
child: Text(
  'Running',
  style: subtitleTextStyle.copyWith(
    fontSize: 13,
    fontWeight: medium,
  ),
),
),
),
Container(
  padding: EdgeInsets.symmetric(
    horizontal: 12,
    vertical: 10,
  ),
),
margin: EdgeInsets.only(right: 16),
decoration: BoxDecoration(
  borderRadius: BorderRadius.circular(12),
  border: Border.all(
    color: subtitleColor,
  ),
  color: transparentColor,
),
child: Text(
  'Training',
  style: subtitleTextStyle.copyWith(
    fontSize: 13,
    fontWeight: medium,
  ),
),
),
),
Container(
  padding: EdgeInsets.symmetric(
    horizontal: 12,
    vertical: 10,
  ),
),
margin: EdgeInsets.only(right: 16),

```

```

        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(12),
          border: Border.all(
            color: subtitleColor,
          ),
          color: transparentColor,
        ),
        child: Text(
          'Basketball',
          style: subtitleTextStyle.copyWith(
            fontSize: 13,
            fontWeight: medium,
          ),
        ),
      ),
    ),
    Container(
      padding: EdgeInsets.symmetric(
        horizontal: 12,
        vertical: 10,
      ),
      margin: EdgeInsets.only(right: 16),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(12),
        border: Border.all(
          color: subtitleColor,
        ),
        color: transparentColor,
      ),
      child: Text(
        'Hiking',
        style: subtitleTextStyle.copyWith(
          fontSize: 13,
          fontWeight: medium,
        ),
      ),
    ),
  ],
),
);
}

Widget popularProductsTitle() {
  return Container(
    margin: EdgeInsets.only(
      top: defaultMargin,
      left: defaultMargin,
      right: defaultMargin,
    ),
    child: Text(
      'Popular Products',
      style: primaryTextStyle.copyWith(
        fontSize: 22,
        fontWeight: semiBold,
      ),
    ),
  );
};

```

```

    }

    Widget popularProducts() {
      return Container(
        margin: EdgeInsets.only(top: 14),
        child: SingleChildScrollView(
          scrollDirection: Axis.horizontal,
          child: Row(
            children: [
              SizedBox(
                width: defaultMargin,
              ),
              Row(
                children: productProvider.products
                  .map(
                    (product) => ProductCard(product),
                  )
                  .toList(),
              ),
            ],
          ),
        ),
      );
    }

    Widget newArrivalsTitle() {
      return Container(
        margin: EdgeInsets.only(
          top: defaultMargin,
          left: defaultMargin,
          right: defaultMargin,
        ),
        child: Text(
          'Produk Tersedia',
          style: primaryTextStyle.copyWith(
            fontSize: 22,
            fontWeight: semiBold,
          ),
        ),
      );
    }

    Widget newArrivals() {
      return Container(
        margin: EdgeInsets.only(
          top: 14,
        ),
        child: Column(
          children: productProvider.products
            .map(
              (product) => ProductTile(product),
            )
            .toList(),
        ),
      );
    }
  }

```

```

    return ListView(
      children: [
        header(),
        // categories(),
        // popularProductsTitle(),
        // popularProducts(),
        newArrivalsTitle(),
        newArrivals(),
      ],
    );
  }
}

```

e. Pelanggan – Detail Produk

```

import 'package:carousel_slider/carousel_slider.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:flutter_application_1/models/product_model.dart';
// import
'package:flutter_application_1/pages/detail_chat_page.dart';
import
'package:flutter_application_1/providers/cart_provider.dart';
import
'package:flutter_application_1/providers/wishlist_provider.dart';
import 'package:flutter_application_1/theme.dart';

class ProductPage extends StatefulWidget {
  final ProductModel product;
  ProductPage(this.product);

  @override
  _ProductPageState createState() => _ProductPageState();
}

class _ProductPageState extends State<ProductPage> {
  List images = [
    'assets/image_shoes.png',
    'assets/image_shoes.png',
    'assets/image_shoes.png',
  ];

  List familiarShoes = [
    'assets/image_shoes.png',
    'assets/image_shoes2.png',
    'assets/image_shoes3.png',
    'assets/image_shoes4.png',
    'assets/image_shoes5.png',
    'assets/image_shoes6.png',
    'assets/image_shoes7.png',
    'assets/image_shoes8.png',
  ];

  int currentIndex = 0;

  @override
  Widget build(BuildContext context) {

```

```

WishlistProvider wishlistProvider =
Provider.of<WishlistProvider>(context);
CartProvider cartProvider =
Provider.of<CartProvider>(context);

Future<void> showSuccessDialog() async {
  return showDialog(
    context: context,
    builder: (BuildContext context) => Container(
      width: MediaQuery.of(context).size.width - (2 *
defaultMargin),
      child: AlertDialog(
        backgroundColor: backgroundColor3,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(30),
        ),
        content: SingleChildScrollView(
          child: Column(
            children: [
              Align(
                alignment: Alignment.centerLeft,
                child: GestureDetector(
                  onTap: () {
                    Navigator.pop(context);
                  },
                  child: Icon(
                    Icons.close,
                    color: primaryTextColor,
                  ),
                ),
              ),
              Image.asset(
                'assets/icon_success.png',
                width: 100,
              ),
              SizedBox(
                height: 12,
              ),
              Text(
                'Hurray :)',
                style: primaryTextStyle.copyWith(
                  fontSize: 18,
                  fontWeight: semiBold,
                ),
              ),
              SizedBox(
                height: 12,
              ),
              Text(
                'Produk dimasukkan kedalam keranjang',
                style: secondaryTextStyle,
              ),
              SizedBox(
                height: 20,
              ),
              Container(
                width: 154,

```

```

        height: 44,
        child: TextButton(
          onPressed: () {
            Navigator.pushNamed(context, '/cart');
          },
          style: TextButton.styleFrom(
            backgroundColor: primaryColor,
            shape: RoundedRectangleBorder(
              borderRadius:
BorderRadius.circular(12),
            ),
          ),
          child: Text(
            'Lihat Keranjang',
            style: primaryTextStyle.copyWith(
              fontSize: 16,
              fontWeight: medium,
            ),
          ),
        ),
      ),
    ],
  ),
),
);
}

Widget indicator(int index) {
  return Container(
    width: currentIndex == index ? 16 : 4,
    height: 4,
    margin: EdgeInsets.symmetric(
      horizontal: 2,
    ),
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      color: currentIndex == index ? primaryColor :
Color(0xffC4C4C4),
    ),
  );
}

Widget familiarShoesCard(String imageUrl) {
  return Container(
    width: 54,
    height: 54,
    margin: EdgeInsets.only(
      right: 16,
    ),
    decoration: BoxDecoration(
      image: DecorationImage(
        image: AssetImage(imageUrl),
      ),
      borderRadius: BorderRadius.circular(6),
    ),
  ),
}

```



```

    );
  }

Widget header() {
  int index = -1;

  return Column(
    children: [
      Container(
        margin: EdgeInsets.only(
          top: 20,
          left: defaultMargin,
          right: defaultMargin,
        ),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            GestureDetector(
              onTap: () {
                Navigator.pop(context);
              },
              child: Icon(
                Icons.chevron_left,
              ),
            ),
            Icon(
              Icons.shopping_bag,
              color: backgroundColor1,
            ),
          ],
        ),
      ),
      CarouselSlider(
        items: widget.product.galleries
          .map(
            (image) => Image.network(
              image.url,
              width: MediaQuery.of(context).size.width,
              height: 310,
              fit: BoxFit.cover,
            ),
          )
          .toList(),
        options: CarouselOptions(
          initialPage: 0,
          onPageChanged: (index, reason) {
            setState(() {
              currentIndex = index;
            });
          },
        ),
      ),
      SizedBox(
        height: 20,
      ),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,

```

```

        children: widget.product.galleries.map((e) {
          index++;
          return indicator(index);
        }).toList(),
      ),
    ],
  );
}

Widget content() {
  int index = -1;

  return Container(
    height: MediaQuery.of(context).size.height,
    width: double.infinity,
    margin: EdgeInsets.only(top: 17),
    decoration: BoxDecoration(
      borderRadius: BorderRadius.vertical(
        top: Radius.circular(24),
      ),
      color: backgroundColor1,
    ),
    child: Column(
      children: [
        // NOTE: HEADER
        Container(
          margin: EdgeInsets.only(
            top: defaultMargin,
            left: defaultMargin,
            right: defaultMargin,
          ),
          child: Row(
            children: [
              Expanded(
                child: Column(
                  crossAxisAlignment:
CrossAxisAlignment.start,
                  children: [
                    Text(
                      widget.product.name,
                      style: primaryTextStyle.copyWith(
                        fontSize: 18,
                        fontWeight: semiBold,
                      ),
                    ),
                    Text(
                      "",
                      style: secondaryTextStyle.copyWith(
                        fontSize: 12,
                      ),
                    ),
                  ],
                ),
              ),
            ],
          ),
        ],
      ),
    ),
  ),

```

```

// NOTE: PRICE
Container(
  width: double.infinity,
  margin: EdgeInsets.only(
    top: 20,
    left: defaultMargin,
    right: defaultMargin,
  ),
  padding: EdgeInsets.all(16),
  decoration: BoxDecoration(
    color: backgroundColor2,
    borderRadius: BorderRadius.circular(4),
  ),
  child: Row(
    mainAxisAlignment:
MainAxisAlignment.spaceBetween,
    children: [
      Text(
        'Harga',
        style: primaryTextStyle,
      ),
      Text(
        'Rp. ${widget.product.price}',
        style: priceTextStyle.copyWith(
          fontSize: 16,
          fontWeight: semiBold,
        ),
      ),
    ],
  ),
),

// NOTE: DESCRIPTION
Container(
  width: double.infinity,
  margin: EdgeInsets.only(
    top: defaultMargin,
    left: defaultMargin,
    right: defaultMargin,
  ),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'Deskripsi',
        style: primaryTextStyle.copyWith(
          fontWeight: medium,
        ),
      ),
      SizedBox(
        height: 12,
      ),
      Text(
        widget.product.description,
        style: subtitleTextStyle.copyWith(
          fontWeight: light,

```

```

        ),
        textAlign: TextAlign.justify,
      ),
    ],
  ),
),

// NOTE: BUTTONS
Container(
  width: double.infinity,
  margin: EdgeInsets.all(defaultMargin),
  child: Row(
    children: [
      Expanded(
        child: Container(
          height: 54,
          child: TextButton(
            onPressed: () {
              cartProvider.addCart(widget.product);
              showSuccessDialog();
            },
            style: TextButton.styleFrom(
              shape: RoundedRectangleBorder(
                borderRadius:
BorderRadius.circular(12),
              ),
              backgroundColor: primaryColor,
            ),
            child: Text(
              'Masukkan ke keranjang',
              style: primaryTextStyle.copyWith(
                fontSize: 16,
                fontWeight: semiBold,
              ),
            ),
          ),
        ),
      ),
    ],
  ),
),
);

return Scaffold(
  backgroundColor: backgroundColor6,
  body: ListView(
    children: [
      header(),
      content(),
    ],
  ),
);
}
}

```

### 5.1.2 Implementasi Sistem

Implementasi sistem merupakan tahap penerapan sistem yang akan dilakukan jika sistem telah disetujui termasuk program yang sesuai berdasarkan tahap perancangan yang sudah dibuat pada tahap sebelumnya. Adapun waktu dan tempat untuk implementasi sistem yaitu :

Tempat : Peternakan Mang Asetp  
 Alamat : Kp. Sukalilah RT 03 RW 14 Desa Padaulun Kec. Majalaya  
 Waktu : Agustus 2022

### 5.1.3 Spesifikasi Sistem

Spesifikasi sistem merupakan informasi mengenai spesifikasi untuk perangkat keras dan perangkat lunak yang digunakan untuk menjalankan sistem aplikasi. Selain itu, pada bagian ini juga dijelaskan mengenai instalasi sistemnya dan menjalankan sistemnya.

#### a. Spesifikasi Perangkat Keras

Tabel 5.1 Spesifikasi Perangkat Keras

<i>Processor</i>	AMD A4-9120 RADEON R3, 4 CORES 2C+2G 2,2 Ghz
RAM	4 GB
HDD	512 GB

#### b. Spesifikasi Perangkat Lunak

Tabel 5.2 Spesifikasi Perangkat Lunak

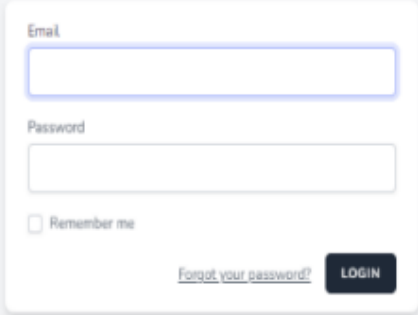
Browser	Google Chrome
<i>Database</i>	MySQL
<i>Framework-Web</i>	Laravel
<i>Framework-Android</i>	Flutter

#### c. Instalasi Sistem

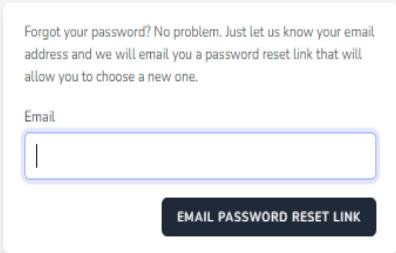
Instalasi sistem yaitu menginstal *software* yang digunakan untuk menjalankan sistem aplikasi. Browser merupakan sebuah aplikasi mesin pencari dan digunakan menjalankan aplikasi yang penulis kembangkan. Tentunya berbagai aplikasi browser manapun bisa untuk menjalankan aplikasi ini namun dalam penelitian ini menggunakan browser Google Chrome. Berikut langkah-langkah instalasi Google Chrome :

- 1) Unduh installer Google Chrome yang bisa didapatkan di website google chrome.
  - 2) Jika sudah terunduh, lakukan *double* klik untuk menginstal, lalu klik *next*
  - 3) Tunggu hingga proses instalasi selesai
- d. Menjalankan Sistem

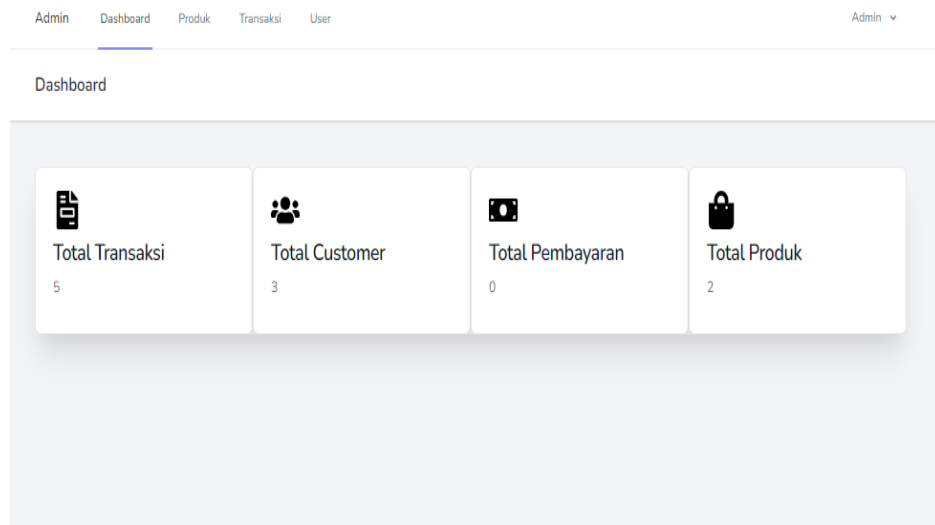
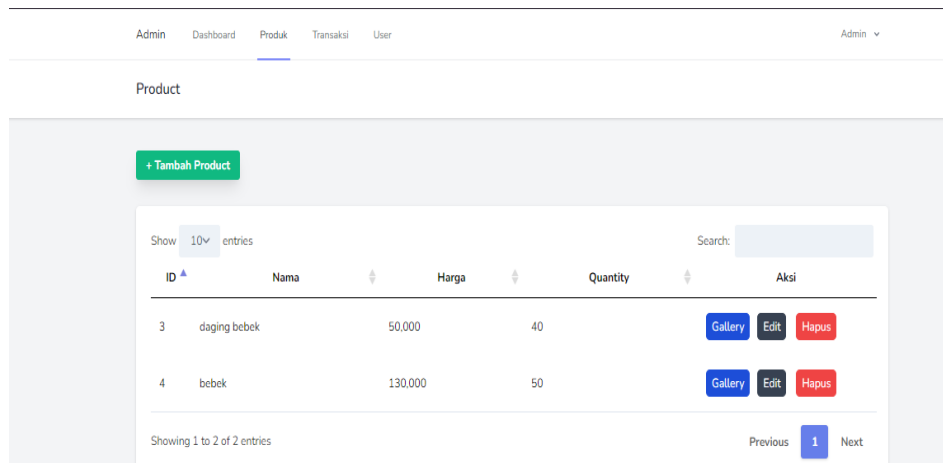
Menjalankan sistem yaitu berisi mengenai seperti apa jalannya sistem atau aplikasinya. Berikut gambar hasil dari menjalankan sistemnya:



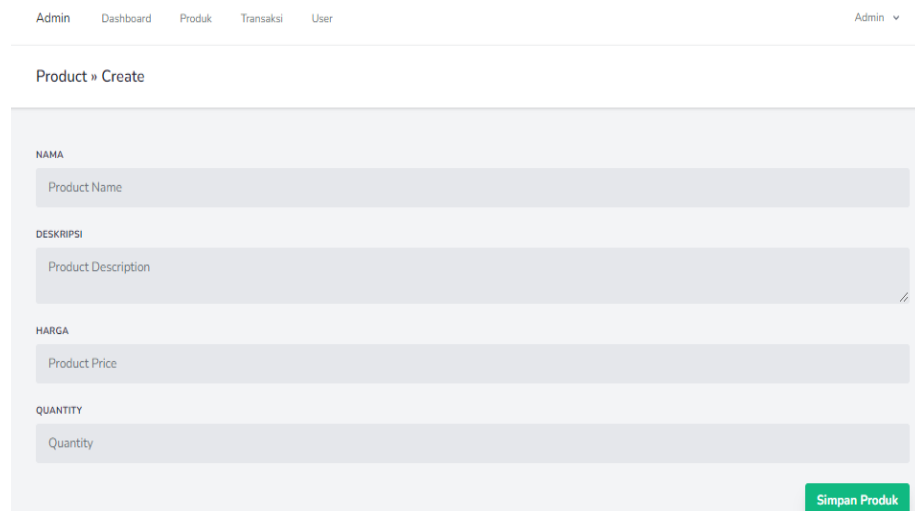
Gambar 5.1 Halaman Admin - *Login*



Gambar 5.2 Halaman Admin – Lupa *Password*

Gambar 5.3 Halaman Admin - *Dashboard*

Gambar 5.4 Halaman Admin - Produk



Gambar 5.5 Halaman Admin – Tambah Produk

Admin Dashboard Produk Transaksi User Admin

Product » daging bebek » Edit

NAMA  
daging bebek

DESKRIPSI  
bebek

HARGA  
50000

QUANTITY  
40

Update Produk


Gambar 5.6 Halaman Admin – Edit Produk

Admin Dashboard Produk Transaksi User Admin

Product » daging bebek » Gallery

+ Upload Photos

Show 10▼ entries Search:

ID	Photo	Featured	Action
9		No	Hapus

Showing 1 to 1 of 1 entries Previous 1 Next

Gambar 5.7 Halaman Admin – Detail Produk

Admin Dashboard Produk Transaksi User Admin

User

Show 10▼ entries Search:

ID	Nama	Email	Roles	Aksi
1	Admin	admin@test.com	ADMIN	Edit Hapus
5	alfa	alfa@gmail.com	USER	Edit Hapus
6	wahyu@gmail.com	wahyu@gmail.com	USER	Edit Hapus
7	Tina Lestari	atehtina024@gmail.com	USER	Edit Hapus

Showing 1 to 4 of 4 entries Previous 1 Next

Gambar 5.8 Halaman Admin – Users



Admin Dashboard Produk Transaksi User Admin ▾

User » Admin » Edit

NAME  
Admin

EMAIL  
admin@test.com

USERNAME  
Username

ROLES  
ADMIN ▾

Update User

Gambar 5.9 Halaman Admin – Edit User

Admin Dashboard Produk Transaksi User Admin ▾

Transaction

Dari :

Sampai :

Excel PDF Show 10 rows Refresh

Search:

ID ▲	Tanggal	Nama	Total Harga	Status	Aksi
1	27 Aug 2022 07:08:33	alfa	50,000	PENDING	Show Edit
2	27 Aug 2022 07:18:20	alfa	130,000	PENDING	Show Edit
3	28 Aug 2022 02:44:16	Admin	100,000	PENDING	Show Edit
4	28 Aug 2022 02:50:06	Admin	100,000	PENDING	Show Edit
5	28 Aug 2022 08:05:50	Admin	50,000	PENDING	Show Edit

Showing 1 to 5 of 5 entries

Previous 1 Next

Gambar 5.10 Halaman Admin – Transaksi

Admin Dashboard Produk Transaksi User Admin ▾

Transaction » Edit

STATUS  
PENDING ▾

Update Transaction

Gambar 5.11 Halaman Admin – Transaksi Edit

Admin	Dashboard	Produk	Transaksi	User	Admin ▾
-------	-----------	--------	-----------	------	---------

Transaction » #1	
Detail Transaksi	
Name	alfa
Email	alfa@gmail.com
Alamat	Bandung
Pembayaran	MANUAL
Total Harga	50,000
Harga Pengiriman	0
Status	PENDING
Nama Akun	Alfa
No. Rekening	1212121

Gambar 5.12 Halaman Admin – Detail Transaksi

### Login

Sign In to Continue

Email Address

Password

Sign In

Gambar 5.13 Halaman Pelanggan – Login

### Sign Up

Register and Happy Shopping

Full Name

Username

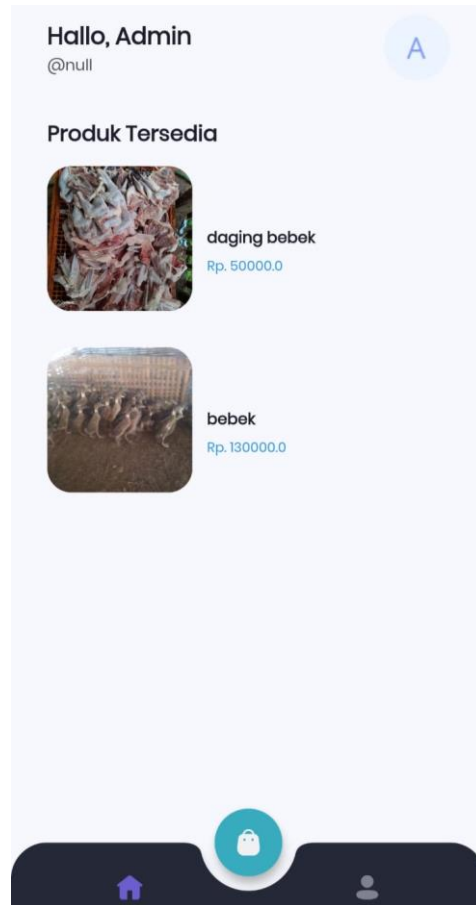
Email Address

Password

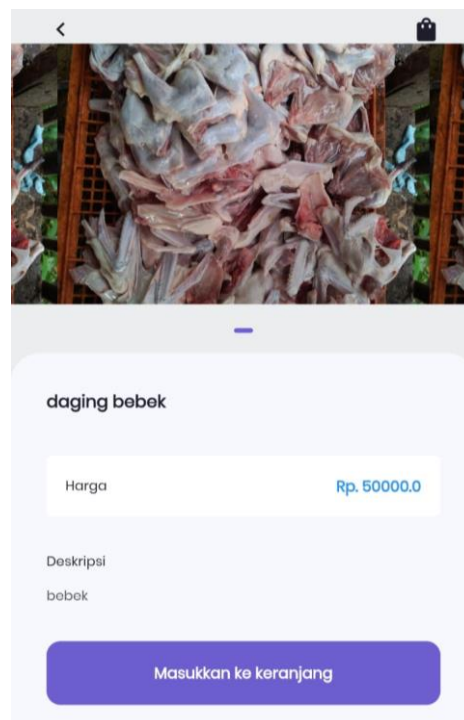
Address

Sign Up

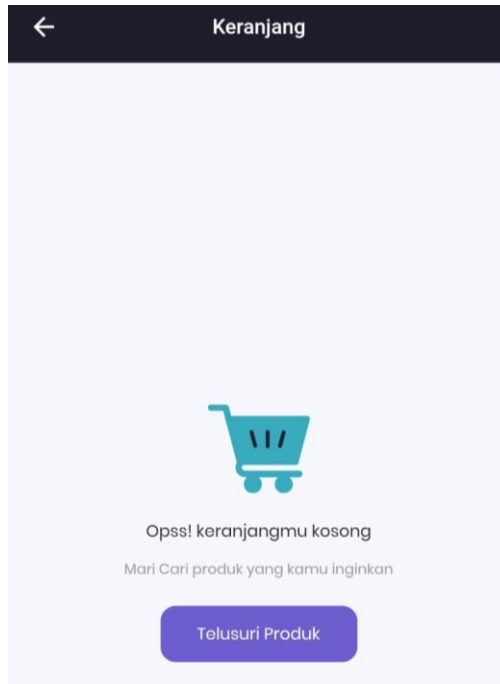
Gambar 5.14 Halaman Pelanggan – Registrasi



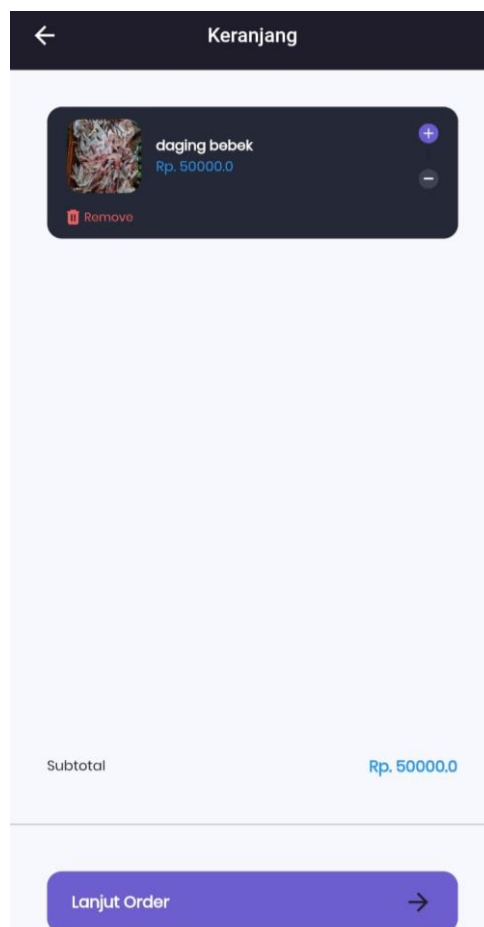
Gambar 5.15 Halaman Pelanggan – Home



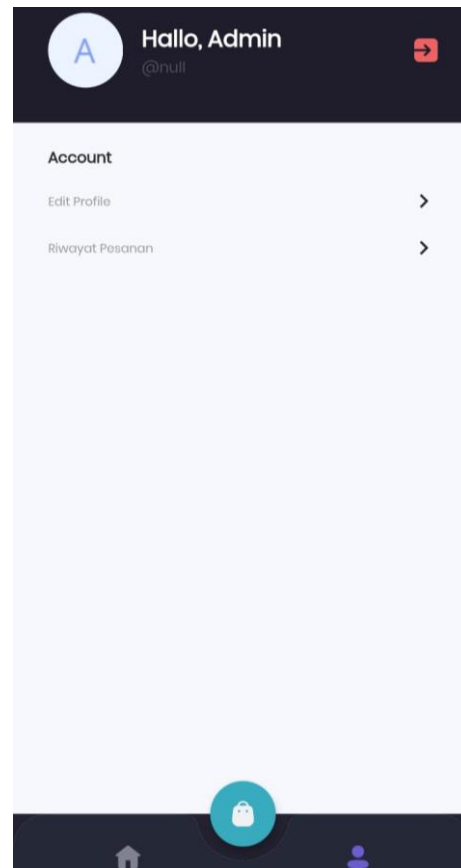
Gambar 5.16 Halaman Pelanggan – Detail Produk



Gambar 5.17 Halaman Pelanggan – Keranjang Kosong

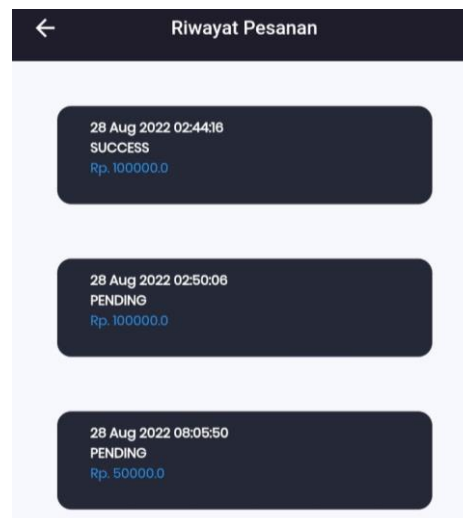


Gambar 5.18 Halaman Pelanggan – Keranjang Isi



Gambar 5.19 Halaman Pelanggan – Profil/Akun

Gambar 5.20 Halaman Pelanggan – Edit Profil



Gambar 5.21 Halaman Pelanggan – Riwayat Pesanan

## 5.2 Pengujian

Tahap pengujian merupakan tahap untuk menguji coba aplikasi. Uji coba dilakukan oleh aktor Admin dan Pelanggan sesuai dengan perancangan. Pengujian menggunakan metode Black Box Testing. Berikut tabel pengujian:

Tabel 5.3 Pengujian

No	Halaman yang diuji	Aksi Aktor	Reaksi Sistem		Hasil
			Benar	Salah	
A	Admin				
1	Login	Memasukkan email dan password	Masuk ke halaman dashboard	Menampilkan pesan tidak ada akun dari data login yang dimasukkan	Sesuai
2	Logout	Pilih logout	Akan melakukan logout atau keluar dari sistem	Tidak melakukan logout	Sesuai
3	Data Produk	Menambah data produk	Data produk akan bertambah	Menampilkan pesan kesalahan jika ada data wajib diisi tetapi tidak diisi	Sesuai
		Mengedit data produk	Data produk terbaru akan tersimpan	Menampilkan pesan kesalahan jika ada data wajib diisi tetapi tidak diisi	Sesuai

		Mengelola produk galeri	Melakukan perubahan ( <i>upload</i> , <i>hapus</i> ) pada produk galeri akan otomatis tersimpan	Data produk galeri tidak diperbarui	Sesuai
		Menghapus data produk	Data produk berhasil dihapus	Data produk tidak terhapus	Sesuai
4	Data <i>User</i>	Mengedit data <i>user</i>	Data <i>user</i> terbaru akan tersimpan	Menampilkan pesan kesalahan jika data yang di edit tidak sesuai <i>form</i>	Sesuai
		Menghapus data <i>user</i>	Data <i>user</i> berhasil dihapus	Data <i>user</i> tidak terhapus	Sesuai
5	Data Transaksi	Unduh laporan transaksi (Excel / Pdf)	Muncul dialog unduh data laporan transaksi	Tidak muncul dialog unduh data laporan transaksi	Sesuai
		Edit status transaksi	Status transaksi berhasil diubah	Status transaksi tidak berubah	Sesuai
<b>B</b>	<b>Pelanggan</b>				
1	<i>Login</i>	Memasukkan email dan <i>password</i>	Masuk ke halaman home	Menampilkan pesan tidak ada akun dari data <i>login</i> yang dimasukkan	Sesuai
2	<i>Logout</i>	Pilih <i>logout</i>	Akan melakukan <i>logout</i> atau keluar dari sistem	Tidak melakukan <i>logout</i>	Sesuai
3	Transaksi Produk	Memasukkan produk ke keranjang	Produk berhasil masuk ke keranjang	Produk tidak masuk ke keranjang	Sesuai
		Melakukan pemesanan produk	Produk berhasil di pesan dan masuk ke riwayat pesanan	Produk tidak berhasil di pesan atau produk tidak masuk ke riwayat pesanan	Sesuai
		Melihat detail produk	Menampilkan detail produk	Tidak dapat menampilkan detail produk	Sesuai
4	Data Akun atau Profil	Mengedit profil	Data profil akun berhasil diubah	Data profil akun tidak berubah	Sesuai

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Dari hasil penelitian yang telah dilaksanakan, berikut adalah kesimpulan yang dapat dihasilkan.

1. Aplikasi Penjualan binatang ternak berbasis android ini berhasil dibangun dengan menggunakan PHP, laravel, flutter, HTML, Tailwind CSS. Memudahkan dalam proses penjualan barang karena bisa dipesan dari jarak jauh. Kemudian memudahkan dalam proses transaksi penjualannya.
2. Aplikasi penjualan peternakan bebek Alfalah ini memudahkan dalam pembuatan data laporan penjualan yang diperlukan sebagai bukti transaksi. Sehingga dengan cara tersebut cukup praktis dalam pencatatan penjualan.
3. Aplikasi penjualan peternakan bebek Alfalah ini bisa menjadi media pros mosi kepada publik karena memudahkan untuk diakses oleh semua kalangan. Memudahkan untuk disurvei produk harga dan barang yang dijual dan alamat penjualan jika publik ingin menyurvei secara langsung kepada tempat peternakan secara langsung. Memudahkan juga untuk memberitahukan kepada publik jika ada perubahan harga dari sebelumnya mahal menjadi lebih murah sehingga bisa menjadi strategi dalam penjualan barang kepada masyarakat luas.

#### **6.2 Saran**

Dari hasil penelitian yang telah dilaksanakan peneliti masih banyak kekurangan dalam satu dan hal yang lain, maka peneliti merokomendasikan atau menyarankan beberapa hal, yaitu:

1. Dalam Aplikasi penjualan binatang ternak ini masih banyak kekurangan terutama dalam hal tampilan dikarenakan dalam pembuatan aplikasi ini peneliti memfokuskan terlebih dahulu kepada fungsi dari aplikasi, maka kedepanya bisa di kembangkan lagi dalam hal tampilan aplikasi.
2. Aplikasi penjualan binatang ternak berbasis android ini bisa ditambahkan fitur notifikasi untuk mempermudah pelanggan melihat status pesanan.



3. Untuk Kedepanya Web admin bisa di tambahkan fitur statistic untuk mempermudah melihat data pada laporan transaksi.
4. Aplikasi bisa di tambahkan metode pembayaran lain agar opsi pelanggan punya banyak opsi dalam membayar.
5. Jika Aplikasi ini akan digunakan secara menyeluruh kepada konsumen atau pembeli di sarankan menggunakan server dengan bandwidth yang lebih besar agar ketika di akses tidak mengalami lemot.

## DAFTAR PUSTAKA

- Ahmad, R. F., & Hasti, N. (2018). Sistem Informasi Penjualan Sandal Berbasis Web. *Jurnal Teknologi dan Informasi*, 8(1), 67–72. <https://doi.org/10.34010/jati.v8i1.911>
- Aldiansyah, H., Widipangestu, I., & Supriyadi, D. (2022). Perancangan Sistem Informasi Penjualan Fashion Pria Berbasis Web Dengan Menggunakan Metode Extreme Programming Pada Toko Aa Distro Pamulang (Studi Kasus Pada Toko Aa Distro Pamulang). *Jurnal ESIT (E-Bisnis, Sistem Informasi, Teknologi Informasi)*, 17(1), Article 1. <http://openjournal.unpam.ac.id/index.php/ESIT/article/view/19802>
- Andi, J. (2015). Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A-GPS ) Dengan Platform Android. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 1(1), 1–8.
- Anggraini, Y., Pasha, D., Damayanti, D., & Setiawan, A. (2020). Sistem Informasi Penjualan Sepeda Berbasis Web Menggunakan Framework Codeigniter. *Jurnal Teknologi Dan Sistem Informasi*, 1(2), 64–70. <https://doi.org/10.33365/jtsi.v1i2.236>
- Arfianto, F. R., & Nugrahanti, F. (2018). Rancang Bangun Aplikasi Penjualan Perumahan Berbasis Web Pada Cv. Grand Permata Residence Magetan. *Seminar Nasional Teknologi Informasi dan Komunikasi*, 1(2017), 1.
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 3(2), 207. <https://doi.org/10.30998/string.v3i2.3048>
- Harumy, T. H. F., Sitorus, J., & Lubis, M. (2018). Sistem Informasi Absensi Pada PT. Cospar Sentosa Jaya Menggunakan Bahasa Pemrograman Java. *Jurnal Teknik Dan Informatika*, 5(1), 63–70.
- Hermanto, B. (2019). Sistem Informasi Manajemen Keuangan Pada PT. Hulu Balang Mandiri Menggunakan Framework Laravel. *Jurnal Komputasi*, 7(1), 17–26.
- Huda, B., & Apriyanto, S. (2019). Aplikasi Sistem Informasi Lowongan Pekerjaan Berbasis Android dan Web Monitoring. *Buana Ilmu*, 4(1), 11–24.
- Huda, B., & Priyatna, B. (2019). Penggunaan Aplikasi Content Management System (CMS) Untuk Pengembangan Bisnis Berbasis E-commerce. *Systematics*, 1(2), 81. <https://doi.org/10.35706/sys.v1i2.2076>

- Kusniyati, H., & Pangondian Sitanggang, N. S. (2016). Aplikasi Edukasi Budaya Toba Samosir Berbasis Android. *Jurnal Teknik Informatika*, 9(1), 9–18. <https://doi.org/10.15408/jti.v9i1.5573>
- Listiyah, A. (2022). *Uji Usability pada Institutional Repository Perpustakaan UIN Maulana Malik Ibrahim Malang dengan metode System Usability Scale (SUS) dan Discovery Prototyping* [Undergraduate, Universitas Islam Negeri Maulana Malik Ibrahim]. <http://etheses.uin-malang.ac.id/36831/>
- Mundzir, M. (2018). *Buku Sakti Pemrograman Web Seri PHP*. Anak Hebat Indonesia.
- Novendri. (2019). Pengertian Web. *Lentera Dumai*, 10(2), 46–57.
- Nurfitriana, E., Apriliah, W., Ferliyanti, H., Basri, H., & Ratnawati, R. (2020). Implementasi Model Waterfall Dalam Sistem Informasi Akuntansi Piutang Jasa Penyewaan Kendaraan Pada PT. TRICIPTA SWADAYA KARAWANG. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 15(1), 34–43. <https://doi.org/10.35969/interkom.v15i1.66>
- Shinta, N. E. (2021). *Pengembangan Aplikasi Blog Menggunakan Flutter dan Laravel*.
- Syabania, R., & Rosmawarni, N. (2021). Perancangan Aplikasi Costumer Relationship Management (CRM) Pada Penjualan Barang Pre-Order Berbasis Webstie | Jurnal Rekayasa Informasi. *Jurnal Rekayasa Informasi*, 10(1). <https://ejournal.istn.ac.id/index.php/rekayasainformasi/article/view/951>
- Tabrani, M., & Aghniya, I. R. (2019). Implementasi Metode Waterfall Pada Program Simpan Pinjam Koperasi Subur Jaya Mandiri Subang. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 14(1), 41–50. <https://doi.org/10.35969/interkom.v14i1.46>

## LAMPIRAN

### Lampiran 1 : Hasil Wawancara

#### HASIL WAWANCARA


Nama Narasumber : Asep Ahsan  
Tanggal : 12 Mei 2022  
Lokasi : Kp. Sukalilah RT 03 RW 14 Desa Padaulun Kec. Majalaya  
Jabatan : Pemilik Usaha Peternakan

Wawancara ini berfungsi sebagai salah satu cara pengumpulan data yang diperlukan dalam penelitian yang berjudul “Pembuatan aplikasi penjualan binatang ternak berbasis android”.

Berikut pertanyaan dan jawaban mengenai pengembangan aplikasi penjualan ternak bebek berbasis android yaitu :

Pertanyaan	Jawaban
Di perternakan ini hewan apa saja yang di perjual belikan dan data penjualan nya apa saja?	Data bebek, data penjualan, Data pembeli
Apakah sebelum ini sudah ada apikasi untuk penjualan ternak di perternakan ini?	Untuk saat ini di perternakan ini belum ada aplikasi untuk penjualan ternak di sini
Jumlah ternak bebek yang ada di perternakan ini perhari nya dijual berapa banyak?	Biasanya kami menjual bebek disini perhari nya itu kurang lebih ada 50 atau 60 ekor perhari
Berapakan penghasilan penjualan perhari?	Biasanya nyampe delapan ratus atau satu juta perhari
Siapa saja pengguna aplikasi penjualan perternakan?	Pengguna aplikasi yaitu admin,pemilik perternakan

Pemilik Perternakan



Asep Ahsan