

**LAPORAN PRAKTIK KERJA LAPANG (PKL)**

**CV SARANA UTAMA SOLUSINDO**

**SISTEM JUAL BELI SAYUR *ONLINE* BERBASIS ANDROID  
MENGGUNAKAN *WEB SERVICE***



**Disusun oleh:**

Kadek Dwi Aryasa 15515020111124

Gusti Agung Suastika adinata 155150201111273

Nigel Erico Pangestu

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2018**

# **PENGESAHAN**

## **LAPORAN PRAKTIK KERJA LAPANGAN (PKL)**

Diajukan Untuk Memenuhi Sebagai Persyaratan Kurikulum  
Program Studi Teknik Informatika  
Jurusan Teknik Informatika  
Universitas Brawijaya

### **SISTEM JUAL BELI SAYUR *ONLINE* BERBASIS ANDROID MENGGUNAKAN *WEB SERVICE***

Dilaksanakan di :  
CV SARANA UTAMA SOLUSINDO  
Lantai 2 Jalan Anggrek garuda No. 49, kota malang  
Tanggal :  
1 Juli 2018 – 30 Agustus 2018

Disusun Oleh :  
Kadek Dwi Aryasa 155150201111124  
Gusti Agung Suastika Adinata 155150201111273

Telah diperiksa dan disetujui oleh :

Mengetahui,	Menyetujui,
Kepala Jurusan Teknik Informatika	Dosen Pembimbing PKL

Tri Astoto Kurniawan, S.T, M.T, Ph.D. Komang Candra Brata, S.Kom., M.T., M.Sc.  
NIP: 19710518 200312 1 001 NIP: 201607 890711 1 000

## **LEMBAR PENGESAHAN**

### **LAPORAN PRAKTIK KERJA LAPANGAN (PKL)**

Diajukan Untuk Memenuhi Sebagai Persyaratan Kurikulum

Program Studi Teknik Informatika

Jurusan Teknik Informatika

Universitas Brawijaya

SISTEM JUAL BELI SAYUR *ONLINE BERBASIS ANDROID*

MENGGUNAKAN *WEB SERVICE*

Dilaksanakan di :

CV SARANA UTAMA SOLUSINDO

Lantai 2 Jalan Anggrek garuda No. 49, kota malang

Tanggal :

1 Juli 2018 – 30 Agustus 2018

Disusun Oleh :

Kadek Dwi Aryasa 155150201111124

Gusti Agung Suastika Adinata 155150201111273

Nigel Erico Pangestu

Telah diperiksa dan disetujui oleh :

Pihak CV. SARANA UTAMA SOLUSINDO

Menyetujui,

Menyetujui,

General Manager

Pembimbing Lapangan

Sony Winarto

Selvina Yustriasanti

## KATA PENGANTAR

*Assalamu'alaikum Wr. Wb.*

Puji syukur kehadirat Allah SWT yang telah memberikan rahmad, taufik dan hidayahnya kepada ilahi sehingga kami dapat menyelesaikan laporan praktek kerja lapang ini dengan baik. Adapun laporan praktek kerja lapang(PKL) yang kami buat ini yaitu berjudul "*Sistem Jual Beli Sayur Online Berbasis Android Menggunakan Web Service*".

Sejak awal sampai dengan akhir penulisan ini, tidak sedikit bantuan yang kami terima dan karenanya dalam kesempatan ini kami ingin menyampaikan terima kasih kepada:

1. Allah S.W.T. atas berkah dan hidayah serta bimbingan-Nya dalam pelaksanaan kuliah kerja nyata.
2. Ibu, Bapak dan seluruh keluarga atas segenap dukungan yang diberikan.
3. Bapak Ir. Heru Nurwasito, M.Kom. selaku Pembantu Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Agus Wahyu Widodo, S.T, M.Cs. selaku Ketua Program Studi Informatika / Ilmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya
6. Bapak Komang Candra Brata, S.Kom., M.T., M.Sc. selaku dosen pembimbing yang telah menyediakan waktu untuk berdiskusi dan berbagai saran serta masukkannya.
7. Selvina Yustriasanti selaku pembimbing praktik kerja lapang yang telah memberikan pengalaman, pengarahan dan saran dalam penyusunan laporan serta menyediakan kami kesempatan dalam PKL di CV. Sarana Utama Solusindo.

8. Teman-teman satu perjuangan Praktik Kerja Lapangan (PKL) di CV. Sarana Utama Solusindo yang meluangkan segenap waktu untuk saling berdiskusi.
9. Teman-teman satu angkatan Program Studi Informatika 2015 tercinta yang selalu memberikan informasi, semangat, dorongan dan bantuan pikiran.
10. Segenap civitas akademika Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya.

Demi kesempurnaan praktik kerja ini, penulis mengharapkan kritik dan saran yang bersifat membangun. Dan harapan kami, laporan Praktek Kerja Lapang (PKL) ini dapat bermanfaat bagi semua pihak.

*Wassalamu'alaikum Wr. Wb.*

Malang, 21 November 2018

Penulis

## ABSTRAK

Sebagai manusia kita melakukan kegiatan jual beli untuk memenuhi kebutuhan sehari-hari kita seperti kebutuhan akan bahan pangan (makanan dan minuman). Di dalamnya terjadi interaksi dan proses persetujuan untuk melakukan sebuah transaksi. Jika dahulu kita melakukan proses transaksi dengan tatap muka, maka sekarang dengan adanya teknologi kita dapat melakukan transaksi dengan online atau melewati telepon genggam kita saja. Dengan berkembangnya aplikasi *smartphone* dan berkembangnya jenis transaksi kita dapat mempermudah interaksi antara pembeli dan pedagang. *Android* dapat menjawab perkembangan pengembangan aplikasi *smartphone* masa sekarang karena bersifat *open source* sehingga akan mudah untuk dikembangkan. Pengembangan aplikasi *android* dalam sistem jual beli sayur, dapat mempermudah pembeli dan penjual untuk melakukan transaksi. Pengembangan aplikasi dengan android mempermudah pengembangan secara berlanjut karena menggunakan pemrograman berbasis objek. Dengan adanya sistem *web service* yang disematkan dalam aplikasi tersebut dapat membuat aplikasi berjalan secara *online* dapat dapat diakses kapan saja dan dimana saja. Untuk bagian aplikasi android menggunakan IDE (*Integrated Development Environment*) *android studio* untuk pengembangannya karena memiliki *library* dan forum yang dapat membantu pengembang. Dengan pengujian *black-box* (kotak hitam) pada fitur-fitur yang telah dibuat mendapatkan hasil jika aplikasi android yang dikembangkan dapat berjalan dengan baik dan tanpa masalah. *Android* sebagai *platform open souce* dapat berkembang dan memiliki *library* yang cukup untuk mempermudah proses pengembangan sistem kedepannya.

**Kata Kunci :** *android*, aplikasi *android*, *android studio*, *web services* , sayur, transaksi, jual beli.

## DAFTAR ISI

PENGESAHAN.....	ii
LEMBAR PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL .....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN .....	xviii
BAB 1 PENDAHULUAN.....	19
1.1    Latar belakang .....	19
1.2    Rumusan masalah.....	20
1.3    Tujuan .....	20
1.4    Manfaat.....	20
1.5    Batasan masalah.....	20
1.6    Sistematika pembahasan .....	21
1.7    Pelaksanaan Praktek Kerja Lapang .....	22
BAB 2 PROFIL PERUSAHAAN.....	23
2.1    Sejarah dan Perkembangan CV Sarana Utama Solusindo (SASINDO) .....	23
2.2    Visi, Misi dan Komitmen Perusahaan. ....	23
2.2.1    Visi.....	23
2.2.2    Misi .....	23
2.2.3    Komitmen.....	24
2.3    Struktur Organisasi CV SARANA UTAMA SOLUSINDO .....	24
2.4    Ruang Lingkup CV SARANA UTAMA SOLUSINDO .....	24
2.4.1    Solusi Bisnis TI (IT Business Solution) dan Konsultan IT (IT Consultant) .....	25
2.4.2    Pendidikan dan Pelatihan (IT Training) .....	26
BAB 3 LANDASAN KEPUSTAKAAN .....	27
3.1    Sistem Penjualan .....	27
3.1.2    Sistem Jual Beli <i>Online</i> .....	27

3.2 <i>Android</i> .....	28
3.2.1 <i>Android Studio</i> .....	30
3.3 <i>Web Service</i> .....	31
3.4 <i>Laravel</i> .....	32
3.4.1 <i>Model-View-Controller (MVC)</i> .....	32
3.5 <i>Unified Modeling Language (UML)</i> .....	33
3.5.1 <i>Structured Diagram</i> .....	34
3.5.2 <i>Behavior Diagram</i> .....	36
3.6 <i>Entity Relationship Diagram</i> .....	40
BAB 4 BAB 4 METODOLOGI .....	42
4.1 Penentuan Objek Penelitian.....	42
4.2 Studi Literatur.....	43
4.3 Analisa Kasus .....	43
4.4 Perancangan Konsep dan Model.....	43
4.5 Implementasi.....	44
4.6 Pengujian dan Analisis .....	44
4.7 Kesimpulan.....	44
BAB 5 ANALISIS DAN PERANCANGAN .....	45
5.1 Analisa Kebutuhan Sistem.....	46
5.1.1 Deskripsi Umum Sistem.....	46
5.1.2 Deskripsi Umum Perangkat Lunak .....	47
5.1.3 Fungsi Perangkat Lunak.....	47
5.1.4 Karakteristik Pengguna.....	47
5.2 Analisa Kebutuhan Perangkat Lunak .....	48
5.2.1 Kebutuhan Fungsional dan Non-Fungsional.....	49
5.2.2 Pemodelan Kebutuhan.....	51
5.3 Perancangan Perangkat Lunak .....	98
5.3.1 Perancangan Algoritma .....	98
5.3.2 Perancangan Antarmuka.....	103
5.3.3.1 Perancangan Tampilan Halaman <i>Login</i> .....	103
5.3.3.2 Perancangan Tampilan Halaman <i>Register</i> .....	104
5.3.3.3 Perancangan Tampilan Halaman <i>Home User</i> .....	104
5.3.3.4 Perancangan Tampilan Halaman Keranjang Belanja .....	105

5.3.3.5 Perancangan Tampilan Halaman Transaksi.....	106
5.3.3.6 Perancangan Tampilan Halaman Profil .....	107
5.3.3.7 Perancangan Tampilan Gudang Sayur .....	108
5.3.3.8 Perancangan Tampilan <i>Manage User</i> .....	109
5.3.3.9 Perancangan Tampilan <i>Manage Sayur</i> .....	110
5.3.3.10 Perancangan Tampilan <i>Manage Transaksi</i> .....	111
5.3.3.11 Perancangan Tampilan <i>Login Web</i> .....	112
5.3.3.12 Perancangan Tampilan <i>Manage Sayur Web</i> .....	113
5.3.3.13 Perancangan Tampilan <i>Manage Transaksi Web</i> .....	114
5.3.3.14 Perancangan Tampilan <i>Manage User Web</i> .....	115
5.3.3.14 Perancangan Tampilan <i>Manage User Web</i> .....	115
5.3.3.15 Perancangan Tampilan Detail Sayur Web .....	116
5.3.3.16 Perancangan Tampilan Detail Transaksi Web .....	117
<b>BAB 6 IMPLEMENTASI .....</b>	<b>118</b>
6.1 Batasan Implementasi .....	118
6.2 Spesifikasi Sistem.....	119
6.2.1 Spesifikasi Perangkat Keras .....	119
6.2.2 Spesifikasi Perangkat Lunak.....	119
6.3 Implementasi Algoritma .....	120
6.3.1 Implementasi Algoritma Tambah Data Sayur.....	120
6.3.2 Implementasi Algoritma Lihat Data Sayur .....	122
6.3.3 Implementasi Algoritma Lihat Data User .....	124
6.4 Antarmuka.....	126
<b>BAB 7 PENGUJIAN DAN ANALISIS .....</b>	<b>134</b>
7.1 Pengujian .....	134
7.1.1 Pengujian Validasi .....	134
7.2 Analisis.....	153
7.2.1 Analisis Hasil Pengujian Validasi.....	153
<b>BAB 8 PENUTUP .....</b>	<b>154</b>
8.1 Kesimpulan .....	154
8.2 Saran.....	154
<b>DAFTAR PUSTAKA .....</b>	<b>156</b>
<b>LAMPIRAN 1.....</b>	<b>158</b>

LAMPIRAN 2.....	159
-----------------	-----

## DAFTAR TABEL

Tabel 1.1 Kegiatan.....	22
Tabel 3.1 Notasi pada <i>Structured diagram</i> .....	35
Tabel 3.2 Notasi pada <i>Structured diagram</i> (lanjutan).....	36
Tabel 3.3 Notasi pada <i>Use Case Diagram</i> .....	37
Tabel 3.4 Notasi pada <i>Sequence Diagram</i> .....	38
Tabel 3.5 Notasi pada <i>Sequence Diagram</i> (lanjutan).....	39
Tabel 3.6 Notasi pada <i>Entity Relationship Diagram</i> .....	41
Tabel 5.1 Karakter Pengguna.....	48
Tabel 5.2 Kebutuhan Fungsional .....	49
Tabel 5.3 Kebutuhan Fungsional (lanjutan) .....	50
Tabel 5.4 Kebutuhan Fungsional (lanjutan) .....	51
Tabel 5.5 Kebutuhan Non-Fungsional Sistem .....	51
Tabel 5.6 Skenario <i>Use Case Sign in</i> .....	56
Tabel 5.7 Skenario <i>Use Case Sign up</i> .....	56
Tabel 5.8 Skenario <i>Use Case Lihat</i> daftar sayur pada <i>home</i> .....	57
Tabel 5.9 Skenario <i>Use Case Tambah</i> keranjang belanja.....	57
Tabel 5.10 Skenario <i>Use Case hapus</i> keranjang belanja .....	57
Tabel 5.11 Skenario <i>Use Case Search</i> sayur .....	58
Tabel 5.12 Skenario <i>Use Case Lihat Keranjang</i> belanja.....	58
Tabel 5.13 Skenario <i>Use Case Edit</i> jumlah sayur pada keranjang. ....	58
Tabel 5.14 Skenario <i>Use Case Hapus</i> daftar pembelian .....	59
Tabel 5.15 Skenario <i>Use Case Melengkapi Data pengiriman</i> .....	59
Tabel 5.16 Skenario <i>Use Case Lihat transaksi</i> .....	59
Tabel 5.17 Skenario <i>Use Case Melihat detail transaksi</i> . ....	60
Tabel 5.18 Skenario <i>Use Case lihat bantuan</i> .....	60
Tabel 5.19 Skenario <i>Use Case lihat profil</i> .....	60
Tabel 5.20 Skenario <i>Use Case Edit Profil</i> .....	61

Tabel 5.21 Skenario <i>Use Case Logout</i> .....	61
Tabel 5.22 Skenario <i>Use Case</i> lihat daftar sayur pada gudang. ....	61
Tabel 5.23 Skenario <i>Use Case</i> Menambah sayur.....	62
Tabel 5.24 Skenario <i>Use Case</i> Hapus Data sayur.....	62
Tabel 5.25 Skenario <i>Use Case</i> Edit data sayur.....	63
Tabel 5.26 Skenario <i>Use Case</i> Melihat daftar <i>user</i> .....	63
Tabel 5.27 Skenario <i>Use Case</i> Melihat data <i>user</i> .....	63
Tabel 5.28 Skenario <i>Use Case</i> Hapus data <i>user</i> .....	64
Tabel 5.29 Skenario <i>Use Case</i> Mengubah status transaksi.....	64
Tabel 5.30 Skenario <i>Use Case</i> Lihat transaksi <i>On progress</i> .....	64
Tabel 5.31 Skenario <i>Use Case</i> Lihat transaksi <i>Completed</i> .....	65
Tabel 5.32 Skenario <i>Use Case</i> <i>Login</i> pada web.....	65
Tabel 5.33 Skenario <i>Use Case</i> Manajemen sayur.....	65
Tabel 5.34 Skenario <i>Use Case</i> tambah sayur.....	66
Tabel 5.35 Skenario <i>Use Case</i> lihat data sayur. ....	66
Tabel 5.36 Skenario <i>Use Case</i> edit data sayur.....	66
Tabel 5.37 Skenario <i>Use Case</i> hapus data sayur. ....	67
Tabel 5.38 Skenario <i>Use Case</i> <i>search</i> sayur.....	67
Tabel 5.39 Skenario <i>Use Case</i> manajemen transaksi.....	67
Tabel 5.40 Skenario <i>Use Case</i> lihat detail transaksi.....	68
Tabel 5.41 Skenario <i>Use Case</i> manajemen pengguna.....	68
Tabel 5.42 Skenario <i>Use Case</i> hapus pengguna. ....	68
Tabel 5.43 Skenario <i>Use Case</i> tambah pengguna.....	69
Tabel 5.44 Skenario <i>Use Case</i> <i>logout</i> .....	69
Tabel 6.1 Spesifikasi Perangkat Keras .....	119
Tabel 6.2 Spesifikasi Perangkat Lunak .....	120
Tabel 6.3 Kode Program Tambah Sayur .....	121
Tabel 6.4 Kode Program <i>Edit</i> Data Sayur .....	123
Tabel 6.5 Kode Program Lihat Data User .....	125
Tabel 7.1 Tabel Pengujian Validasi.....	135

Tabel 7.2 Tabel Pengujian Validasi (lanjutan) .....	136
Tabel 7.3 Tabel Pengujian Validasi (lanjutan) .....	137
Tabel 7.4 Tabel Pengujian Validasi (lanjutan) .....	138
Tabel 7.5 Tabel Pengujian Validasi (lanjutan) .....	139
Tabel 7.6 Tabel Pengujian Validasi (lanjutan) .....	140
Tabel 7.7 Tabel Pengujian Validasi (lanjutan) .....	141
Tabel 7.8 Tabel Pengujian Validasi (lanjutan) .....	142
Tabel 7.9 Tabel Pengujian Validasi (lanjutan) .....	143
Tabel 7.10 Tabel Pengujian Validasi (lanjutan).....	144
Tabel 7.11 Tabel Pengujian Validasi (lanjutan).....	145
Tabel 7.12 Tabel Pengujian Validasi (lanjutan).....	146
Tabel 7.13 Tabel Pengujian Validasi (lanjutan).....	147
Tabel 7.6 Tabel Pengujian Validasi (lanjutan) .....	148
Tabel 7.14 Tabel Pengujian Validasi (lanjutan).....	149
Tabel 7.15 Tabel Pengujian Validasi (lanjutan).....	150
Tabel 7.16 Tabel Pengujian Validasi (lanjutan).....	151
Tabel 7.17 Tabel Pengujian Validasi (lanjutan).....	152

## DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi .....	24
Gambar 3.1. Illustrasi <i>Android Stack</i> .....	29
Gambar 3.2. <i>Class Diagram</i> .....	34
Gambar 3.3. <i>Package diagram</i> .....	35
Gambar 3.4. <i>Use Case Diagram</i> .....	37
Gambar 3.6. <i>Sequence Diagram</i> .....	38
Gambar 3.7. <i>Use Case Scenario</i> . .....	39
Gambar 3.7. <i>Activity diagram</i> .....	40
Gambar 4.1 Diagram Alir Metodologi .....	42
Gambar 5.2 Deskripsi Sistem .....	46
Gambar 5.3 <i>Entity Relationship Diagram</i> Untuk Aplikasi Sayur <i>Online</i> .....	52
Gambar 5.4 <i>Use Case Diagram</i> Untuk Platform Mobile Dengan Aktor Guest.....	53
Gambar 5.5 <i>Use Case Diagram</i> Untuk Platform Mobile Dengan Aktor User.....	53
Gambar 5.6 <i>Use Case Diagram</i> Untuk Platform Mobile Dengan Aktor Admin....	54
Gambar 5.7 <i>Use Case Diagram</i> Untuk Platform Web.....	55
Gambar 5.8 <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada Bagian <i>User</i> .....	70
Gambar 5.9 <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada Bagian Admin .....	71
Gambar 5.10 <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada <i>Package Admin</i> Bagian 1 .....	72
Gambar 5.11 <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada <i>Package Admin</i> Bagian 2 .....	73
Gambar 5.12 <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada <i>Package Admin</i> Bagian 3 .....	74
Gambar 5.13 Detail <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada <i>Package Fragment</i> bagian 1.....	75
Gambar 5.14 Detail <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada <i>Package Fragment</i> bagian 2 .....	76

Gambar 5.15 Detail <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada <i>Package Fragment</i> bagian 3 .....	76
Gambar 5.16 Detail <i>Class Diagram</i> Untuk Platform <i>Android</i> Pada <i>Package drawersayur</i> .....	77
Gambar 5.17 <i>Class Diagram</i> Untuk Platform Web Untuk Proses Pada Web .....	78
Gambar 5.18 <i>Class Diagram</i> Untuk Platform Web Untuk Proses <i>Web Service</i> ....	79
Gambar 5.19 Detail <i>Class Diagram</i> Untuk Platform Web Pada <i>Folder Controller</i>	80
Gambar 5.20 <i>Class Diagram</i> Untuk Platform Web Pada <i>Folder API</i> .....	81
Gambar 5.21 <i>Class Diagram</i> Untuk Platform Web Pada <i>Folder App</i> .....	82
Gambar 5.22 <i>Activity Diagram</i> “hapus data <i>user</i> ” .....	83
Gambar 5.23 <i>Activity Diagram</i> “perbarui data sayur” .....	83
Gambar 5.24 <i>Activity Diagram</i> “hapus data sayur” .....	84
Gambar 5.25 <i>Activity Diagram</i> “search sayur” .....	85
Gambar 5.26 <i>Activity Diagram</i> “menambah sayur” .....	85
Gambar 5.27 <i>Activity Diagram</i> “pesan sayur” .....	86
Gambar 5.28 <i>Activity Diagram</i> “tambah sayur pada web” .....	87
Gambar 5.29 <i>Activity Diagram</i> “search sayur pada web” .....	87
Gambar 5.30 <i>Activity Diagram</i> “ubah sayur pada web” .....	88
Gambar 5.31 <i>Activity Diagram</i> “tambah <i>user</i> pada web”.....	89
Gambar 5.32 <i>Activity Diagram</i> “hapus <i>user</i> pada web” .....	89
Gambar 5.33 <i>Activity Diagram</i> “lihat data transaksi pada web” .....	90
Gambar 5.34 <i>Sequence Diagram</i> “menambah sayur” .....	91
Gambar 5.35 <i>Sequence Diagram</i> “perbarui data sayur” .....	91
Gambar 5.36 <i>Sequence Diagram</i> “hapus sayur” .....	92
Gambar 5.37 <i>Sequence Diagram</i> “hapus <i>user</i> ” .....	93
Gambar 5.38 <i>Sequence Diagram</i> “search sayur” .....	93
Gambar 5.39 <i>Sequence Diagram</i> “pesan sayur”.....	94
Gambar 5.40 <i>Sequence Diagram</i> “Tambah sayur web” .....	95
Gambar 5.41 <i>Sequence Diagram</i> “Search sayur web” .....	95
Gambar 5.42 <i>Sequence Diagram</i> “ubah sayur web” .....	96

Gambar 5.43 Sequence Diagram “hapus user web”.....	97
Gambar 5.44 Sequence Diagram “tambah user web” .....	97
Gambar 5.45 Sequence Diagram “lihat transaksi web” .....	98
Gambar 5.46 Diagram Alir Proses <i>Login</i> Sebagai Admin .....	99
Gambar 5.47 Diagram Alir Proses Tambah Data Sayur .....	100
Gambar 5.48 Diagram Alir Proses Lihat Data Sayur .....	101
Gambar 5.49 Diagram Alir Proses Perbarui Data Sayur.....	101
Gambar 5.50 Diagram Alir Proses Hapus Data Sayur .....	102
Gambar 5.51 Perancangan Tampilan Halaman <i>Login</i> .....	103
Gambar 5.52 Perancangan Tampilan Halaman <i>Register</i> .....	104
Gambar 5.53 Perancangan Tampilan Halaman <i>Home User</i> .....	105
Gambar 5.54 Perancangan Tampilan Halaman Keranjang Belanja .....	106
Gambar 5.55 Perancangan Tampilan Halaman Transaksi .....	107
Gambar 5.56 Perancangan Tampilan Halaman Profil.....	108
Gambar 5.57 Perancangan Tampilan Halaman Gudang Sayur .....	109
Gambar 5.58 Perancangan Tampilan Halaman <i>Manage User</i> .....	110
Gambar 5.59 Perancangan Tampilan Halaman Manage Sayur.....	111
Gambar 5.60 Perancangan Tampilan Halaman <i>Manage Transaksi</i> .....	112
Gambar 5.61 Perancangan Tampilan Halaman <i>Login Web</i> .....	113
Gambar 5.62 Perancangan Tampilan Halaman <i>Manage Sayur Web</i> .....	114
Gambar 5.63 Perancangan Tampilan Halaman <i>Manage Transaksi Web</i> .....	114
Gambar 5.64 Perancangan Tampilan Halaman <i>Manage User Web</i> .....	115
Gambar 5.65 Perancangan Tampilan Halaman <i>Manage User Web</i> .....	116
Gambar 5.66 Perancangan Tampilan Halaman Detail Sayur Web .....	116
Gambar 5.67 Perancangan Tampilan Halaman Detail Transaksi Web .....	117
Gambar 6.1 Diagram Alir Implementasi.....	118
Gambar 6.2 Implementasi Halaman <i>Login</i> .....	127
Gambar 6.3 Implementasi Halaman <i>Register</i> .....	127
Gambar 6.4 Implementasi Halaman <i>Home User</i> .....	128
Gambar 6.5 Implementasi Halaman Keranjang Belanja .....	129

Gambar 6.6 Implementasi Halaman Transaksi.....	130
Gambar 6.7 Implementasi Halaman Profil .....	130
Gambar 6.8 Implementasi Halaman Gudang Sayur .....	131
Gambar 6.9 Implementasi Halaman <i>Manage User</i> .....	132
Gambar 6.10 Implementasi Halaman <i>Manage Sayur</i> .....	132
Gambar 6.11 Implementasi Halaman <i>Manage Transaksi</i> .....	133
Gambar 7.1 Diagram Alir Pengujian dan Analisis .....	134

## DAFTAR LAMPIRAN

LAMPIRAN 1.....	158
1. FOTO DOKUMENTASI KEGIATAN .....	158
LAMPIRAN 2.....	159
1. <i>USER MANUAL</i> .....	159
1.1 Admin .....	159
1.1.1 Bagian I – <i>Login</i> .....	159
1.1.2 Bagian II – Menambahkan sayur.....	160
1.1.3 Bagian III – <i>Edit</i> sayur. ....	162
1.1.4 Bagian IV – Hapus sayur .....	164
1.1.5 Bagian V – Melihat dan menghapus daftar <i>user</i> .....	164
1.1.6 Bagian VI – Melihat dan mengubah status transaksi.....	166
2.2. <i>User</i> .....	169
1.2.1 Bagian I – <i>Login</i> dan <i>Register</i> . ....	169
1.2.2 Bagian II – Memesan sayur.....	170
1.2.3 Bagian III – Melihat transaksi.....	172
1.2.4 Bagian IV – <i>Edit</i> profil .....	173
1.2.5 Bagian V – <i>Logout</i> .....	173

# BAB 1 PENDAHULUAN

## 1.1 Latar belakang

Sistem Informasi merupakan salah satu hal yang terpenting dalam era globalisasi saat ini. Dengan pesatnya pertumbuhan teknologi informasi, maka mengharuskan kita untuk turut serta dalam mengikuti perkembangan tersebut. Seiring dengan perkembangan teknologi informasi, kebutuhan akan informasi yang cepat, tepat dan akurat sangat diperlukan. Karena itu, keberadaan sistem yang terkomputerisasi yang handal sudah menjadi kebutuhan mutlak bagi perusahaan dalam proses bisnisnya. Sistem informasi berbasis android merupakan salah satu bentuk pemanfaatan *handphone* untuk untuk menghasilkan sebuah sistem informasi penjualan yang dapat mengelola data dan melakukan pembelian secara *online*.

Dalam pelayanan informasi, dalam kasus ini petani sayur tentunya belum mempunyai suatu media yang menyediakan layanan untuk menginformasikan tentang apa saja yang mereka jual secara kepada konsumen. Tentunya kita sebagai konsumen harus mendatangi pasar untuk mengetahuinya dan untuk melakukan pembelian sayur. Dengan masalah tersebut akan menimbulkan ketidak-efisienan waktu dalam melakukan kegiatan berbelanja dan melakukan kegiatan sehari-hari. Pembuatan sistem informasi dan penjualan sayur berbasis android adalah satu cara untuk mempermudah hal tersebut.

Dalam perancangan proyek tim ini, penulis dan tim menggunakan bahasa pemrograman yang digunakan oleh *Android studio* yaitu *java* dan *xml* untuk *front end* dan *webservice php framework laravel* untuk *back end*. Dengan menggunakan *Android studio* dan *php framework laravel* memudahkan penulis dan tim untuk pembuatan aplikasi ini sesuai dengan proses bisnis yang telah dijelaskan oleh pemilik dari CV. Sarana Utama Solusindo.

## **1.2 Rumusan masalah**

Berdasarkan permasalahan yang telah dijelaskan pada latar belakang maka disusun rumusan masalah sebagai berikut :

1. Bagaimanakah merancang aplikasi sayur *online* berbasis android menggunakan *webservice* ?
2. Bagaimanakah mengimplementasikan aplikasi sayur *online* berbasis android menggunakan *webservice* ?

## **1.3 Tujuan**

Tujuan dari pembuatan aplikasi ini adalah sebagai berikut:

1. Merancang aplikasi sayur *online* berbasis android.
2. Mengimplementasikan aplikasi sayur *online* berbasis android.

## **1.4 Manfaat**

Penulisan dokumentasi ini memiliki beberapa manfaat diantaranya :

1. Mempermudah konsumen untuk membeli sayur secara digital menggunakan aplikasi sayur *online* sehingga dapat membantu dalam melakukan kegiatan berbelanja tanpa harus mendatangi pasar.
2. Mempermudah petani sayur dalam menjual produknya melalui aplikasi sayur *online* ke konsumen.

## **1.5 Batasan masalah**

Batasan masalah pada pengembangan ini sebagai berikut :

1. Tidak ada forum diskusi untuk aplikasi ini
2. Pembuatan aplikasi sayur *online* berbasis android ini menggunakan bahasa pemograman *java* , *xml* dan *php framework laravel* untuk *webservice*
3. Aplikasi ini memiliki website untuk mengolah data sayur

## **1.6 Sistematika pembahasan**

Sistematika penulisan pada pengembangan aplikasi ini direncanakan sebagai berikut :

### **BAB 1 PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, sistematika penulisan laporan, dan pelaksanaan PKL.

### **BAB 2 PROFIL PERUSAHAAN**

Bab ini membahas tentang sejarah dan perkembangan serta profil CV. Sarana Utama Solusindo, Visi, Misi, Komitmen CV. Sarana Utama Solusindo, Struktur organisasi CV. Sarana Utama Solusindo beserta tugas, ruang lingkup di CV. Sarana Utama Solusindo.

### **BAB 3 LANDASAN KEPUSTAKAAN**

Bab ini berisi tentang dasar teori perangkat lunak dalam perancangan, pengembangan dan pengujian.

### **BAB 4 METODOLOGI**

Bab ini dibahas mengenai penerapan aplikasi Jual Beli Sayur *Online* dengan layanan *web service* secara detail dengan langkah-langkah pengerjaan serta tampilan gambar-gambar dari implementasi yang dilakukan.

### **BAB 5 ANALISIS DAN PERANCANGAN**

Bab ini menguraikan analisis kebutuhan serta perancangan sistem yang menjadi objek studi kasus pembuatan sistem.

### **BAB 6 IMPLEMENTASI**

Bab ini berisi tentang tahap-tahap implementasi yang telah dilakukan dalam pengembangan aplikasi.

### **BAB 7 PENGUJIAN DAN ANALISIS**

Bab ini berisi tentang tahap-tahap pengujian fungsionalitas dari sistem dan analisis hasil pengujian.

## BAB 8 PENUTUP

Bab ini menguraikan kesimpulan yang diperoleh dari pembuatan dan pengujian sistem informasi yang dibuat dalam pembuatan sistem ini ini serta saran-saran untuk pengembangan lebih lanjut.

### 1.7 Pelaksanaan Praktek Kerja Lapang

- **Waktu**

PKL dilaksanakan mulai tanggal 2 April 2018 hingga tanggal 31 Agustus 2018. Adapun kegiatan yang dilakukan selama pelaksanaan PKL di CV Sarana Utama Solusindo dapat dilihat pada Tabel 1.1 berikut ini:

**Tabel 1.1 Kegiatan**

NO	Kegiatan	Juli 2018				Agustus 2018			
		I	II	III	IV	II	III	IV	I
1	Pengenalan CV SASINDO secara garis besar dan ruang lingkup kerja	X							
2	Penentuan objek yang akan dibuat	X							
3	Pembagian tugas penggeraan baik isi laporan dan kode sistem	X							
4	Analisis kebutuhan <i>user</i>	X	X	X					
5	Implementasi Program	X	X	X	X	X	X	X	X
6	Pembuatan Laporan	X	X	X	X	X	X	X	X

- **Tempat**

PKL dilaksanakan di:

Nama Instansi : CV Sarana Utama Solusindo (SASINDO)

Alamat : Lantai 2 Jalan Anggrek Garuda 49 , Kota Malang

Kode Pos : 65141

E-mail : cs.sasindo@gmail.com

Website : <http://sasindo.id>

## BAB 2 PROFIL PERUSAHAAN

### 2.1 Sejarah dan Perkembangan CV Sarana Utama Solusindo (SASINDO)

CV. Sarana Utama Solusindo (SASINDO) adalah salah satu perusahaan berbasis Teknologi Informasi (TI) yang mendedikasikan diri untuk peningkatan fungsi TI pada setiap sendi kehidupan khususnya di Indonesia. Perusahaan ini memiliki beberapa ruang lingkup bisnis utama yaitu solusi bisnis IT / *IT Business Solution* (IT-Busol), Konsultan TI / *IT Consultant* (IT-C), Pendidikan dan Pelatihan IT/ *IT Education and Training* (IT-ET)

CV. Sarana Utama Solusindo didirikan atas dasar pemikiran dan keinginan yang sangat kuat, bahwa sudah saatnya untuk menerapkan TI di dalam setiap sendi kehidupan, karena TI dapat membantu setiap pekerjaan manusia. Diharapkan dengan menerapkan TI untuk membantu kerja manusia maka hasil dari pekerjaan akan lebih baik jika dibandingkan dengan bekerja tanpa menggunakan TI.

### 2.2 Visi, Misi dan Komitmen Perusahaan.

CV. Sarana Utama Solusindo yang bergerak dalam bidang teknologi memiliki visi, misi dan komitmen agar perusahaannya terus berkembang dan meningkatkan sendi kehidupan dengan menggunakan teknologi.

#### 2.2.1 Visi

Menjadi perusahaan TI yang terdepan dalam penerapan TI dalam setiap sendi kehidupan.

#### 2.2.2 Misi

1. Memberikan edukasi kepada masyarakat tentang pentingnya TI untuk kehidupan.

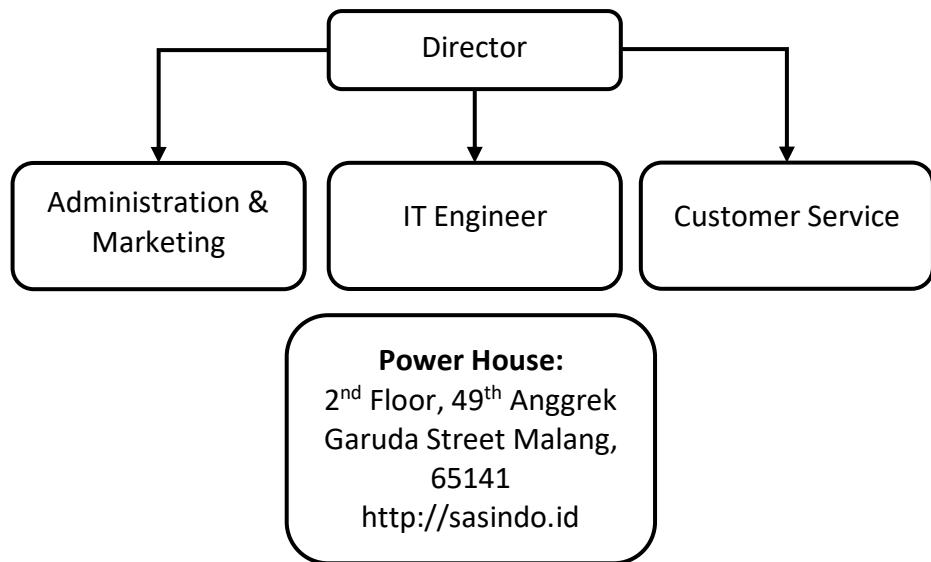
2. Memberikan solusi terbaik dalam setiap masalah yang dapat diselesaikan dengan TI.
3. Ikut serta memberikan kontribusi dalam perkembangan di dunia TI.

### **2.2.3 Komitmen**

TI dapat membantu menyelesaikan masalah manusia dalam kehidupan, untuk itu TI harus menjadi bagian dari kehidupan manusia.

## **2.3 Struktur Organisasi CV SARANA UTAMA SOLUSINDO**

Berikut ini struktur organisasi dalam CV. Sarana Utama Solusindo dijelaskan pada Gambar 2.1 :



**Gambar 2.1 Struktur Organisasi**

## **2.4 Ruang Lingkup CV SARANA UTAMA SOLUSINDO**

Beberapa pekerjaan di bidang TI yang menjadi ruang lingkup bisnis CV. Sarana Utama Solusindo adalah:

1. Solusi Bisnis TI (*IT Business Solution*)
2. Konsultan TI (*IT Consultant*)
3. Pendidikan & Pelatihan TI (*IT Training*)

## **2.4.1 Solusi Bisnis TI (IT Business Solution) dan Konsultan IT (IT Consultant)**

CV. Sarana Utama Solusindo siap menjadi partner anda dalam pengembangan TI baik itu dalam hal pengembangan perangkat lunak (*software*) maupun pengembangan perangkat keras (*hardware*). Beberpas fokus dari CV. Sarana Utama Solusindo adalah:

1. Pengembangan Sistem Informasi Manajemen (*Management Information System*)
  - Sistem Informasi Puskesmas (SIMAS) – *on going*  
Sistem informasi ini merupakan sistem informasi alur proses pengobatan yang ada di puskesmas.
  - Sistem Informasi Geografis Puskesmas (SIMAS) – *on going* Sistem informasi ini berbasis geografis dengan melakukan pemetaan wilayah menurut alamat dari pasien. Pemetaan dilakukan berdasarkan poin-poin tertentu yang ingin diketahui salah satunya berdasarkan penyakit yang diderita pasien.
  - *Command Center* (Ngalam *Command Center*) – *on going* Sistem informasi yang berfungsi untuk mengetahui informasi semua kegiatan yang ada di beberapa dinas pemerintahan secara terpusat.
  - Smart City – *on going*  
Sistem yang digunakan untuk memudahkan kegiatan yang ada di dalam pemerintahan yang berbasis teknologi.
2. Pengembangan Aplikasi Dekstop & Perangkat Bergerak (Dekstop & *Mobile Application*)
  - Aplikasi Antrean Puskesmas – *on going*
3. Pengembangan Perangkat Cerdas (*Smart Device*)
  - Deteksi Dini Bahaya Banjir dan Banjir Rob (*Flood Early Warning System*) – *completed*

- Deteksi Dini Bahaya Tsunami (*Tsunami Early Warning System*) – *completed*
- Deteksi Dini Bahaya Tanah Longsor (*Landslide Early Warning System*) – *on going*
- Pemantauan Cuaca dan Klimatologi – *completed*

#### **2.4.2 Pendidikan dan Pelatihan (IT Training)**

CV. Sarana Utama Solusindo siap memberikan pelatihan dan pendampingan dalam proses penelitian baik yang dilakukan oleh mahasiswa maupun pihak dari luar / umum.

Beberapa pelatihan yang dilakukan oleh CV. Sarana Utama Solusindo adalah:

1. Pelatihan penelitian berbasis Kecerdasan Buatan (*Artificial Intelligence*)
  - *Data Mining*
  - *Forecasting* (Peramalan)
  - *Decission Support System* (Sistem Pendukung Keputusan)
  - *Expert System* (Sistem Pakar)
  - *Optimization Algorithm* (Algoritma Optimasi)
  - *Computer Vision*
  - dll
2. Pelatihan penelitian berbasis Rekayasa Perangkat Lunak (*Software Engineering*)
  - Aplikasi Dekstop (*Dekstop Application*)
  - Aplikasi Perangkat Bergerak (*Mobile Application*)
  - Aplikasi Berbasis Web (*Web Application*)

## BAB 3 LANDASAN KEPUSTAKAAN

### 3.1 Sistem Penjualan

Kata sistem berasal dari bahasa Yunani yaitu *systema*, *systema* sendiri memiliki pengertian himpunan bagian atau komponen yang independen namun saling berhubungan secara teratur dan merupakan suatu keseluruhan yang utuh. Sistem terdiri atas struktur dan proses. Struktur sistem merupakan unsur-unsur dan elemen yang membentuk sebuah sistem itu sendiri, sedangkan proses sistem menjelaskan bagaimana cara kerja setiap unsur atau elemen yang terdapat pada sistem dapat mencapai tujuan yang telah ditentukan. Setiap sistem merupakan bagian dari sistem lain yang lebih besar dan terdiri atas berbagai sistem yang lebih kecil, yang disebut Subsistem. (Rusdiana. H.A & Irfan, Moch., 2014).

Dengan demikian dapat disimpulkan bahwa sistem merupakan kumpulan atau himpunan dari beberapa bagian yang memiliki hubungan dan keterkaitan dan juga kerja sama dalam satu kesatuan untuk mencapai tujuan dari sistem tersebut. Sebuah sistem dibuat dengan tujuan untuk mencapai sebuah hasil atau tujuan tertentu dalam ruang lingkup yang sempit.

#### 3.1.2 Sistem Jual Beli *Online*

Sistem Jual Beli *Online* merupakan sistem jual beli atau transaksi yang dilakukan dengan bantuan teknologi informasi antara satu atau dua lebih organisasi atau antara sebuah organisasi dengan satu atau lebih pelanggan akhir antara satu atau lebih jaringan komputer. (Rusdiana. H.A & Irfan, Moch., 2014).

Keuntungan utama melakukan transaksi secara *online* adalah menghemat biaya serta dapat menjangkau pelanggan yang lebih luas. Selain itu, keuntungan lain dari transaksi secara *online*, yaitu:

1. Distribusi yang lebih murah dari dokumen dan produk digital;
2. Kemampuan memberikan layanan dukungan kepada pelanggan;
3. Kanal pemasaran yang baru;

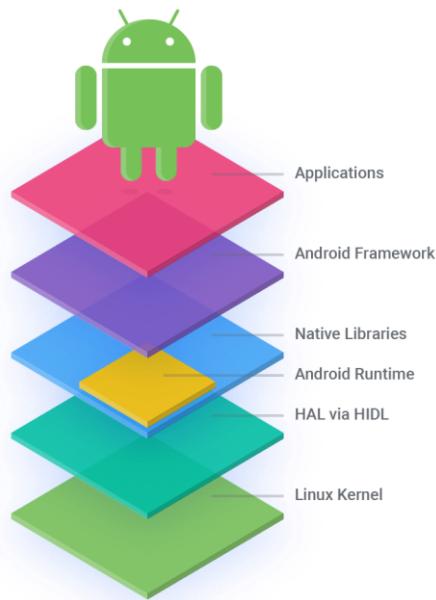
4. Memiliki kemampuan untuk menarik pelanggan baru;
5. Menyediakan satu titik lokasi kontak untuk bermacam-macam produk dan jasa;
6. Dapat digunakan sebagai media riset pasar.

Selain keuntungan atau kelebihan tersebut transaksi secara *online* juga memiliki beberapa kelemahan, yaitu :

1. Keamanan akses dan transmisi;
2. Beban *traffic* yang terlalu *overload*;
3. Kesulitan sensor dan kesulitan mengukur kinerja dari situs, keberhasilannya akan diukur dengan jumlah orang yang mengunjungi atau diukur dengan cara lain.

### **3.2 *Android***

*Android* adalah sebuah perangkat lunak atau *platform open source* untuk perangkat seluler dan merupakan sebuah proyek aplikasi *open source* yang dipimpin pengembangannya oleh *Google*. *Google* menawarkan beberapa program dalam pengembangan sebuah sistem android salah satunya adalah *Android Open Source Project* (*AOSP*) yaitu sebuah wadah informasi dan *source code* yang diperlukan untuk membuat varian kustomisasi dari *stack Android*, perangkat port dan aksesoris ke dalam platform *Android*, dan memastikan kompatibilitas sebuah perangkat untuk menjaga ekosistem dari pengembangan *Android* memiliki lingkungan yang sehat dan stabil untuk para penggunanya (*Android*, 2018).



**Gambar 3.1. Illustrasi *Android Stack***

Sumber : Android (2018)

*Android* diimplementasikan dalam bentuk tumpukan arsitektur perangkat lunak atau sebuah *Software Stack* yang terdiri dari kernel Linux, perangkat *Runtime* dan *libraries* yang sesuai, *Framework* aplikasi dan sekumpulan aplikasi. Kernel Linux bekerja sebagai jembatan antara perangkat keras dan lapisan atas, ART melakukan penerjemahan *bytecode* menjadi perintah yang dikenali oleh *processor* perangkat, *Android Libraries* menyediakan bantuan untuk menangani perintah-perintah umum (*String*, jaringan, dan manipulasi file), *Native libraries* adalah sekumpulan layanan yang merupakan dari bahasa C/C++, *Android Framework* adalah serangkaian layanan yang secara kolektif membentuk lingkungan untuk menjalankan dan mengelola android, lalu yang terakhir aplikasi yang merupakan sekumpulan aplikasi android *native* dengan implementasi tertentu. Aplikasi yang terdapat dan dibuat pada android sebagian besar ditulis dalam Bahasa *Java* dan dikompilasi ke dalam format *bytecode* pada lingkungan pengembangan aplikasi dengan *Android Studio* dan ketika aplikasi dijalankan pada perangkat maka *bytecode* tadi akan dikompilasi oleh *ART* agar dapat dibaca oleh *CPU*. Tujuan utama dari pembuatan arsitektur *Android* yang seperti itu adalah

untuk meningkatkan kinerja dan efisiensi, baik dalam mengeksekusi aplikasi yang telah dibuat dan dalam penerapan penggunaan kembali dan pengembangan dalam desain aplikasi android yang dibuat. (Smyth, 2015)

Tujuan Android adalah untuk menghindari kegagalan dalam pengembangan yaitu dimana satu pemegang industri mempunyai kendali dan membatasi inovasi yang muncul dari pengguna dan pengembang lain yang merupakan sebuah prinsip dalam pengembangan *open source* untuk tetap membuka dan memberikan ruang untuk segala inovasi. Untuk itu, Android adalah sistem operasi yang berkualitas produksi untuk menjembatani pembuatan produk-produk perangkat lunak yang baik bagi konsumen, lengkap dengan *source code* yang dapat disesuaikan dan dikembangkan dan juga dapat dipindah hampir ke semua perangkat dan memiliki dokumentasi publik yang tersedia untuk semua orang dan dapat diakses secara mudah.

### **3.2.1 *Android Studio***

*Android Studio* adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi *Android*, yang berpedoman dengan *IntelliJ IDEA* (Android Developer,2018). Selain memiliki *tools editor* dan pengembangan yang kuat dari *IntelliJ*, *Android Studio* juga menawarkan banyak fitur yang dapat digunakan dalam meningkatkan produktivitas pengguna saat membuat dan mengembangkan aplikasi *Android*, seperti:

1. Sistem bangun berbasis *Gradle* yang fleksibel
2. Emulator yang kaya dan kaya fitur
3. Lingkungan terpadu tempat Anda dapat mengembangkan untuk semua perangkat *Android*
4. Instant Run untuk mendorong perubahan pada aplikasi yang sedang berjalan tanpa membangun *APK* baru
5. Template kode dan integrasi *GitHub* untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
6. Alat dan kerangka pengujian ekstensif

7. Alat serat untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya
8. Dukungan *C++* dan *NDK*
9. Dukungan bawaan untuk *Google Cloud Platform*, sehingga mudah untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine*.

Dengan segala fitur yang telah diberikan dalam *IDE android studio*, hal tersebut memudahkan para pengembang perangkat lunak android dalam melakukan pengembangan yang lebih baik lagi. Untuk mendapatkan informasi terkait pengembangan dengan *android studio* juga sangat mudah karena telah banyak situs dan banyak pengembang yang menggunakan Android Studio sebagai media pengembangan aplikasi android. Pengembang android juga dibantu dengan banyaknya *library* yang disediakan untuk mengembangkan sebuah aplikasi yang lebih luas dan lebih bermanfaat lagi.

### **3.3 Web Service**

*Web services* adalah suatu sistem perangkat lunak yang didesain untuk mendukung interaksi mesin ke mesin pada suatu jaringan. Ia mempunyai suatu interface yang diuraikan dalam suatu format machine-processible seperti *WSDL* (*Web Service Description Language*). Sistem lain yang berinteraksi dengan *Web service* dilakukan melalui interface/antar muka menggunakan pesan seperti pada *SOAP*. Pada umumnya pesan ini melalui *HTTP* dan *XML* yang merupakan salah satu standard web. (W3C,2018)

*Web Service* dapat diakses secara *remote* oleh berbagai perangkat dengan sebuah perantara tertentu. Secara umum,*web service* dapat diidentifikasi dengan menggunakan *URL* seperti hanya web pada umumnya. Namun yang membedakan *web service* dengan web pada umumnya adalah interaksi yang diberikan oleh *web service*. Berbeda dengan *URL web* pada umumnya, *URL web service* hanya mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi yang kemudian akan diolah lagi oleh perangkat maupun aplikasi yang terhubung.

### **3.4 *Laravel***

*Laravel* adalah sebuah *framework* aplikasi web PHP *open source* dengan sintaksis yang lebih ekspresif dan ditujukan untuk pengembangan aplikasi web mengikuti pola arsitektur *MVC* (Chen, et al., 2017). *Laravel* melakukan pengembangan yang mempermudah tugas-tugas sederhana dan umum dengan kerangka yang kuat dan mudah dipahami pada sebagian besar proyek web, seperti otentifikasi, *routing*, *session* dan *caching*. Dalam pengembangannya, *Laravel* menggabungkan beberapa fitur-fitur dari *framework* lain yang pernah kita lihat sebelumnya termasuk *framework* yang diterapkan dalam bahasa lain, seperti *Ruby on Rails*, *ASP.NET MVC*, dan *Sinatra*.

*Laravel* dapat diakses secara mudah, lebih kuat dan menyediakan *tools* yang memadai untuk pengembangan aplikasi yang besar dan kuat. *Laravel* memiliki susunan yang baik sebagai sebuah kontrol, sistem migrasi yang baik, dan dukungan pengujian unit yang terintegrasi secara ketat memberi bantuan yang baik bagi pengembang untuk mengembangkan aplikasi dengan baik (Laravel, 2018).

Selain *Laravel* ada juga *framework* lain yang biasa digunakan yaitu *Codelginter*. Salah satu perbedaan dari *Laravel* dan *Codelgniter* adalah pada *Laravel* menggunakan *library* yang lebih banyak dari pada *Codelgniter* sehingga *Laravel* membutuhkan waktu yang lebih banyak pada saat menampilkan halaman web (Erinton, Negara, dan Sanjoyo, 2017).

#### **3.4.1 *Model-View-Controller (MVC)***

*Model-View-Controller* merupakan sebuah pola arsitektur yang membagi kode dalam 3 basis kode program, yaitu:

- 1. *Model***

*Model* merupakan sebuah basis dimana pengembang dapat melakukan manipulasi data. *Model* sendiri berada di antara data dan

aplikasi. Data dapat disimpan dalam berbagai jenis sistem *database* seperti *MySQL*, *file XML*, atau bahkan *Excel*.

### 2. *View*

*View* merupakan kumpulan kode yang bertugas untuk menampilkan sebuah tampilan visual data yang diterima *Controller* dari *Model*.

Sedangkan dalam *Laravel* menggunakan template Blade untuk tampilan dengan menggunakan kode PHP seperti biasa.

### 3. *Controller*.

*Controller* bertugas menangani permintaan dan meneruskan data dari *Model* menuju *View*. Jadi dapat dianggap *Controller* merupakan sebuah penghubung antara *Model* dan *View*.

Pola arsitektur MVC memungkinkan aplikasi web memiliki banyak pandangan berbeda dari satu model umum (Armel, 2014). Dalam pengembangan MVC, satu *Model* untuk satu tabel akan dapat dibuat dan melalui satu *Model* tersebut, beberapa *View* dapat dibuat. MVC dengan cepat menjadi standar industri yang digunakan dalam setiap pengembangan web secara modern. Banyak framework menjadikan MVC untuk memisahkan logika antar lapisan aplikasi.

## 3.5 *Unified Modeling Language (UML)*

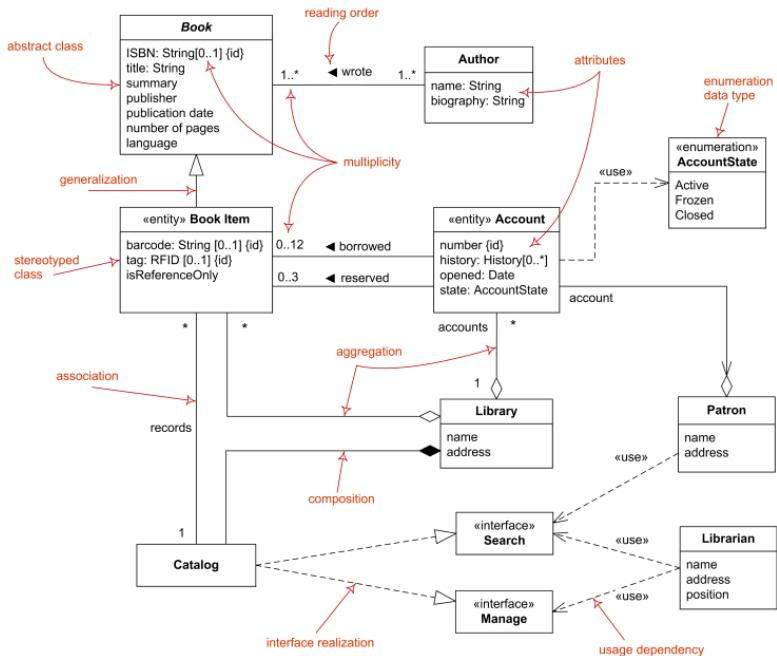
*Modelling Language* merupakan alat perancangan sistem yang berorientasi pada objek. *UML* diagram memiliki tujuan utama untuk membantu tim pengembangan proyek untuk dapat berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program (*UML Diagrams*, 2018). *UML* mempunyai tiga kategori utama yaitu struktur diagram, *behaviour* diagram dan *interaction* diagram. Dimana masing-masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi.

### 3.5.1 Structured Diagram.

Diagram dengan model terstruktur menggambarkan organisasi suatu sistem dalam hal komponen-komponen yang membentuk sistem tersebut seperti kelas, objek, metode yang digunakan serta hubungan antar komponen tersebut dalam sistem. (Sommerville, 2011) Struktur diagram yang digunakan dalam pembuatan aplikasi ini, yaitu :

#### 1. Class Diagram.

*Class diagram* Menggambarkan struktur statis dari kelas dalam sistem yang dibuat dan menggambarkan atribut, operasi dan hubungan antara kelas. Berikut merupakan contoh dari *Class Diagram* :

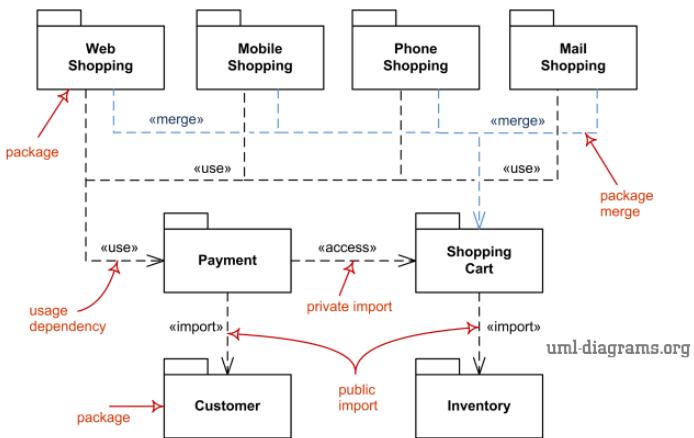


Gambar 3.2. Class Diagram

Sumber : UML Diagrams(2018)

#### 2. Package Diagram.

*Package Diagram* biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Dalam penggunaan notasi pada *package diagram* memiliki persamaan dengan penggunaan pada *class diagram*. Berikut merupakan contoh dari *Pakcage Diagram* :



**Gambar 3.3. Package diagram**

Sumber : UML Diagrams (2018)

**Tabel 3.1 Notasi pada Structured diagram**

No	Tipe	Deskripsi	Simbol
1	<i>Class</i>	Notasi yang digunakan untuk menggambarkan sebuah objek, atribut, maupun operasi.	<pre> Class name -Attribute name -/derived attribute name +Operation name () </pre>
2	<i>Association</i>	Notasi yang digunakan untuk menunjukkan hubungan antar kelas, banyak kelas atau dengan kelas itu sendiri.	<pre> 1..*   verb phrase   0..1 </pre>
3	<i>Package</i>	Notasi yang digunakan untuk menunjukkan package atau kompulan dari kelas yang dibungkus dalam package.	
4	<i>Generalization</i>	Notasi yang digunakan untuk menunjukkan hubungan antar kelas yang menggunakan pewarisan dari kelas utama ke kelas anak.	

**Tabel 3.2 Notasi pada Structured diagram (lanjutan)**

5	<i>Dependency</i>	Notasi yang menunjukkan hubungan kelas yang bergantung pada kelas lain.	
6	<i>Agregation</i>	Notasi yang menunjukkan hubungan kelas yang terdiri dari beberapa komponen yang lebih kecil.	

Sumber : Dennis (2012)

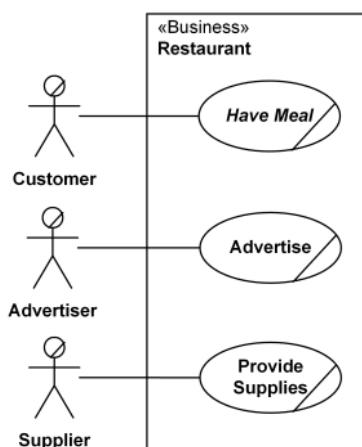
### 3.5.2 *Behavior Diagram*.

Menggambarkan ciri-ciri *behavior/metode/ fungsi* dari sebuah sistem secara dinamis atau *business process*. *Behavior Diagram* menunjukkan apa yang terjadi atau apa yang seharusnya terjadi ketika suatu sistem merespons atau menerima suatu stimulus atau *trigger* dari lingkungannya (Sommerville,2011).

*Behavior diagram* yang digunakan dalam pembuatan aplikasi ini, yaitu :

#### 1. *Use Case Diagram*

Menggambarkan untuk menggambarkan serangkaian tindakan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Setiap *use case* harus memberikan beberapa hasil yang dapat diamati dan berharga kepada para aktor atau stakeholder lain dari sistem. Berikut merupakan contoh dari *Use Case Diagram* :



**Gambar 3.4. Use Case Diagram**

Sumber : UML Diagrams (2018)

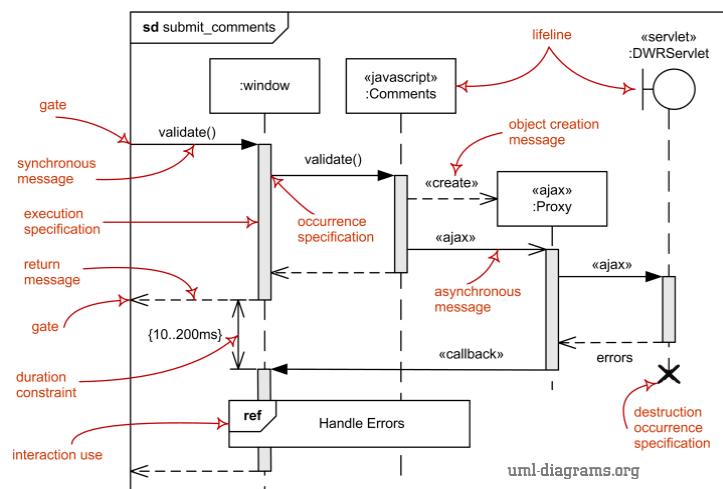
**Tabel 3.3 Notasi pada Use Case Diagram**

No	Tipe	Deskripsi	Simbol
1	<i>Actor</i>	Notasi yang digunakan untuk menggambarkan seseorang ( <i>user</i> ) atau sistem lain yang menggunakan fitur atau fungsi yang berada dalam sistem. Perlu dicatat bahwa <i>actor</i> berinteraksi dengan <i>use case</i> tetapi tidak memiliki kontrol atas <i>use case</i> .	
2	<i>Use case</i>	Notasi yang digunakan untuk menggambarkan fungsionalitas atau fitur yang disediakan oleh sistem yang saling bertukar pesan antar unit dengan <i>actor</i> , yang dinyatakan dalam bentuk kata kerja.	
3	<i>Asociation</i>	Notasi yang digunakan untuk menunjukkan hubungan antar siapa dan apa yang meminta interaksi secara langsung antar <i>use case</i> maupun <i>actor</i> .	
4	<i>extend</i>	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika dalam kondisi tertentu atau memiliki syarat.	
5	<i>include</i>	<i>Include</i> , merupakan di dalam <i>use case</i> lain atau pemanggilan <i>use case</i> oleh <i>use case</i> lain.	

Sumber : Dennis (2012)

## 2. Sequence Diagram

*Sequence diagram* adalah jenis diagram interaksi yang paling umum, yang berfokus pada pertukaran pesan antara sejumlah *lifetime* (Dennis ,et al., 2012). Sequence diagram menggambarkan interaksi dengan berfokus pada urutan pesan yang dipertukarkan, bersama dengan spesifikasi kemunculannya yang sesuai pada *lifetime*. Berikut merupakan contoh dari *Sequence Diagram* :



**Gambar 3.6. Sequence Diagram**

Sumber : UML Diagrams(2018)

**Tabel 3.4 Notasi pada Sequence Diagram**

No	Tipe	Deskripsi	Simbol
1	Actor	Notasi yang digunakan untuk menggambarkan seseorang ( <i>user</i> ) atau sistem lain yang menggunakan fitur atau fungsi yang berada dalam sistem.	
2	Object	Notasi yang digunakan untuk menggambarkan <i>object</i> yang berpartisipasi dalam sebuah <i>sequence</i> dengan mengirim pesan atau menerima pesan.	

**Tabel 3.5 Notasi pada Sequence Diagram (lanjutan)**

3	<i>Lifeline</i>	Notasi yang digunakan untuk menunjukkan lama hidup suatu <i>object</i> dalam <i>sequence</i> .	
4	<i>Focus Control</i>	Notasi yang digunakan untuk menunjukkan bahwa sebuah <i>object</i> sedang aktif dan menjalankan tugasnya.	
5	<i>Message</i>	Notasi yang digunakan untuk menunjukkan bahwa sebuah pesan sedang dikirim dari suatu <i>object</i> ke <i>object</i> yang lain.	

Sumber : Dennis (2012)

### 3. Use Case Scenario

Menggambarkan *Use Case* secara detail dengan segala skenario yang ditetapkan dan yang mungkin terjadi. *Use Case* skenario berguna saat nanti dalam proses pengujian. Berikut merupakan contoh dari *Use Case* Skenario:

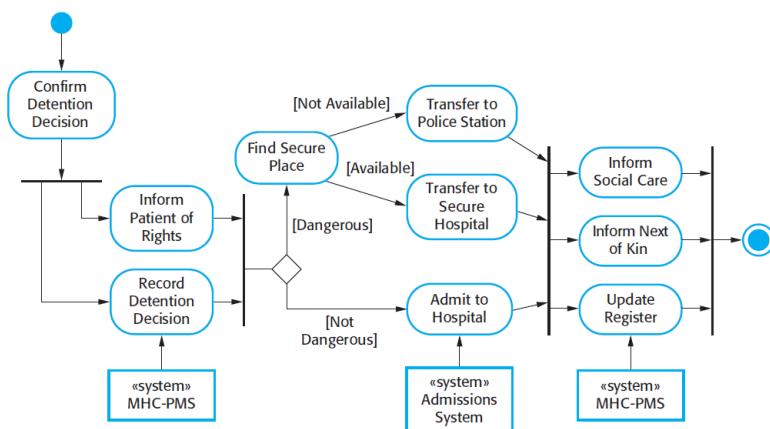
4.1 Create User Group	
<b>Brief Description</b>	Create User Group is the functionality by which an authorized individual creates a new user group.  Please note the two group types and details of their creation: <ul style="list-style-type: none"><li>• (Public) General Group: can only be created by an xxx User Administrator.</li><li>• (Public) Article Group: will be created whenever a user is launched around an Article Available for User in the xxx environment.</li></ul>
<b>Use Case Pre-Conditions</b>	1. The UGC has been identified by the system. 2. The UGC has the appropriate permission to create the user group.
<b>Use Case Post-Conditions</b>	1. The user group has been created and is present and persistent in the system with the appropriate attributes.
<b>List of Actors</b>	• User Group Creator (UGC): <ol style="list-style-type: none"><li>1. User Administrator</li><li>2. System</li></ol>
<b>User Experience Links</b>	
<b>Basic Flow One: General Group</b>	1. The use case begins when the User Administrator accesses the Manage User interface of xxx. 2. The User Administrator indicates that s/he wants to create a new group and provides the requested information: <ol style="list-style-type: none"><li>a. Title of user group (required)</li><li>b. Description of user group (required)</li></ol>
<b>Basic Flow Two: Article Group</b>	Creation of this group is a post-condition of the Create Topic use case for "(Public) Article Topic." Please recognize that the end user of the system is oblivious to the handling of the "create (Public) Article Group" step.
<b>Alternate Flows</b>	

**Gambar 3.7. Use Case Scenario.**

Sumber : Bittner (2006)

#### 4. Activity Diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain *activity diagram* menunjukkan aktivitas yang terlibat dan yang dilaksanakan ketika menjalankan suatu proses atau pemrosesan data (Sommerville, 2011). *Activity diagram* dimaksudkan untuk menunjukkan aktifitas yang terjadi pada suatu proses yang dijalankan di dalam sistem. Awal proses ditunjukan dengan lingkaran berisi warna dan akhir proses ditunjukkan dengan lingkaran yang berisi warna di dalam lingkaran lain. Kegiatan atau aktifitas yang harus dijalani ditandai dengan persegi bersudut bulat. Tanda panah menunjukan perpindahan aktifitas ke aktifitas yang lainnya.



Gambar 3.7. *Activity diagram*.

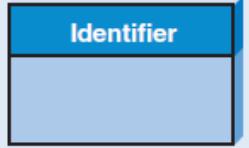
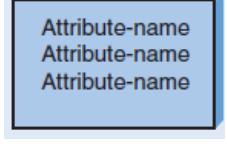
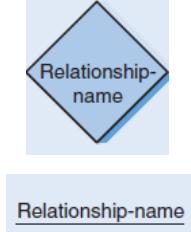
Sumber : Sommerville (2011)

#### 3.6 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) merupakan sebuah diagram yang menggambarkan tentang bagaimana sebuah informasi dibuat, disimpan dan digunakan dalam sebuah sistem (Dennis, 2012). Dalam ERD kita dapat melihat informasi yang digunakan dalam sistem dan keterkaitannya dengan informasi lainnya. Informasi yang memiliki kesamaan ditempatkan dalam kotak yang disebut entitas. Antar entitas diberikan garis penghubung yang menggambarkan

hubungan antar entitas dan diberikan simbol yang menggambarkan aturan yang berlaku antar entitas dalam sistem.

**Tabel 3.6 Notasi pada *Entity Relationship Diagram***

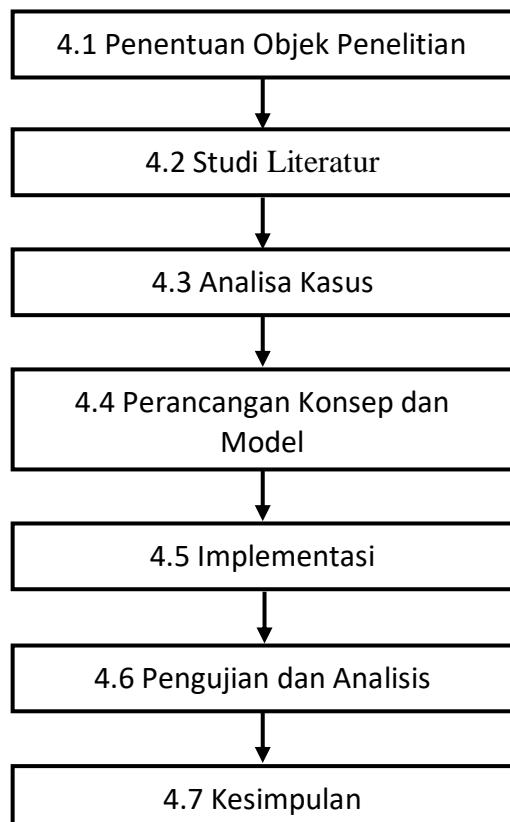
No	Tipe	Deskripsi	Simbol
1	<i>Entity</i>	Notasi yang digunakan untuk menggambarkan seseorang ( <i>user</i> ) atau tempat atau benda, memiliki <i>identifier</i> dan memiliki lebih dari 1 data.	
2	<i>Attribute</i>	Notasi yang digunakan untuk menggambarkan bagian dari sebuah <i>entity</i> .	
3	<i>Relation</i>	Notasi yang digunakan untuk menggambarkan hubungan antar <i>entity</i> , merupakan kata kerja, memiliki cardinality (1:1, 1:N, M:N), memiliki modal ( <i>null</i> , <i>not null</i> ).	

Sumber : Dennis (2012)

## BAB 4 METODOLOGI

Metode penelitian ini akan membahas mengenai jual beli sayur *online* berbasis *android* menggunakan *webservice*. Tahapan metodologi penelitian yang digunakan pada laporan PKL ini adalah penentuan objek, studi literatur, analisa kasus, konsep dan desain sistem, implementasi sistem, pengujian dan analisis, serta penarikan kesimpulan.

Adapun diagram alir metodologi tersebut dijelaskan pada Gambar 4.1 berikut :



Gambar 4.1 Diagram Alir Metodologi

### 4.1 Penentuan Objek Penelitian

Merupakan langkah awal yang bertujuan menentukan objek penentuan yang bisa didapat dari kegiatan PKL yang dilakukan di CV. Sarana Utama Solusindo

yang kemudian bisa dijadikan suatu yang akan dikerjakan. Melakukan diskusi awal dengan *software developer* CV. Sarana Utama Solusindo yang bersangkutan, kemudian diberikan daftar objek apa saja yang dapat dijadikan sebagai objek untuk dikerjakan pada PKL ini.

## 4.2 Studi Literatur

Dalam tahap studi literatur, dilakukan studi pustaka dan literatur untuk mendapatkan informasi yang berhubungan dengan pengembangan sistem, sehingga informasi tersebut diharapkan dapat mempermudah dan membantu dalam melaksanakan pengembangan sistem. Literatur-literatur yang digunakan diperoleh dari buku, *paper* nasional maupun internasional dan dokumentasi internet.

## 4.3 Analisa Kasus

Analisa kasus ini digunakan untuk mengetahui informasi mengenai data sayur yang dibutuhkan melalui proses *input* data yang dilakukan oleh pihak penjual atau tengkulak sayur , gudang sayur akan digunakan untuk menyediakan sayur yang akan diberikan kepada tengkulak atau penjual sayur, pembeli yang akan membeli dan melihat sayur yang dijual ,dengan data tersebut kelompok kami mengembangkan pengolahan data tersebut ke dalam suatu aplikasi sayur *online*, serta didukung pengumpulan data secara tidak langsung melalui internet.

## 4.4 Perancangan Konsep dan Model

Proses perancangan konsep dan model sistem dibagi menjadi beberapa tahapan yaitu:

- a. Menganalisa kebutuhan sistem dan menerjemahkan ke dalam fungsi fungsi pada aplikasi.
- b. Menentukan objek aplikasi berdasarkan kebutuhan yang ada.
- c. Pembuatan desain *interface*.

Pada tahap ini dilakukan proses perancangan sistem untuk mengimplementasikan ke dalam aplikasi. Adapun proses-proses yang dilakukan dalam tahap perancangan ini adalah analisis kebutuhan perangkat lunak dan perancangan sayur *online* berbasis *android* menggunakan *web service*.

#### **4.5 Implementasi**

Pada tahap ini dilakukan proses pengimplementasian rancangan sistem yang telah ditentukan pada tahap sebelumnya menjadi suatu sistem. Implementasi sistem dibuat dengan menggunakan *php framework laravel*. Kemudian untuk *user interface* menggunakan android studio dengan bahasa pemrograman *java* dan *xml*.

#### **4.6 Pengujian dan Analisis**

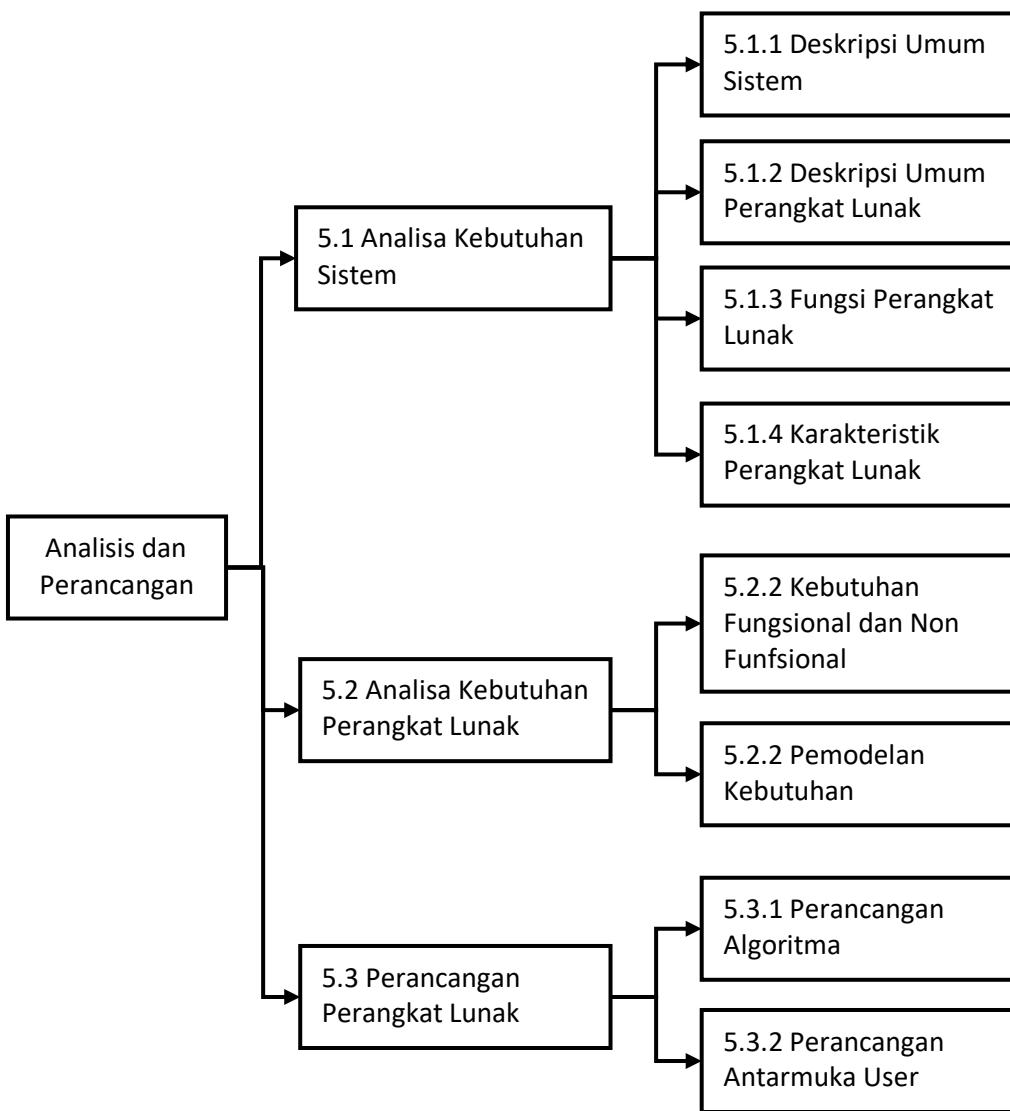
Pengujian yang akan digunakan yaitu pengujian *Black Box*. Pengujian *Black Box* merupakan pengujian fungsionalitas sistem untuk mengetahui apakah sistem telah berjalan dengan benar sesuai dengan parameter yang telah ditentukan. Uji coba ini dilakukan oleh tim internal dan juga dilakukan oleh calon pengguna untuk mengetahui kelebihan dan kekurangan sistem.

#### **4.7 Kesimpulan**

Penarikan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, dan pengujian sistem aplikasi telah dilaksanakan. Kesimpulan diambil melalui hasil pengujian dan analisis terhadap sistem yang dibuat.

## BAB 5 ANALISIS DAN PERANCANGAN

Perancangan aplikasi dalam pembuatan sistem sayur *online* diawali dengan analisis. Analisis dilakukan untuk mengetahui kebutuhan-kebutuhan sistem yang dijelaskan pada analisa kebutuhan sistem dan kebutuhan perangkat lunak, kemudian dilakukan perancangan sistem dari hasil analisa berupa perancangan perangkat lunak seperti pada Gambar 5.1.



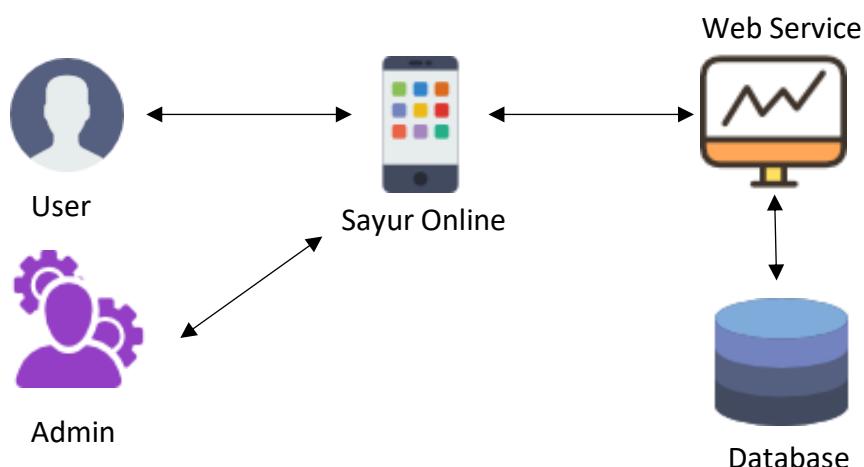
Gambar 5.1 Diagram Alir Analisis dan Perancangan

## 5.1 Analisa Kebutuhan Sistem

Pada tahap analisa kebutuhan sistem bertujuan untuk memodelkan informasi yang akan digunakan pada tahap perancangan. Analisa kebutuhan sistem yang diperlukan meliputi deskripsi umum sistem, deskripsi umum perangkat lunak, fungsi perangkat lunak, karakteristik pengguna, dan lingkungan operasi.

### 5.1.1 Deskripsi Umum Sistem

User dapat melakukan instalasi aplikasi pada perangkat smartphone. Pada perangkat user dijalankan aplikasi sayur *online* , untuk menghubungkan sistem aplikasi sayur *online* ke *web service* dibutuhkan koneksi, melalui url *web service* yang mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi. Database digunakan sebagai media penyimpanan data dari aplikasi sayur online. Berikut adalah deskripsi item sistem dari aplikasi sayur online yang ditunjukkan pada gambar 5.2



Gambar 5.2 Deskripsi Sistem

### **5.1.2 Deskripsi Umum Perangkat Lunak**

Perangkat lunak sayur online dibuat berbasis *android* menggunakan *java* dan *xml* yang dihubungkan ke *web service*. Sistem dapat berinteraksi dengan web service untuk menjalankan perintah *user* dan admin, seperti pada fitur lihat data dimana admin dapat melihat data sayur , data *user* dan transaksi yang dilakukan oleh *user*. Data sayur ditampilkan pada menu beranda admin. Admin dapat melakukan edit data melalui fitur edit yang disediakan pada masing-masing data sayur , data *user* , data transaksi, dan profil admin. Untuk melakukan hapus data, *user* dapat menghapus data dari sistem pada fitur hapus data yang ditampilkan. Admin juga dapat melakukan tambah sayur yang akan dijual dari gudang. Untuk fitur yang digunakan oleh *user* adalah melakukan edit profile , pembelian sayur , melihat data sayur , menyimpan barang belanja ke *cart* , melihat detail transaksi dan ada fitur bantuan jika *user* bingung dengan aplikasi sayur online.

Untuk mengelola seluruh informasi pada sistem, masuk ke dalam sistem melalui fitur login dengan menginputkan *username* serta *password* admin atau *user* dan keluar sistem melalui fitur *logout*.

### **5.1.3 Fungsi Perangkat Lunak**

Fungsi aplikasi sayur online ini diantaranya :

1. Mempermudah konsumen untuk membeli sayur secara digital menggunakan aplikasi sayur online sehingga dapat membantu dalam melakukan kegiatan berbelanja tanpa harus mendatangi pasar.
2. Mempermudah petani sayur dalam menjual produknya melalui aplikasi sayur online ke konsumen.

### **5.1.4 Karakteristik Pengguna**

Karakteristik pengguna digunakan untuk melihat fitur apa saja yang terdapat didalam aplikasi sayur online. Berikut adalah deskripsi fitur sistem dari aplikasi sayur online yang ditunjukan pada tabel 5.1.

**Tabel 5.1 Karakter Pengguna**

No	Identifikasi Pengguna	Karakteristik
1	User	Melakukan login Melakukan register Melihat data sayur Menyimpan barang belanja ke <i>cart</i> Membeli sayur Melihat fitur bantuan Melihat status transaksi pembelian Mengubah profile Melakukan logout
2	Admin	Melakukan <i>login</i> Melihat data sayur Meng-edit data sayur Menghapus data sayur Menambah data sayur dari gudang Melihat data <i>user</i> Menghapus data <i>user</i> Melihat status transaksi pembelian Mengubah status transaksi pembelian Mengubah profil Melakukan <i>logout</i>

## 5.2 Analisa Kebutuhan Perangkat Lunak

Pada tahap analisa kebutuhan perangkat lunak bertujuan untuk memodelkan kebutuhan fungsional dan non-fungsional dari perangkat lunak yang akan digunakan, serta menjelaskan mengenai kebutuhan antarmuka dan pemodelan dalam bentuk diagram *use case*, scenario *use case*, *sequence diagram* dan *class diagram*.

### 5.2.1 Kebutuhan Fungsional dan Non-Fungsional

#### a) Kebutuhan Fungsional

Kebutuhan fungsional adalah layanan sistem yang harus disediakan, bagaimana sistem bereaksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu. Berikut adalah kebutuhan fungsional dari aplikasi sayur online yang ditunjukan pada tabel 5.2.

**Tabel 5.2 Kebutuhan Fungsional**

No	Aktor	Nama Fungsi	Deskripsi
1	User/Admin	Login	Untuk menyeleksi pengguna yang memiliki autorisasi agar dapat mengakses ke dalam sistem dengan memasukkan <i>email</i> dan <i>password</i>
2	User/Admin	Lihat daftar sayur	User/admin dapat melihat tampilan data sayur yang dijual dalam bentuk <i>grid layout</i> .
3	Admin	Lihat data user	Admin dapat melihat tampilan data User yang telah melakukan register
4	User /Admin	Lihat detail transaksi pembelian	Admin/user dapat melihat tampilan detail transaksi pembelian.
5	Admin	Mengubah status transaksi	Admin dapat melakukan merubah atau edit status transaksi Pembelian jika barang sudah dikirim ataupun diterima
6	User/Admin	Lihat transaksi	User/Admin dapat melihar daftar transaksi di dalam menu transaksi,
7	User/Admin	Search sayur	User/Admin dapat mencari sayur yang diinginkan di halaman <i>home</i> .

**Tabel 5.3 Kebutuhan Fungsional (lanjutan)**

8	Admin	Edit data sayur	Admin dapat melakukan merubah atau edit pada data sayur yang tersimpan dalam
9	Admin	Hapus data sayur	Admin dapat melakukan hapus terhadap data sayur yang tersimpan dalam sistem
10	Admin	Menambah data sayur dari gudang	Admin dapat melakukan penambahan sayur yang akan dijual melalui gudang.
11	Admin	Hapus Data User	Admin dapat melakukan hapus terhadap data <i>user</i> yang tersimpan dalam sistem
12	User /Admin	Edit profil	<i>User/admin</i> dapat melakukan <i>edit profile</i> untuk kelengkapan data dari pengguna user ataupun admin
13	User/Admin	Lihat profil	<i>User/Admin</i> dapat melihat profil
14	User	Register	<i>User</i> dapat melakukan pendaftaran pada aplikasi sayur online untuk membeli sayur yang tersedia.
15	User	Order sayur	<i>User</i> dapat memilih dan melakukan penyimpanan data sayur yang akan dibeli ke dalam fitur <i>cart</i> .
16	User	Melengkapi data transaksi	<i>User</i> dapat melakukan pembelian sayur dengan melengkapi data transaksi pada keranjang.
17	User	Edit jumlah sayur pada keranjang	<i>User</i> dapat mengubah jumlah sayur yang terdapat di dalam keranjang.

**Tabel 5.4 Kebutuhan Fungsional (lanjutan)**

18	User	Hapus daftar pembelian	User dapat menghapus daftar sayur yang terdapat di dalam keranjang.
19	User	Melihat keranjang belanja.	User dapat melihat sayur yang dibeli di dalam keranjang
20	Admin/User	<i>Logout</i>	Admin/User keluar dari sistem

**b) Kebutuhan Non-Fungsional**

Kebutuhan non-fungsional adalah batasan atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses dan standarisasi. Berikut adalah kebutuhan non-fungsional dari aplikasi sayur online yang ditunjukkan pada tabel 5.5.

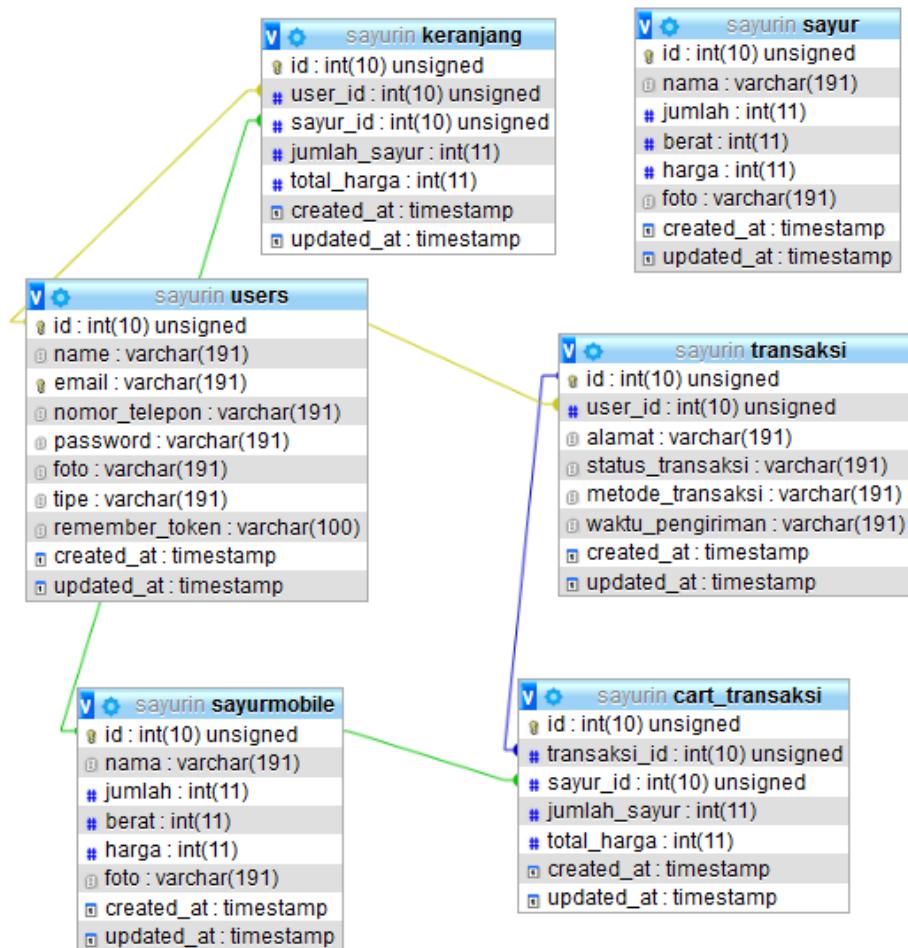
**Tabel 5.5 Kebutuhan Non-Fungsional Sistem**

No	Kebutuhan	Deskripsi
1	Availability	Dengan menguji ketersedian sistem untuk dapat diakses dengan batas minimal waktu operasi 20 jam per hari tanpa <i>downtime</i>
2	Portability	Menguji bahwa aplikasi dapat dijalankan pada smartphone <i>android</i>

## 5.2.2 Pemodelan Kebutuhan

### 5.2.2.1 Entity Relationship Diagram

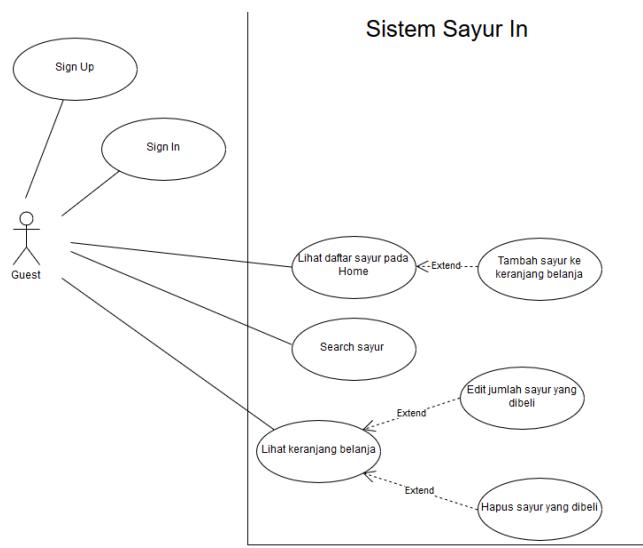
*Entity relationship diagram* digunakan untuk menggambarkan perencanaan data yang akan digunakan pada sistem. Berikut adalah *entity relationship diagram* dari aplikasi sayur online yang ditunjukkan pada gambar 5.3.



**Gambar 5.3 Entity Relationship Diagram Untuk Aplikasi Sayur Online**

#### 5.2.2.2 Use Case Diagram

Diagram *use case* digunakan untuk menggambarkan kebutuhan-kebutuhan dan fungsionalitas dari sistem serta digunakan untuk menunjukkan aksi-aksi yang dilakukan oleh aktor terhadap sistem. Berikut adalah use case diagram dari aplikasi sayur online yang ditunjukkan pada gambar 5.4, 5.5, 5.6, dan 5.7.



**Gambar 5.4 Use Case Diagram Untuk Platform Mobile Dengan Aktor Guest**



**Gambar 5.5 Use Case Diagram Untuk Platform Mobile Dengan Aktor User**



**Gambar 5.6 Use Case Diagram Untuk Platform Mobile Dengan Aktor Admin**



**Gambar 5.7 Use Case Diagram Untuk Platform Web**

### 5.2.2.3 Use Case Skenario

Untuk menjelaskan diagram *use case* secara terperinci dalam skenario *use case*. Penggunaan skenario *use case* ini bertujuan untuk mendapatkan deskripsi secara global mengenai *use case*, kondisi awal dan akhir yang harus dipenuhi oleh *use case* setelah fungsionalitas selesai dijalankan. Dalam skenario ini akan diulas bagaimana tanggapan sistem terhadap aksi yang dilakukan oleh aktor. Berikut adalah skenario *use case* dari aplikasi sayur online pada platform mobile

yang ditunjukan pada tabel 5.6 – 5.31 dan skenario *use case* dari sayur online pada platform web yang ditujukan pada tabel 5.32-5.44.

Tabel 5.6 – 5.31 merupakan skenario *use case* pada aplikasi android Sayur Online.

**Tabel 5.6 Skenario Use Case Sign in**

<i>Use case Sign in.</i>	
Tujuan	Masuk ke dalam sistem.
Aktor	Guest.
Kondisi awal	Halaman <i>Sign in</i> .
Skenario Utama	<ol style="list-style-type: none"> <li>1. Guest mengisi data pada kolom yang kosong,</li> <li>2. Data yang dimasukkan tervalidasi</li> <li>3. Menuju halaman home.</li> </ol>
Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Jika terdapat kolom kosong atau belum diisi sistem akan meminta melengkapi kolom sign in.</li> </ol>
Kondisi akhir	Menampilkan halaman home.

**Tabel 5.7 Skenario Use Case Sign up**

<i>Use case Sign up.</i>	
Tujuan	Mendaftar ke dalam sistem.
Aktor	Guest.
Kondisi awal	Halaman <i>Sign in</i> .
Skenario Utama	<ol style="list-style-type: none"> <li>1. Menekan tombol <i>sign up</i>.</li> <li>2. Guest mengisi kolom dengan lengkap dan format yang benar.</li> <li>3. Guest menekan tombol sign up.</li> <li>4. Guest terdaftar dan dapat <i>login</i> pada halaman <i>sign in</i>.</li> </ol>
Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Jika terdapat kolom kosong atau belum diisi sistem akan meminta melengkapi kolom sign in.</li> <li>2. Jika Mengisi dengan format yang salah maka muncul pesan “format salah”</li> </ol>
Kondisi akhir	Guest terdaftar dalam sistem.

**Tabel 5.8 Skenario Use Case Lihat daftar sayur pada home**

<i>Use case</i> melihat daftar sayur.	
Tujuan	Melihat daftar sayur.
Aktor	User/Admin.
Kondisi awal	Halaman Home
Skenario Utama	1. Guest/User Melihat daftar sayur pada home.
Skenario Alternatif	-
Kondisi akhir	Guest/User Melihat daftar sayur pada home.

**Tabel 5.9 Skenario Use Case Tambah keranjang belanja**

<i>Use case</i> Tambah keranjang belanja.	
Tujuan	Menambahkan Sayur pada keranjang belanja
Aktor	User.
Kondisi awal	Halaman Home
Skenario Utama	1. User Menekan tombol “Add” pada sayur yang ingin dibeli lalu muncul keranjang pada bagian bawah halaman. 2. User dapat mengurangi dan menambahkan sayur yang ingin dibeli dengan menekan tombol tambah dan kurang pada sayur.
Skenario Alternatif	-
Kondisi akhir.	Sayur yang ditambahkan muncul pada keranjang.

**Tabel 5.10 Skenario Use Case hapus keranjang belanja**

<i>Use case</i> hapus keranjang belanja.	
Tujuan	Menghapus Sayur pada keranjang belanja
Aktor	User.
Kondisi awal	Halaman Home
Skenario Utama	1. Menekan tombol kurang pada daftar sayur hingga kuantitasnya 0. Maka sayur pada keranjang belanja akan hilang.
Skenario Alternatif	-
Kondisi akhir	Sayur pada keranjang terhapus.

**Tabel 5.11 Skenario Use Case Search sayur**

<i>Use case Search sayur.</i>	
Tujuan	Search Sayur
Aktor	Guest/User/Admin.
Kondisi awal	Halaman Home
Skenario Utama	1. Guest/User/Admin Mengisikan nama sayur yang ingin dicari pada kolom <i>Search</i> .
Skenario Alternatif	-
Kondisi akhir	Menampilkan sayur yang dicari.

**Tabel 5.12 Skenario Use Case Lihat Keranjang belanja**

<i>Use case lihat keranjang belanja.</i>	
Tujuan	Melihat item pada keranjang belanja.
Aktor	Guest/User.
Kondisi awal	Halaman Home
Skenario Utama	1. Guest/User Menekan tombol “Checkout” pada bottomsheets.
Skenario Alternatif	-
Kondisi akhir	Menampilkan keranjang belanja.

**Tabel 5.13 Skenario Use Case Edit jumlah sayur pada keranjang.**

<i>Use case edit jumlah sayur pada keranjang.</i>	
Tujuan	Mengubah jumlah sayur pada keranjang belanja.
Aktor	User.
Kondisi awal	Halaman <i>Checkout</i> .
Skenario Utama	1. Menekan tombol “+” dan “-” pada sayur yang diinginkan untuk mengubah jumlah sayur.
Skenario Alternatif	-
Kondisi akhir	Jumlah sayur berubah.

**Tabel 5.14 Skenario Use Case Hapus daftar pembelian**

<i>Use case</i> hapus daftar pembelian.	
Tujuan	Menghapus daftar pembelian.
Aktor	<i>User</i> .
Kondisi awal	Halaman <i>Checkout</i> .
Skenario Utama	1. Menekan tombol hapus atau tombol tong sampah pada sayur yang diinginkan dihapus.
Skenario Alternatif	-
Kondisi akhir	Daftar pembelian terhapus.

**Tabel 5.15 Skenario Use Case Melengkapi Data pengiriman**

<i>Use case</i> melengkapi data pengiriman	
Tujuan	Melengkapi data pengiriman.
Aktor	<i>User</i> .
Kondisi awal	Halaman <i>Checkout</i> .
Skenario Utama	1. <i>User</i> mengisi data tanggal pengiriman dan waktu pengiriman. 2. <i>User</i> mengisi alamat yang ingin dituju. 3. Jika sudah lengkap <i>user</i> dapat menekan tombol “Checkout” untuk melakukan transaksi.
Skenario Alternatif	-
Kondisi akhir	Transaksi berhasil dan kembali ke halaman <i>home</i> .

**Tabel 5.16 Skenario Use Case Lihat transaksi.**

<i>Use case</i> Lihat Transaksi.	
Tujuan	Melihat <i>history</i> transaksi.
Aktor	<i>User/Admin</i> .
Kondisi awal	Halaman <i>Home</i> .
Skenario Utama	1. <i>User/Admin</i> Menekan menu <i>history</i> .
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman <i>history</i> .

**Tabel 5.17 Skenario Use Case Melihat detail transaksi.**

<i>Use case</i> Lihat detail transaksi.	
Tujuan	Melihat detail transaksi.
Aktor	User/Admin.
Kondisi awal	Halaman <i>History</i> .
Skenario Utama	1. User/Admin Menekan daftar transaksi pada halaman <i>history</i> .
Skenario Alternatif	-
Kondisi akhir	Menampilkan detail transaksi.

**Tabel 5.18 Skenario Use Case lihat bantuan**

<i>Use case</i> lihat bantuan	
Tujuan	Melihat bantuan.
Aktor	User.
Kondisi awal	Halaman <i>Home</i> .
Skenario Utama	1. User Menekan menu Help.
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman bantuan.

**Tabel 5.19 Skenario Use Case lihat profil**

<i>Use case</i> Lihat profil.	
Tujuan	Melihat profil user.
Aktor	User/Admin.
Kondisi awal	Halaman <i>Home</i> .
Skenario Utama	2. User/Admin Menekan menu profil.
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman profil.

**Tabel 5.20 Skenario Use Case Edit Profil**

<i>Use case edit profil.</i>	
Tujuan	Melakukan edit profil.
Aktor	User/Admin.
Kondisi awal	Halaman Profil.
Skenario Utama	<ol style="list-style-type: none"><li>1. <i>User</i> menekan icon camera untuk mengganti foto <i>user</i></li><li>2. <i>User</i> dapat langsung <i>mengedit</i> nama, <i>email</i>, dan no telp pada <i>field</i> nama, <i>email</i>, dan no telp</li><li>3. <i>User</i> menekan tombol simpan</li></ol>
Skenario Alternatif	-
Kondisi akhir	Profil berubah.

**Tabel 5.21 Skenario Use Case Logout.**

<i>Use case logout.</i>	
Tujuan	Keluar sistem.
Aktor	User/Admin.
Kondisi awal	Halaman Profil.
Skenario Utama	<ol style="list-style-type: none"><li>1. <i>User</i> tombol logout.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Keluar dari sistem.

**Tabel 5.22 Skenario Use Case lihat daftar sayur pada gudang.**

<i>Use case lihat daftar sayur pada gudang</i>	
Tujuan	Melihat daftar sayur pada gudang
Aktor	Admin.
Kondisi awal	Halaman Home.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol menu gudang.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Menampilkan daftar sayur pada gudang

**Tabel 5.23 Skenario Use Case Menambah sayur.**

<i>Use case</i> Menambah sayur.	
Tujuan	Keluar sistem.
Aktor	Admin.
Kondisi awal	Halaman Profil.
Skenario Utama	<ol style="list-style-type: none"><li>2. Admin menekan tombol menu gudang.</li><li>3. Admin memilih sayur yang akan ditambahkan dengan menekan tombol “Add”</li><li>4. Muncul pesan “Apakah anda yakin ingin menambahkan sayur?”</li><li>5. Jika admin memilih “Iya” maka sayur ditambahkan ke halaman home.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Sayur ditambahkan ke halaman <i>home</i> .

**Tabel 5.24 Skenario Use Case Hapus Data sayur**

<i>Use case</i> hapus data sayur.	
Tujuan	Menghapus data sayur.
Aktor	Admin.
Kondisi awal	Halaman Home.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol lihat pada sayur.</li><li>2. Admin menekan tombol hapus.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Sayur berhasil dihapus.

**Tabel 5.25 Skenario Use Case Edit data sayur**

<i>Use case Edit sayur.</i>	
Tujuan	Melakukan edit data sayur.
Aktor	Admin.
Kondisi awal	Halaman Home.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol lihat pada sayur.</li><li>2. Admin menekan tombol edit.</li><li>3. Mengisi data sayur.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Sayur berhasil di-edit

**Tabel 5.26 Skenario Use Case Melihat daftar user**

<i>Use case Melihat daftar user.</i>	
Tujuan	Melihat data user.
Aktor	Admin.
Kondisi awal	Halaman Home.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol daftar user pada menu.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman daftar user.

**Tabel 5.27 Skenario Use Case Melihat data user**

<i>Use case Melihat data user.</i>	
Tujuan	Melihat detail data user.
Aktor	Admin.
Kondisi awal	Halaman daftar user.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol lihat pada daftar user.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman detail data user.

**Tabel 5.28 Skenario Use Case Hapus data user**

<i>Use case</i> Hapus data user.	
Tujuan	Menghapus data <i>user</i> .
Aktor	Admin.
Kondisi awal	Halaman daftar <i>user</i> .
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol lihat pada daftar <i>user</i>.</li><li>2. Admin menekan tombol hapus pada tampilan detail data <i>user</i>.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Data <i>user</i> berhasil dihapus.

**Tabel 5.29 Skenario Use Case Mengubah status transaksi.**

<i>Use case</i> mengubah status transaksi.	
Tujuan	Mengubah status transaksi.
Aktor	Admin.
Kondisi awal	Halaman <i>history</i> .
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin memilih daftar transaksi yang ingin diubah pada tab <i>On Progress</i>.</li><li>2. Memilih status transaksi yang tersedia pada bagian bawah halaman.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Status transaksi berhasil berubah.

**Tabel 5.30 Skenario Use Case Lihat transaksi *On progress*.**

<i>Use case</i> Lihat transaksi <i>On progress</i> .	
Tujuan	Melihat transaksi yang berstatus <i>on progress</i>
Aktor	User/Admin.
Kondisi awal	Halaman <i>history</i> .
Skenario Utama	<ol style="list-style-type: none"><li>1. Memilih tab <i>on progress</i> pada menu transaksi.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Status transaksi berhasil berubah.

**Tabel 5.31 Skenario Use Case Lihat transaksi *Completed*.**

Use case Lihat transaksi <i>Completed</i> .	
Tujuan	Melihat transaksi yang berstatus <i>Completed</i>
Aktor	User/Admin.
Kondisi awal	Halaman <i>history</i> .
Skenario Utama	1. Memilih tab <i>Completed</i> pada menu transaksi.
Skenario Alternatif	-
Kondisi akhir	Status transaksi berhasil berubah.

Tabel 5.32 – 5.44 merupakan skenario *use case* pada aplikasi web Sayur *Online*.

**Tabel 5.32 Skenario Use Case Login pada web..**

Use case login pada web.	
Tujuan	Login pada web
Aktor	Admin.
Kondisi awal	Halaman <i>Login</i> .
Skenario Utama	1. Admin memasukkan data email dan password pada form <i>login</i> dengan lengkap dan benar. 2. Admin menekan tombol <i>login</i> .
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman <i>home admin</i> .

**Tabel 5.33 Skenario Use Case Manajemen sayur.**

Use case manajemen sayur.	
Tujuan	Melakukan manajemen sayur pada web
Aktor	Admin.
Kondisi awal	Halaman Home.
Skenario Utama	1. Admin menekan menu manage sayur pada menu web.
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman manage sayur.

**Tabel 5.34 Skenario Use Case tambah sayur.**

<i>Use case</i> tambah sayur.	
Tujuan	Menambahkan sayur pada gudang.
Aktor	Admin.
Kondisi awal	Halaman manage sayur.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol tambah pada manage sayur.</li><li>2. Akan ditampilkan form untuk tambah sayur, admin mengisi data foto, nama, harga, stok dan kuantitas sayur dengan lengkap.</li><li>3. Menekan tombol tambah.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Sayur ditambahkan dan menampilkan halaman <i>manage</i> sayur.

**Tabel 5.35 Skenario Use Case lihat data sayur.**

<i>Use case</i> lihat sayur.	
Tujuan	Melihat detail data sayur pada gudang.
Aktor	Admin.
Kondisi awal	Halaman manage sayur.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin memilih sayur yang ingin dilihat pada daftar.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman detail sayur.

**Tabel 5.36 Skenario Use Case edit data sayur.**

<i>Use case</i> edit data sayur.	
Tujuan	Melakukan edit data sayur pada gudang.
Aktor	Admin.
Kondisi awal	Halaman detail sayur.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol ubah pada halaman.</li><li>2. Akan ditampilkan form untuk melakukan <i>edit</i> sayur, admin mengisi data yang ingin diubah dengan lengkap.</li><li>3. Menekan tombol ubah.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Sayur diubah dan menampilkan halaman <i>manage</i> sayur.

**Tabel 5.37 Skenario Use Case hapus data sayur.**

<i>Use case</i> hapus data sayur.	
Tujuan	Melakukan hapus data sayur pada gudang.
Aktor	Admin.
Kondisi awal	Halaman detail sayur.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol hapus pada halaman.</li><li>2. Muncul pesan “Apakah anda yakin ingin menghapus sayur?”</li><li>3. Admin menekan tombol “yakin”.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Sayur dihapus dan menampilkan halaman <i>manage</i> sayur.

**Tabel 5.38 Skenario Use Case search sayur.**

<i>Use case</i> search sayur.	
Tujuan	Mencari sayur pada gudang.
Aktor	Admin.
Kondisi awal	Halaman <i>manage</i> sayur.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin mengisikan keyword pada kolom <i>search</i> lalu menekan enter.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Menampilkan sayur yang dicari.

**Tabel 5.39 Skenario Use Case manajemen transaksi.**

<i>Use case</i> manajemen transaksi.	
Tujuan	Melakukan manajemen transaksi pada web.
Aktor	Admin.
Kondisi awal	Halaman home.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan menu <i>manage</i> transaksi pada menu web.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Menampilkan daftar transaksi.

**Tabel 5.40 Skenario Use Case lihat detail transaksi.**

<i>Use case</i> lihat detail transaksi.	
Tujuan	Melihat detail transaksi pada web.
Aktor	Admin.
Kondisi awal	Halaman <i>manage</i> transaksi.
Skenario Utama	1. Admin memilih daftar transaksi yang ditampilkan.
Skenario Alternatif	-
Kondisi akhir	Menampilkan detail transaksi.

**Tabel 5.41 Skenario Use Case manajemen pengguna.**

<i>Use case</i> manajemen pengguna.	
Tujuan	Melakukan manajemen pengguna pada web.
Aktor	Admin.
Kondisi awal	Halaman home.
Skenario Utama	1. Admin memilih menu <i>manage user</i> pada bagian menu.
Skenario Alternatif	-
Kondisi akhir	Menampilkan daftar pengguna.

**Tabel 5.42 Skenario Use Case hapus pengguna.**

<i>Use case</i> hapus pengguna.	
Tujuan	Menghapus pengguna pada web.
Aktor	Admin.
Kondisi awal	Halaman <i>manage user</i> .
Skenario Utama	1. Admin menekan tombol hapus pada daftar user yang ingin dihapus. 2. Muncul pesan “apakah anda yakin ingin menghapus pengguna?” 3. Admin memilih tombol “yakin”
Skenario Alternatif	-
Kondisi akhir	Pengguna berhasil dihapus

**Tabel 5.43 Skenario Use Case tambah pengguna.**

<i>Use case</i> tambah pengguna.	
Tujuan	Menambahkan pengguna dari web.
Aktor	Admin.
Kondisi awal	Halaman manage <i>user</i> .
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol tambah pada halaman.</li><li>2. Akan ditampilkan form untuk mengisi data pengguna, admin mengisikan data dengan lengkap.</li><li>3. Admin menekan tombol tambah.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Pengguna berhasil ditambahkan

**Tabel 5.44 Skenario Use Case logout.**

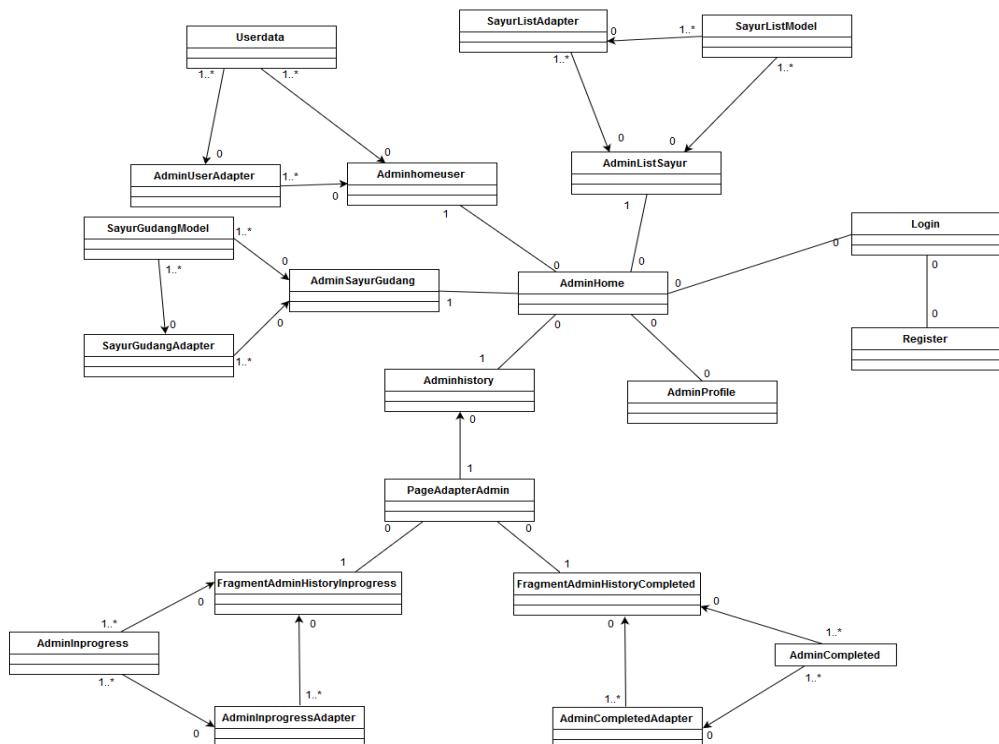
<i>Use case</i> logout.	
Tujuan	Keluar dari sistem web.
Aktor	Admin.
Kondisi awal	Halaman home.
Skenario Utama	<ol style="list-style-type: none"><li>1. Admin menekan tombol <i>hello</i> pada <i>navigation bar</i>, lalu menekan pilihan <i>logout</i>.</li></ol>
Skenario Alternatif	-
Kondisi akhir	Berhasil keluar sistem.

#### **5.2.2.4 Class Diagram**

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti komposisi, asosiasi, pewarisan dan lain-lain yang menyusun sebuah perangkat lunak. *Class diagram* dirancang dari hasil analisis terhadap *use case* yang dimodelkan sebelumnya.

## 1. Class Diagram Untuk Platform Android Pada Bagian User

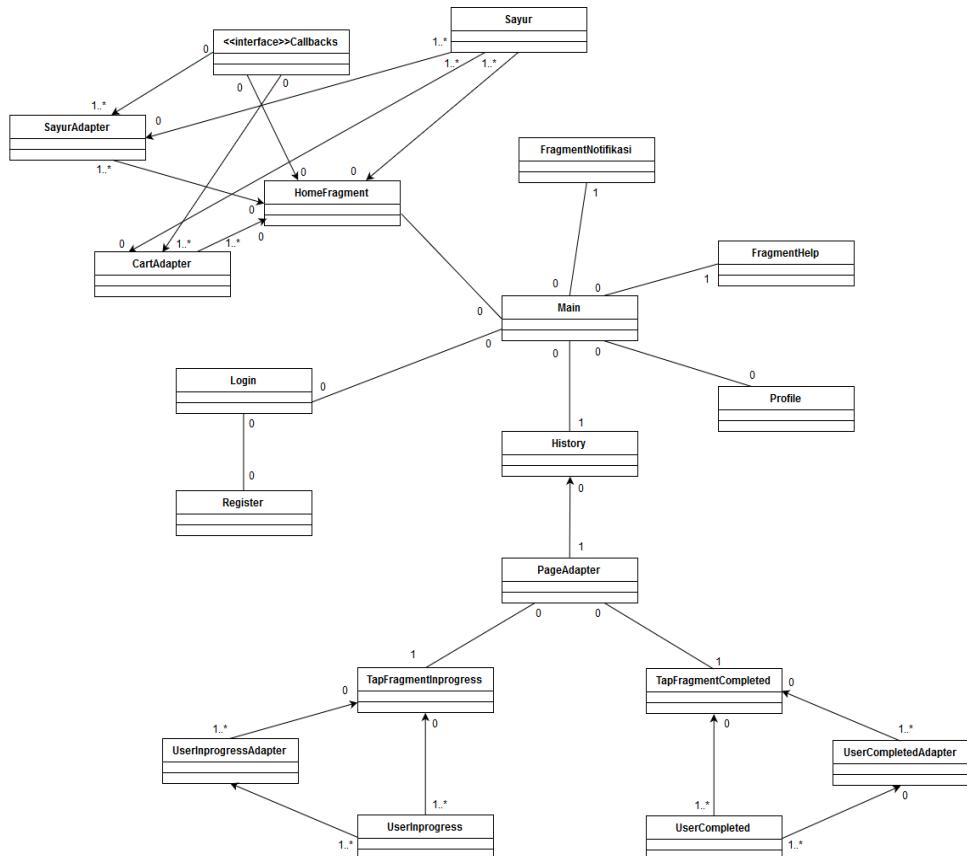
Class diagram ini merupakan penjabaran dalam bentuk kelas dari use case pada gambar 5.5. Pada *Class Diagram* ini terdapat 21 kelas yang saling berelasi dimana relasi antar kelas dapat dilihat dari tanda penghubung yang digunakan. Jika menggunakan tanda panah itu berarti kelas satu digunakan oleh kelas yang lain. Seperti contoh kelas Userdata dan Adminhomeuser memiliki relasi dengan penghubung tanda panah yang dapat dibaca kelas Userdata digunakan pada kelas Adminhomeuser. Kemudian relasi dengan tanda hubung bukan tanda panah itu berarti merupakan sebuah hubungan secara *generalisasi* antar kelas. Kemudian pada setiap kelas terdapat sebuah angka yang menunjukkan jumlah jumlah objek yang dimiliki oleh kelas pada relasi tersebut. Gambar *Class Diagram* untuk platform *android* pada bagian *user* dapat dilihat pada gambar 5.8 berikut ini.



Gambar 5.8 Class Diagram Untuk Platform Android Pada Bagian User

## 2. Class Diagram Untuk Platform Android Pada Bagian Admin

Class diagram ini merupakan penjabaran dalam bentuk kelas dari use case pada gambar 5.6. Pada *Class Diagram* ini terdapat 19 kelas yang saling berelasi dimana relasi antar kelas dapat dilihat dari tanda penghubung yang digunakan. Jika menggunakan tanda panah itu berarti kelas satu digunakan oleh kelas yang lain. Seperti contoh kelas PageAdapter dan History memiliki relasi dengan penghubung tanda panah yang dapat dibaca kelas PageAdapter digunakan pada kelas History. Kemudian relasi dengan tanda hubung bukan tanda panah itu berarti merupakan sebuah hubungan secara *generalisasi* antar kelas. Kemudian pada setiap kelas terdapat sebuah angka yang menunjukkan jumlah jumlah objek yang dimiliki oleh kelas pada relasi tersebut. Gambar *Class Diagram* untuk platform Android pada bagian admin dapat dilihat pada gambar 5.9 berikut ini.

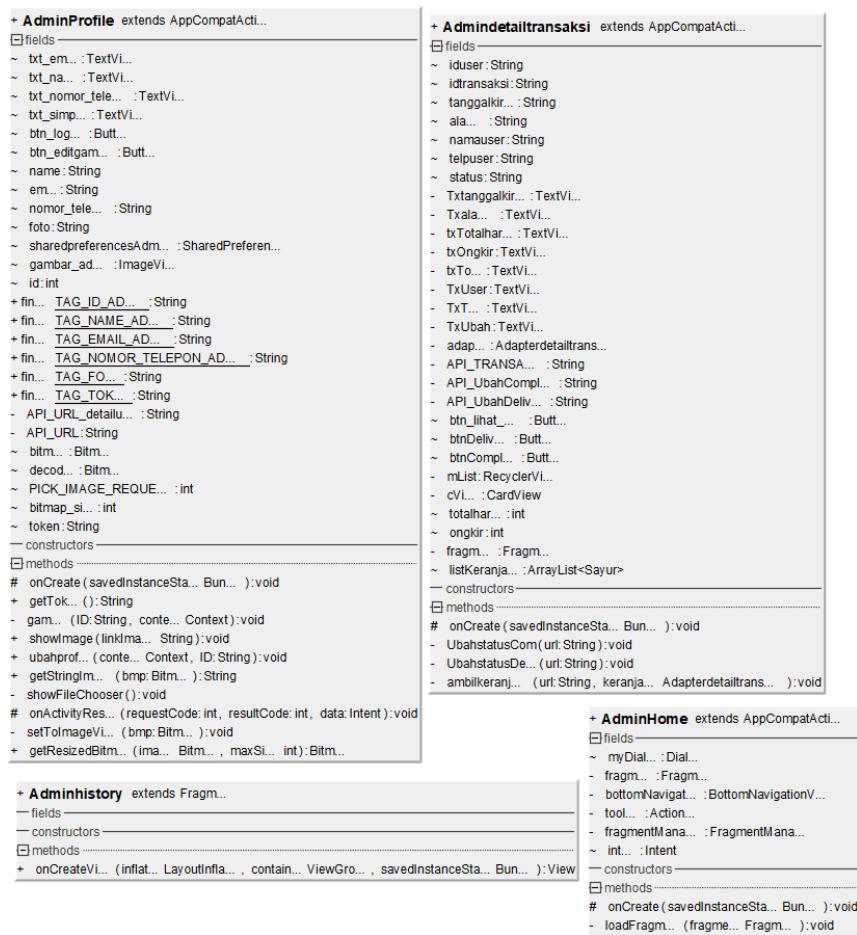


**Gambar 5.9 Class Diagram Untuk Platform Android Pada Bagian Admin**

### 3. Class Diagram Untuk Platform Android Pada Package Admin Bagian

#### 1

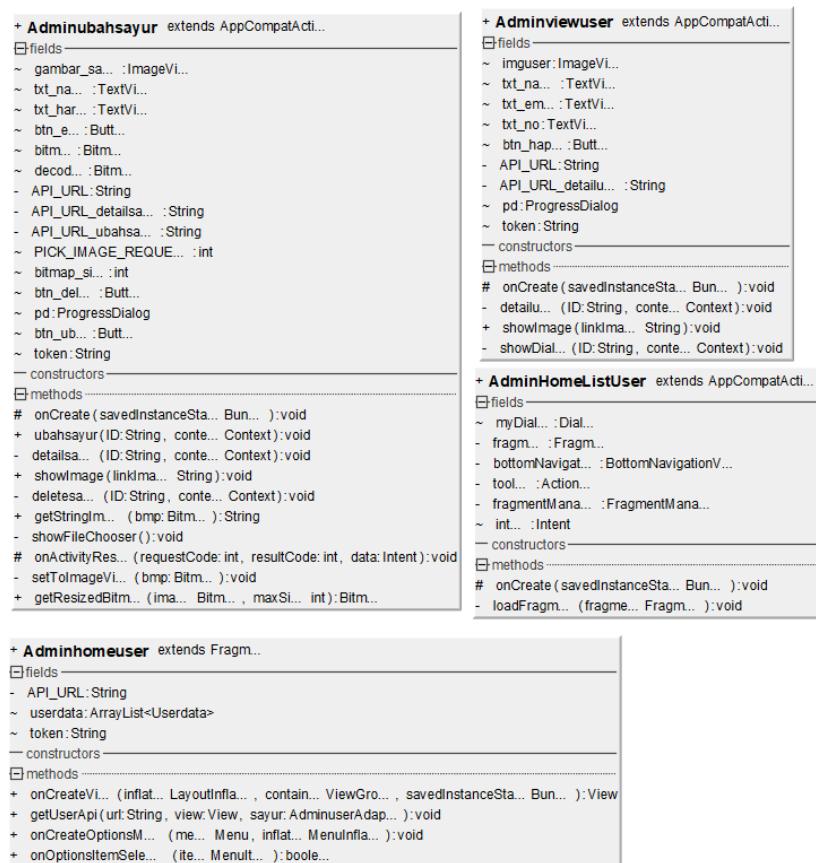
Pada Class Diagram pada package admin bagian 1 ini terdapat Class AdminProfile, Admindetailtransaksi, Adminhistory dan AdminHome. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan pada gambar 5.9. Gambar *Class Diagram* untuk platform *Android* pada *package* admin bagian 1 dapat dilihat pada gambar 5.10 berikut ini.



Gambar 5.10 *Class Diagram* Untuk Platform *Android* Pada *Package Admin Bagian 1*

#### 4. Class Diagram Untuk Platform Android Pada Package Admin Bagian 2

Pada Class Diagram pada package admin bagian 2 ini terdapat Class Adminubahsayur, Adminviewuser, AdminHomeListUser dan Adminhomeuser. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan pada gambar 5.9. Gambar *Class Diagram* untuk platform *andriod* pada *package* admin bagian 2 dapat dilihat pada gambar 5.11 berikut ini.



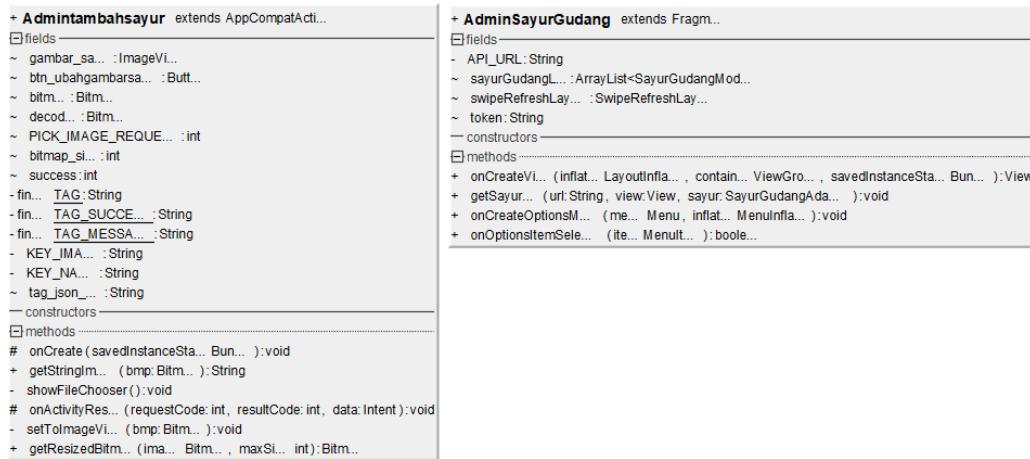
Gambar 5.11 *Class Diagram* Untuk Platform Android Pada *Package Admin Bagian 2*

#### 5. Class Diagram Untuk Platform Android Pada Package Admin Bagian 3

Pada Class Diagram pada package admin bagian 2 ini terdapat Class Admintambahsayur dan AdminSayurGudang. *Class Diagram* ini merupakan

detail *Class Diagram* dari *Class Diagram* yang ditunjukkan pada gambar

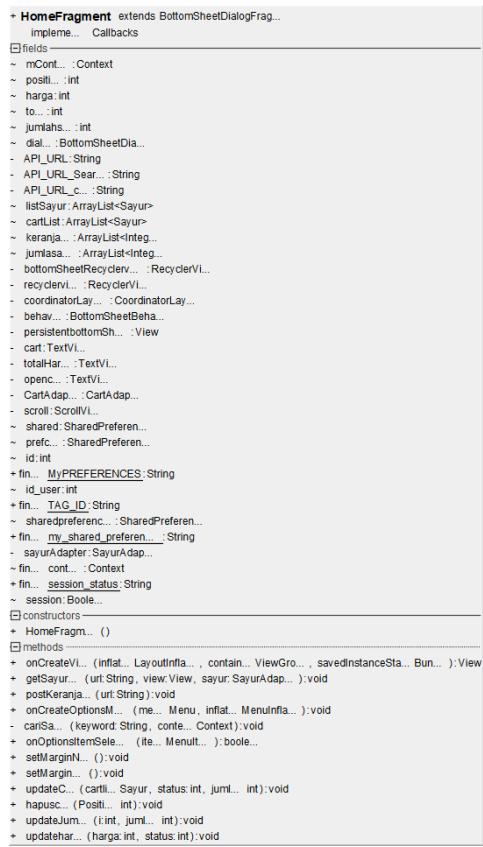
5.9. Gambar *Class Diagram* untuk platform *android* pada package admin bagian 3 dapat dilihat pada gambar 5.12 berikut ini.



**Gambar 5.12 Class Diagram Untuk Platform Android Pada Package Admin Bagian 3**

## 6. *Class Diagram* Untuk Platform *Android* Pada *Package Fragment* Bagian 1

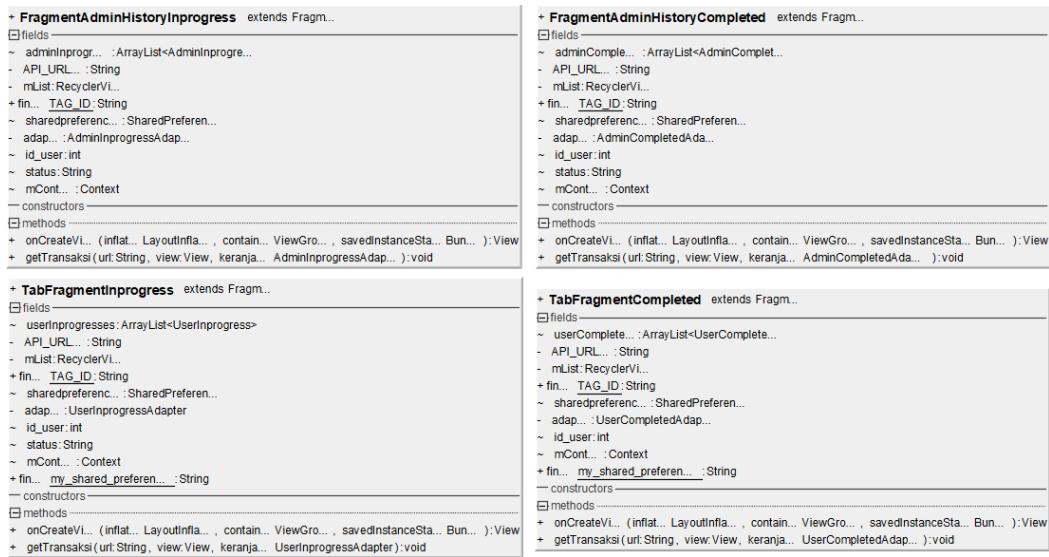
Pada *Class Diagram* pada *package fragment* bagian 1 ini terdapat *Class* HomeFragment. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan pada gambar 5.9. Gambar *Class Diagram* untuk platform *android* pada *package fragment* bagian 1 dapat dilihat pada gambar 5.13 berikut ini.



**Gambar 5.13 Detail *Class Diagram* Untuk Platform Android Pada Package Fragment bagian 1**

## 7. *Class Diagram* Untuk Platform Android Pada Package Fragment Bagian 2

Pada Class Diagram pada package fragment bagian 2 ini terdapat Class FragmentAdminHistoryInprogress, FragmentAdminHistoryCompleted, TabFragmentInprogress, TabFragmentCompleted. Untuk kelas FragmentAdminHistoryInP-rogress dan FragmentAdminHistoryCompleted merupakan detail dari *Class Diagram* yang ditunjukkan pada gambar 5.9. Sedangkan untuk kelass TabFragmentInprogress dan TabFragmentCompleted merupakan detail dari *Class Diagram* yang ditunjukkan pada gambar 5.8. Gambar *Class Diagram* untuk platform android pada package fragment bagian 2 dapat dilihat pada gambar 5.14. berikut ini.



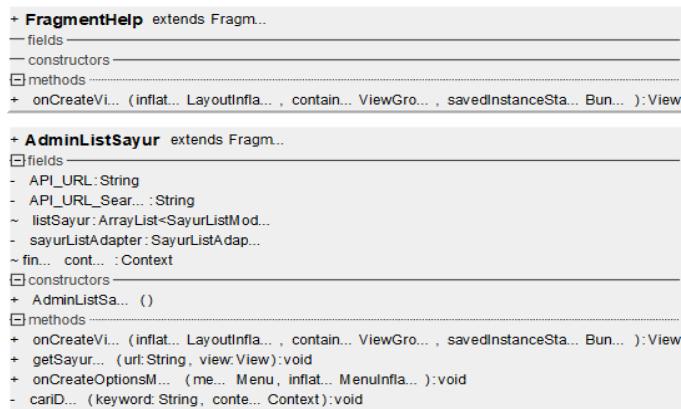
**Gambar 5.14 Detail *Class Diagram* Untuk Platform *Android* Pada Package**

### **Fragment bagian 2**

## **8. *Class Diagram* Untuk Platform *Android* Pada Package Fragment**

### **Bagian 3**

Pada *Class Diagram* pada package fragment bagian 3 ini terdapat *Class FragmentHelp*, *AdminListSayur*. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan dari pada gambar 5.8. Gambar *Class Diagram* untuk platform *android* pada package *fragment* bagian 3 dapat dilihat pada gambar 5.15 berikut ini.

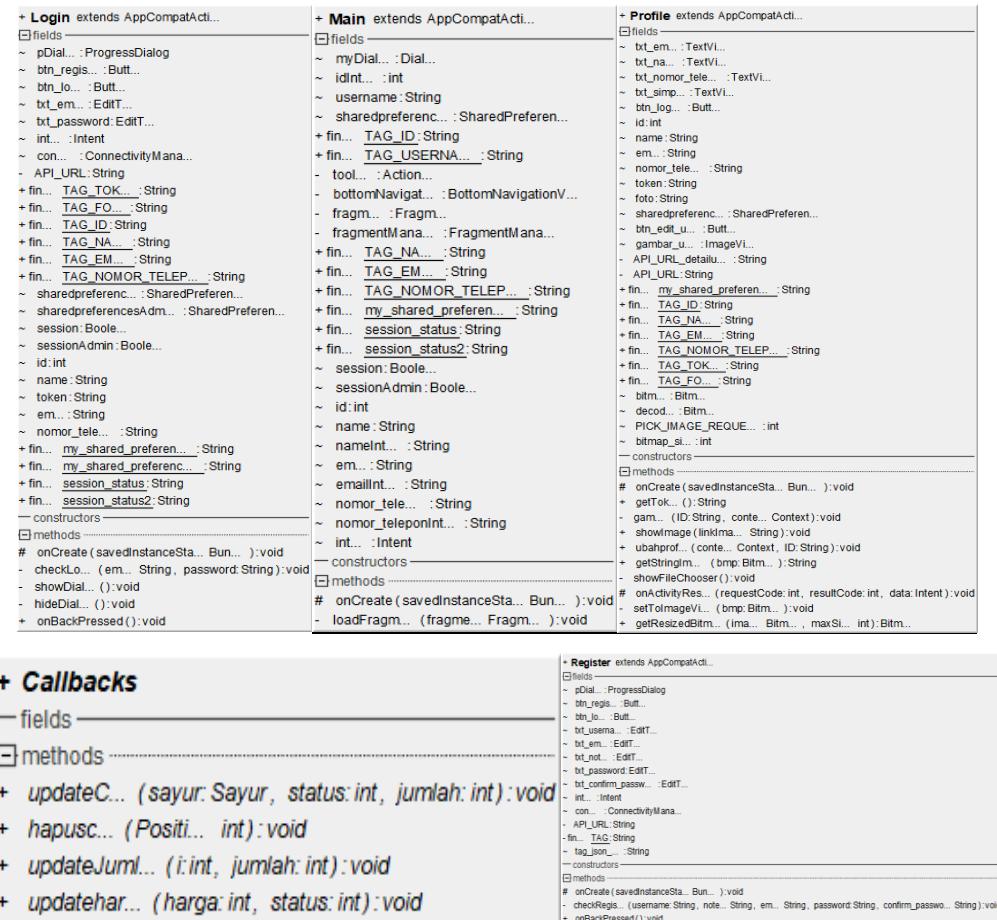


**Gambar 5.15 Detail *Class Diagram* Untuk Platform *Android* Pada Package**

### **Fragment bagian 3**

## 9. Class Diagram Untuk Platform Android Pada Package drawersayur

Pada Class Diagram pada package fragment bagian 3 ini terdapat Class Login untuk melakukan validasi User login, Main untuk bagian home, Profile untuk menampilkan profile user/admin, Interface Callbacks untuk melakukan penyimpanan array pada keranjang dan Register mendaftarkan user baru ke sistem. Untuk kelas Main dan Callbacks merupakan detail Class Diagram dari Class Diagram yang ditunjukkan pada gambar 5.8, kemudian untuk kelas Login, Profile, Register merupakan detail Class Diagram dari Class Diagram yang ditunjukkan pada gambar 5.8 dan 5.9. Gambar *Class Diagram* untuk platform android pada package drawersayur dapat dilihat pada gambar 5.16 berikut ini.

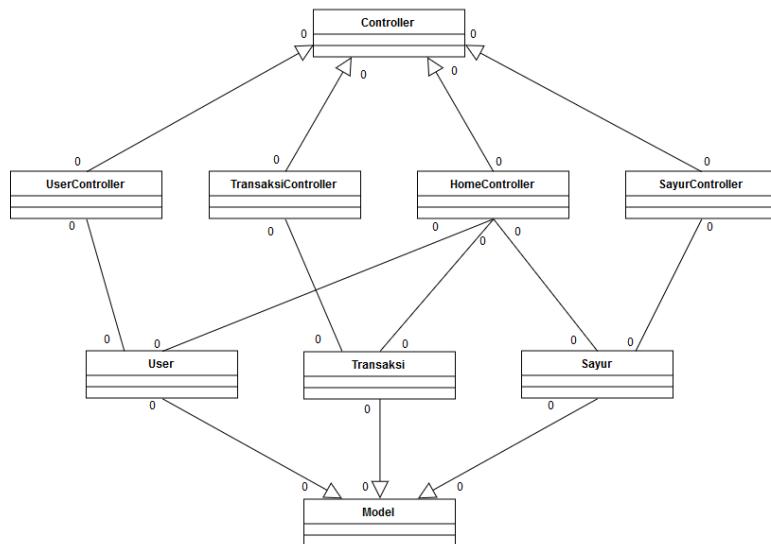


Gambar 5.16 Detail *Class Diagram* Untuk Platform Android Pada Package

*drawersayur*

## 10. Class Diagram Untuk Platform Web Untuk Proses Pada Web

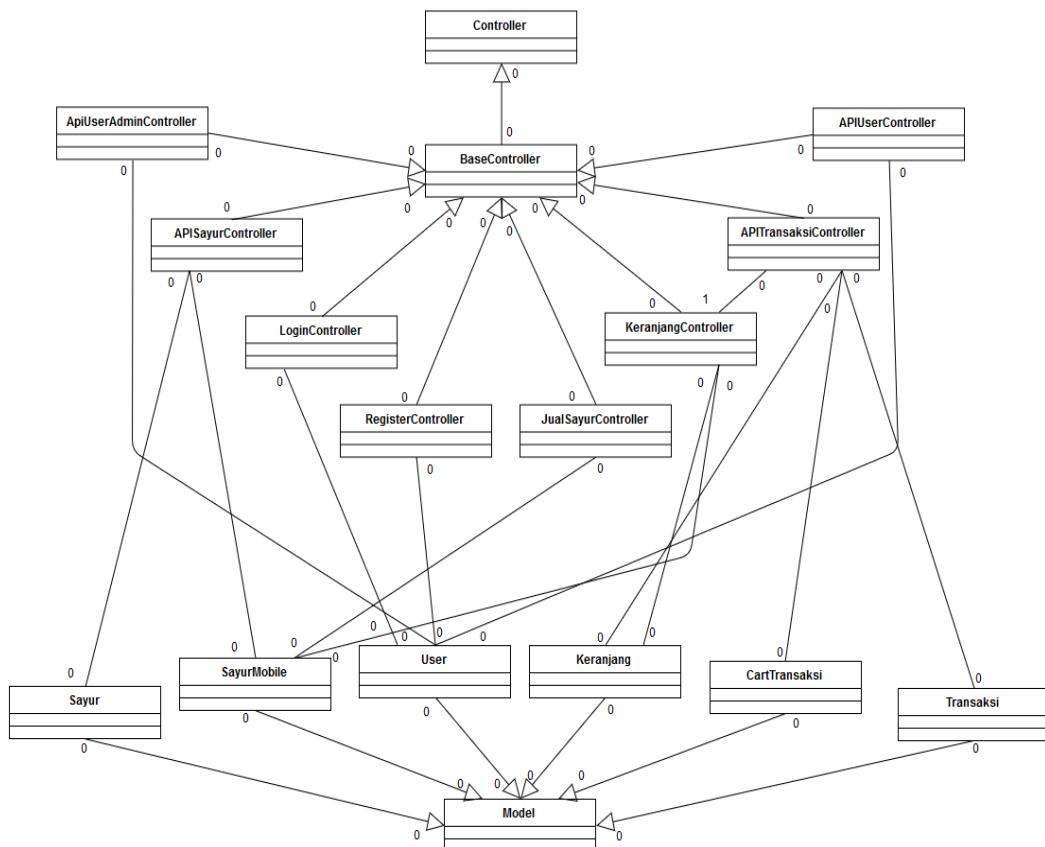
Class diagram ini merupakan penjabaran dalam bentuk kelas dari use case pada gambar 5.7. Pada *Class Diagram* ini terdapat 9 kelas yang saling berelasi dimana relasi antar kelas dapat dilihat dari tanda penghubung yang digunakan. Jika menggunakan tanda panah itu berarti kelas satu merupakan turunan dari kelas yang lain. Seperti contoh kelas Controller dan UserController memiliki relasi dengan penghubung tanda panah yang dapat dibaca kelas UserController merupakan turunan dari kelass Controller. Kemudian relasi dengan tanda hubung bukan tanda panah itu berarti merupakan sebuah hubungan secara *generalisasi* antar kelas. Kemudian pada setiap kelas terdapat sebuah angka yang menunjukkan jumlah jumlah objek yang dimiliki oleh kelas pada relasi tersebut. Gambar *Class Diagram* untuk platform web untuk proses pada web dapat dilihat pada gambar 5.17 berikut ini.



Gambar 5.17 *Class Diagram* Untuk Platform Web Untuk Proses Pada Web

## 11. Class Diagram Untuk Platform Web Untuk Proses Pada Web Service

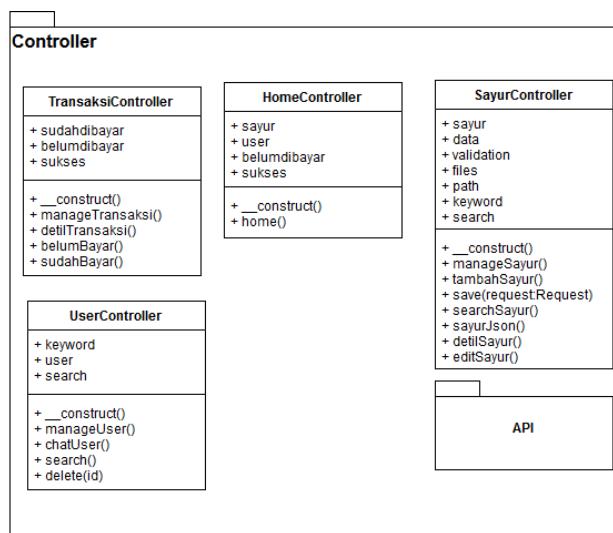
Pada *Class Diagram* ini terdapat 17 kelas yang saling berelasi dimana relasi antar kelas dapat dilihat dari tanda penghubung yang digunakan. Jika menggunakan tanda panah itu berarti kelas satu merupakan turunan dari kelas yang lain. Seperti contoh kelas Controller dan BaseController memiliki relasi dengan penghubung tanda panah yang dapat dibaca kelas BaseController merupakan turunan dari kelass Controller. Kemudian relasi dengan tanda hubung bukan tanda panah itu berarti merupakan sebuah hubungan secara *generalisasi* antar kelas. Kemudian pada setiap kelas terdapat sebuah angka yang menunjukkan jumlah jumlah objek yang dimiliki oleh kelas pada relasi tersebut. Gambar *Class Diagram* untuk platform web untuk proses pada *web service* dapat dilihat pada gambar 5.18 berikut ini.



Gambar 5.18 *Class Diagram* Untuk Platform Web Untuk Proses *Web Service*

## 12. Class Diagram Untuk Platform Web Pada Folder Controller

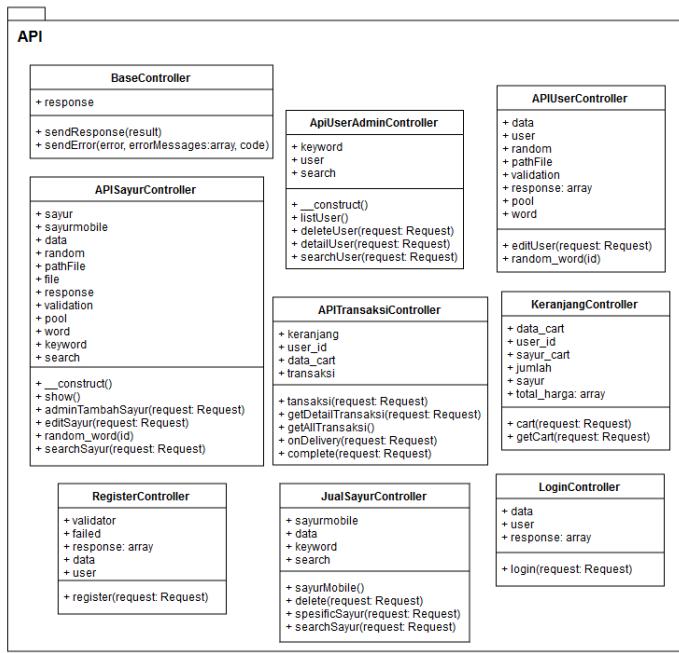
Pada Class Diagram pada folder controller ini terdapat Class TransaksiController, HomeController, SayurController, UserController, dan 1 folder API. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan dari pada gambar 5.17. Gambar *Class Diagram* untuk platform web pada folder *controller* dapat dilihat pada gambar 5.19 berikut ini.



Gambar 5.19 Detail *Class Diagram* Untuk Platform Web Pada Folder Controller

## 13. Class Diagram Untuk Platform Web Pada Folder API

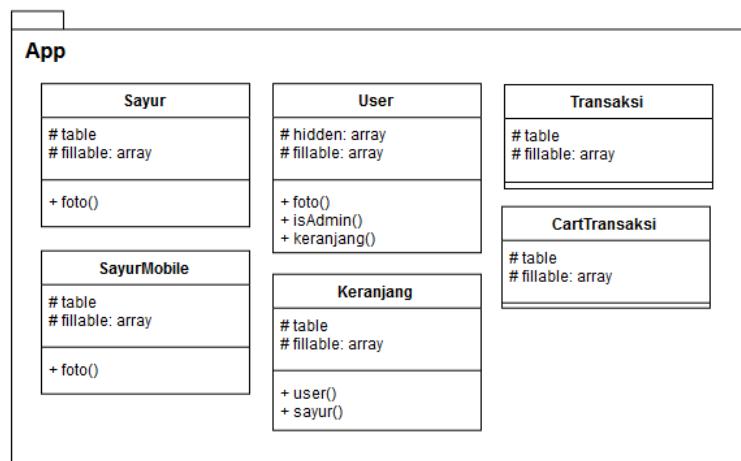
Pada Class Diagram pada folder *api* ini terdapat Class BaseController, ApiUserController, APIUserController, APISayur-Controller, APITransaksiController, KeranjangController, Register-Controller, JualSayurController dan LoginController. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan dari pada gambar 5.18. Gambar *Class Diagram* untuk platform web pada folder *api* dapat dilihat pada gambar 5.20 berikut ini.



**Gambar 5.20 Class Diagram Untuk Platform Web Pada Folder API**

#### 14. Class Diagram Untuk Platform Web Pada Folder App

Pada Class Diagram pada folder *app* ini terdapat Class Sayur, User, Transaksi, SayurMobile, Keranjang dan CartTransaksi. Untuk kelas User, Transaksi, dan Sayur merupakan detail kelas dari *Class Diagram* yang ditunjukkan pada gambar 5.17 dan 5.18. Sedangkan untuk kelas SayurMobile, Keranjang, dan CartTransaksi merupakan detail dari *Class Diagram* yang ditunjukkan pada gambar 5.18. Gambar *Class Diagram* untuk platform web pada folder *app* dapat dilihat pada gambar 5.21 berikut ini.



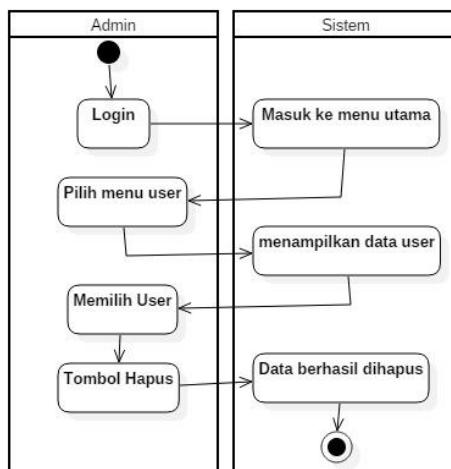
**Gambar 5.21 Class Diagram Untuk Platform Web Pada Folder App**

### 5.2.2.5 Activity Diagram

*Activity diagram* merupakan alur kerja atau kegiatan dari sebuah sistem atau menu yang ada pada perangkat lunak. *Activity diagram* juga digunakan untuk mendefinisikan urutan atau pengelompokan tampilan dari sistem/**user interface** dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan serta rancangan menu yang ditampilkan pada perangkat lunak.

#### 1. *Activity Diagram Hapus User*

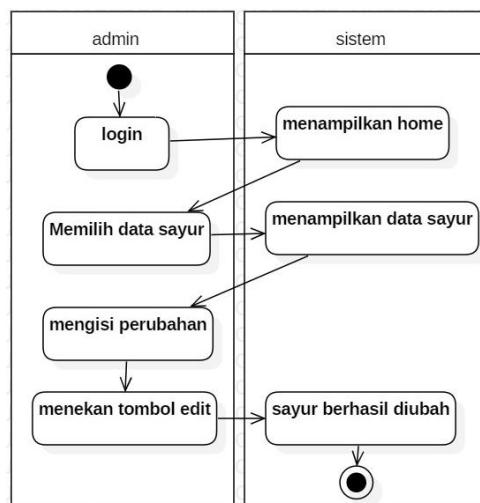
*Activity Diagram Hapus user* merupakan alur kerja program untuk menghapus data *user* yang terdaftar oleh admin. *Activity Diagram* ini merupakan penjabaran dari use case “Hapus data *user*” pada gambar 5.6. Gambar *Activity Diagram* hapus *user* dapat dilihat pada gambar 5.22 berikut ini .



**Gambar 5.22 Activity Diagram “hapus data user”**

## 2. Activity Diagram perbarui data sayur

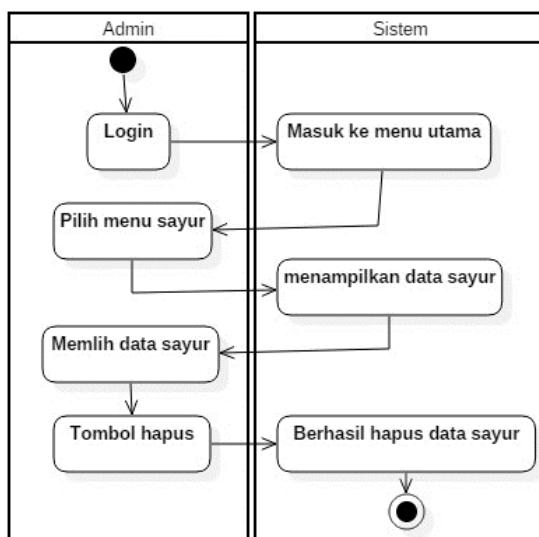
Activity Diagram perbarui data sayur merupakan alur kerja program untuk merubah/meng-edit data sayur yang dijual. Activity Diagram ini merupakan penjabaran dari use case “Perbaharui daya sayur” pada gambar 5.6. Gambar Activity Diagram perbaharui data sayur dapat dilihat pada gambar 5.23 berikut ini.



**Gambar 5.23 Activity Diagram “perbarui data sayur”**

### 3. *Activity Diagram Hapus Sayur*

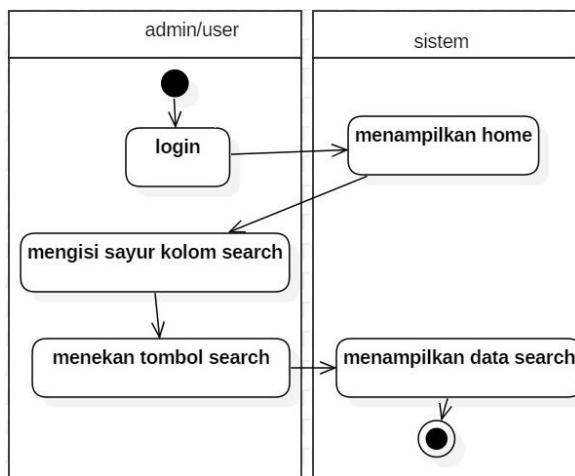
*Activity Diagram hapus sayur* merupakan alur kerja program untuk menghapus data sayur yang dijual. *Activity Diagram* ini merupakan penjabaran dari use case “Hapus data sayur” pada gambar 5.6. Gambar *Activity Diagram* hapus data sayur dapat dilihat pada gambar 5.24 berikut ini .



Gambar 5.24 *Activity Diagram* “hapus data sayur”

### 4. *Activity Diagram Search Sayur*

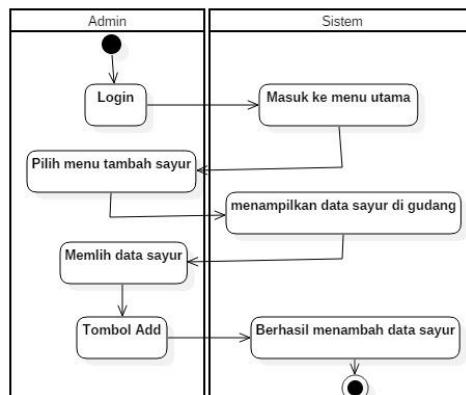
*Activity Diagram search sayur* merupakan alur kerja program untuk mencari dengan spesifik data sayur yang dijual. *Activity Diagram* ini merupakan penjabaran dari use case “Search sayur” pada gambar 5.4, 5.5,dan 5.6. Gambar *Activity Diagram* search sayur dapat dilihat pada gambar 5.25 berikut ini .



**Gambar 5.25 Activity Diagram “search sayur”**

### 5. *Activity Diagram Menambah Sayur*

*Activity Diagram* menambah sayur merupakan alur kerja program untuk menambah data sayur dari gudang *webservice*. *Activity diagram* ini merupakan penjabaran dari use case “Tambah sayur dari sayur gudang” yang terdapat pada gambar 5.6. Berikut Gambar 5.26 *activity diagram* dari menambah sayur :

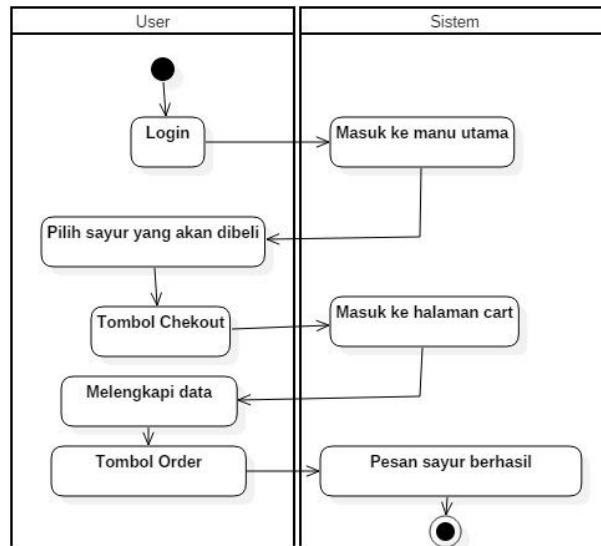


**Gambar 5.26 Activity Diagram “menambah sayur”**

### 6. *Activity Diagram Pesan Sayur*

*Activity Diagram* pesan sayur merupakan alur kerja program untuk *user*/pengguna untuk memesan sayur yang telah dijual di aplikasi sayur online. *Activity diagram* ini merupakan penjabaran dari use case

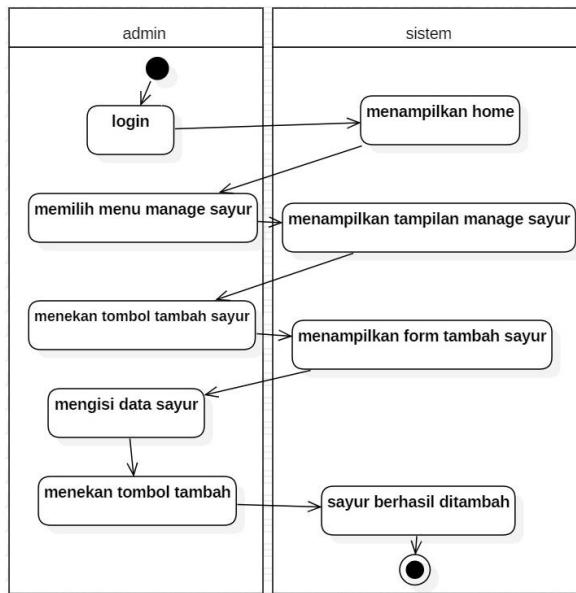
“Melakukan checkout” yang terdapat pada gambar 55. Berikut Gambar 5.27 *activity diagram* dari pesan sayur :



**Gambar 5.27 Activity Diagram “pesan sayur”**

### 7. *Activity Diagram Tambah Sayur pada web*

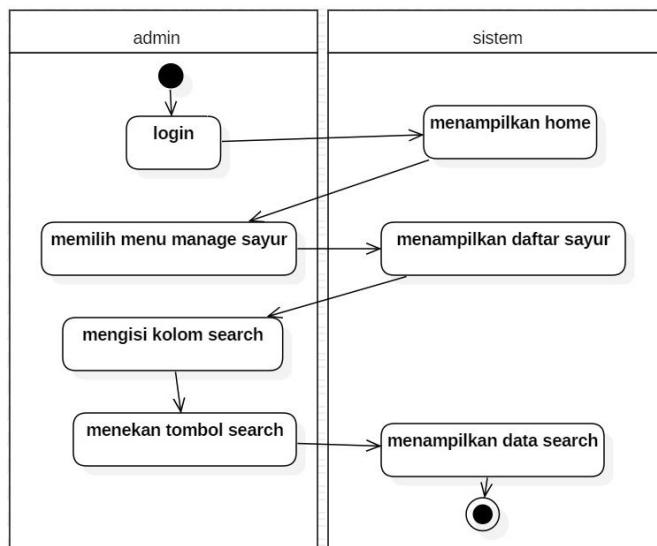
*Activity diagram* menambah sayur pada web merupakan alur dari proses menambahkan sayur pada gudang dengan web. *Activity diagram* ini merupakan penjabaran dari use case “Tambah sayur” yang terdapat pada gambar 5.7. Berikut Gambar 5.28 *activity diagram* dari tambah sayur pada web :



**Gambar 5.28 Activity Diagram “tambah sayur pada web”**

### 8. Activity Diagram Search Sayur pada web

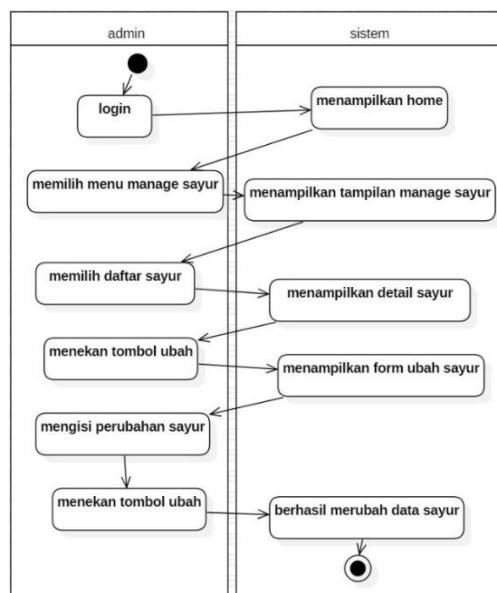
Activity diagram search sayur pada web merupakan alur dari proses mencari daftar sayur yang telah ditambahkan pada web. Activity diagram ini merupakan penjabaran dari use case “Search sayur” yang terdapat pada gambar 5.7. Berikut Gambar 5.29 activity diagram dari search sayur pada web :



**Gambar 5.29 Activity Diagram “search sayur pada web”**

### **9. Activity Diagram ubah Sayur pada web**

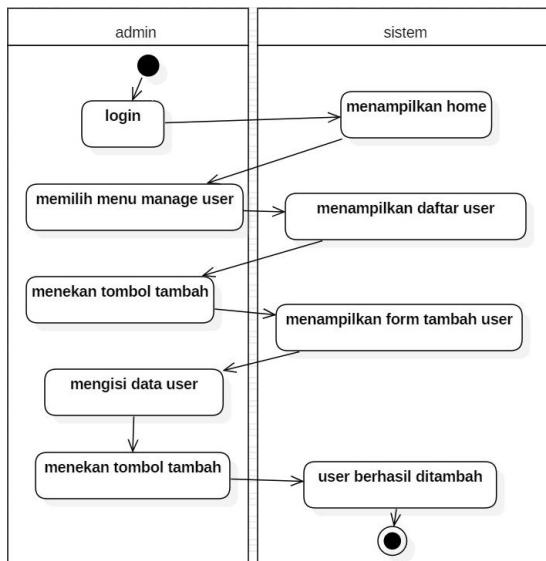
*Activity diagram* ubah sayur pada web merupakan alur dari proses melakukan perubahan pada daftar sayur yang telah ditambahkan pada web. *Activity diagram* ini merupakan penjabaran dari use case “Edit data sayur” yang terdapat pada gambar 5.7. Berikut Gambar 5.30 *activity diagram* dari ubah sayur pada web :



**Gambar 5.30 Activity Diagram “ubah sayur pada web”**

### **10. Activity Diagram tambah user pada web**

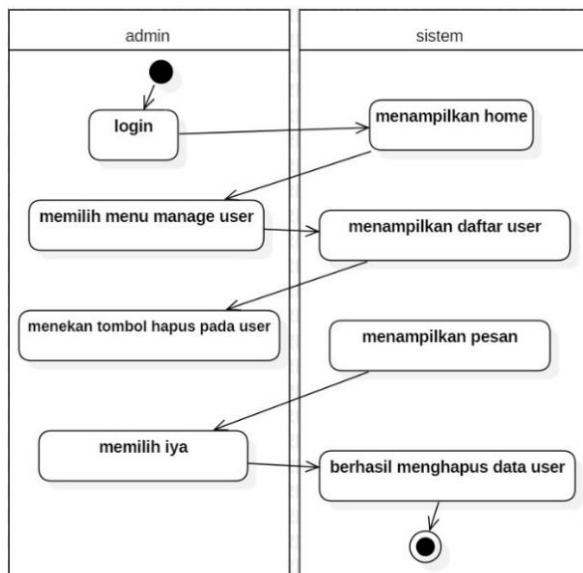
*Activity diagram* tambah user pada web merupakan alur dari proses melakukan penambahan user pada web. *Activity diagram* ini merupakan penjabaran dari use case “Tambah pengguna” yang terdapat pada gambar 5.7. Berikut Gambar 5.31 *activity diagram* dari tambah user pada web:



**Gambar 5.31 Activity Diagram “tambah user pada web”**

### 11. Activity Diagram Hapus User pada web

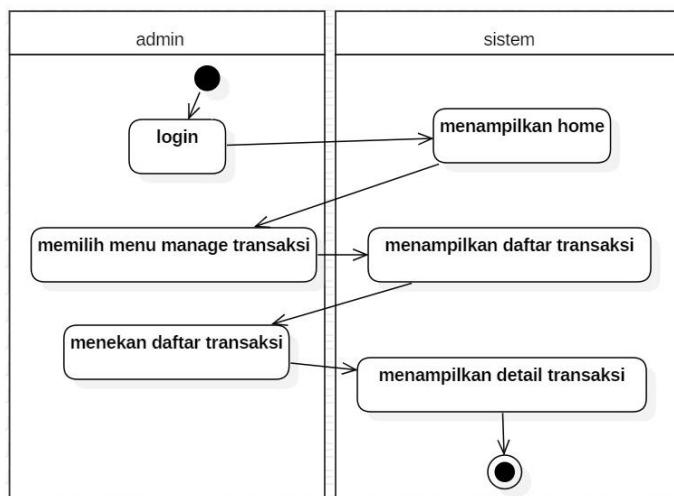
Activity diagram hapus user pada web merupakan alur dari proses melakukan penghapusan user yang telah terdaftar pada web. Activity diagram ini merupakan penjabaran dari use case “Hapus pengguna” yang terdapat pada gambar 5.7. Berikut Gambar 5.32 activity diagram dari hapus user pada web:



**Gambar 5.32 Activity Diagram “hapus user pada web”**

## 12. Activity Diagram Lihat Transaksi pada web

*Activity diagram* lihat data transaksi pada web merupakan alur dari proses melihat transaksi yang dilakukan *user* dan telah terdaftar pada *web*. *Activity diagram* ini merupakan penjabaran dari use case “Lihat detail data transaksi belum dibayar” dan “Lihat detail data transaksi sukses” yang terdapat pada gambar 5.7. Berikut Gambar 5.33 *activity diagram* dari lihat transaksi pada web:



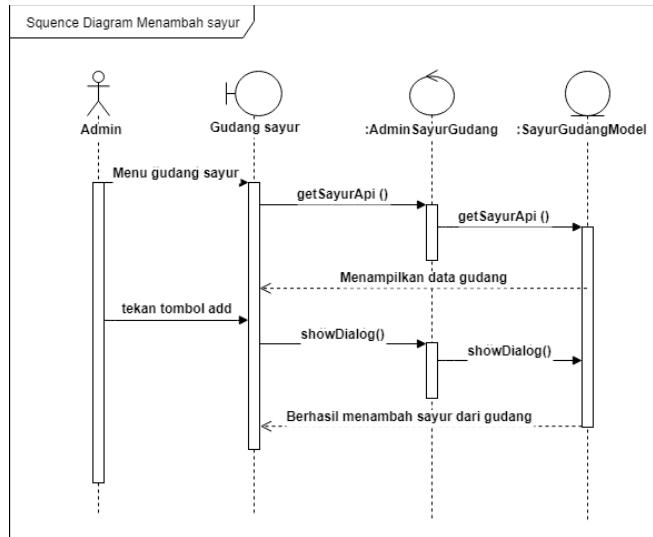
Gambar 5.33 *Activity Diagram* “lihat data transaksi pada web”

### 5.2.2.6 Sequence Diagram

*Sequence diagram* digunakan untuk menggambarkan perilaku sebuah skenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

#### 1. *Sequence Diagram* menambah sayur

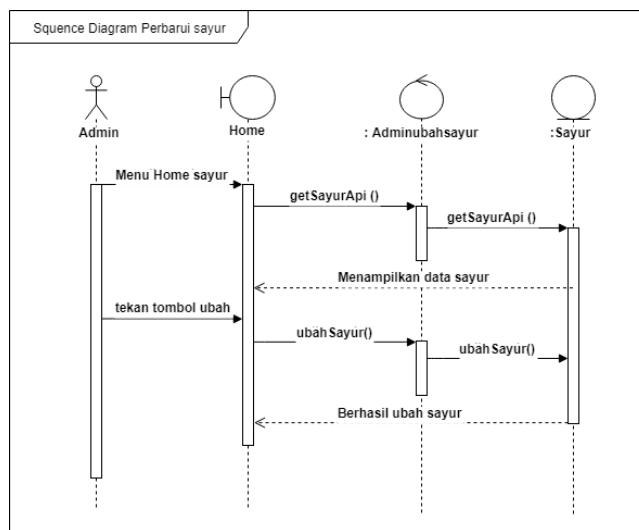
*Sequence Diagram* perbarui data sayur merupakan interaksi antara objek untuk memperbarui/meng-edit data sayur. *Sequence diagram* ini merupakan penjabaran dari use case “Tambah sayur dari gudang” yang terdapat pada gambar 5.6. Berikut Gambar 5.34 *sequence diagram* dari menambah sayur:



Gambar 5.34 Sequence Diagram “menambah sayur”

## 2. Sequence Diagram perbarui data sayur

Sequence Diagram perbarui data sayur merupakan interaksi antara objek untuk memperbarui/meng-edit data sayur. Sequence diagram ini merupakan penjabaran dari use case “Perbarui data sayur” yang terdapat pada gambar 5.6. Berikut Gambar 5.35 sequence diagram dari perbarui data sayur:

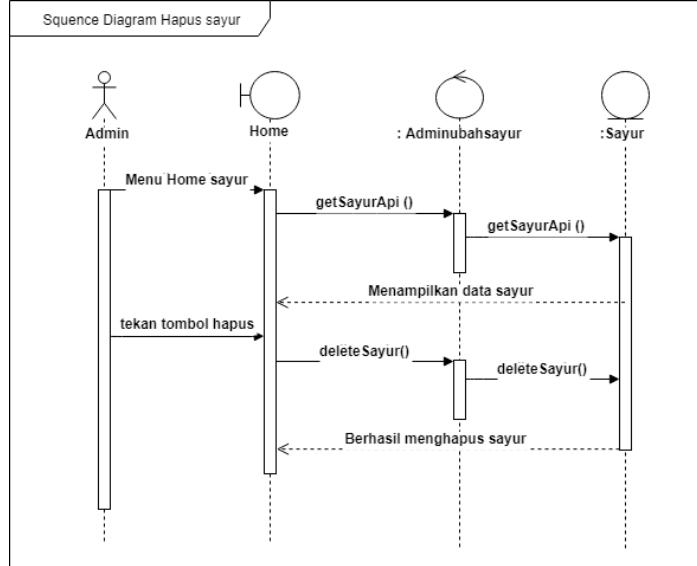


Gambar 5.35 Sequence Diagram “perbarui data sayur”

### 3. *Sequence Diagram* hapus sayur

*Sequence Diagram* hapus sayur merupakan interaksi antara objek untuk menghapus data sayur. *Sequence diagram* ini merupakan penjabaran dari use case “Hapus data sayur” yang terdapat pada gambar

5.6. Berikut Gambar 5.36 *sequence diagram* dari hapus sayur:

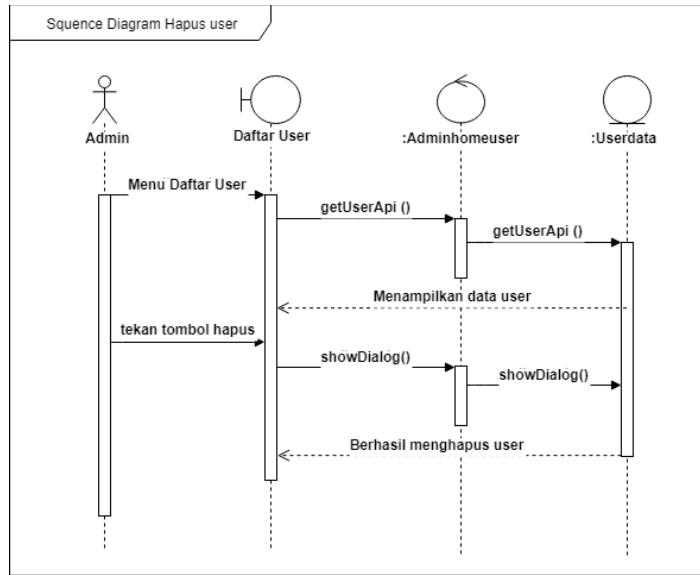


Gambar 5.36 *Sequence Diagram* “hapus sayur”

### 4. *Sequence Diagram* Hapus User

*Sequence Diagram* hapus user merupakan interaksi antara objek untuk menghapus user yang terdaftar. *Sequence diagram* ini merupakan penjabaran dari use case “Hapus data user” yang terdapat pada gambar

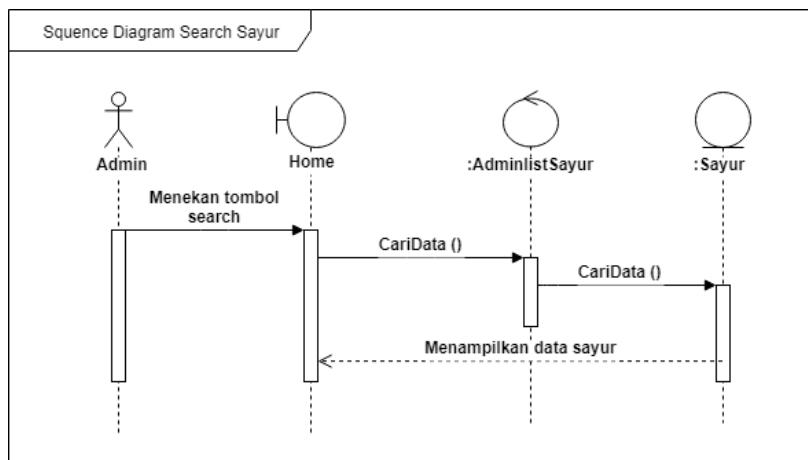
5.6. Berikut Gambar 5.37 *sequence diagram* dari hapus user:



Gambar 5.37 Sequence Diagram “hapus user”

### 5. Sequence Diagram Search Sayur

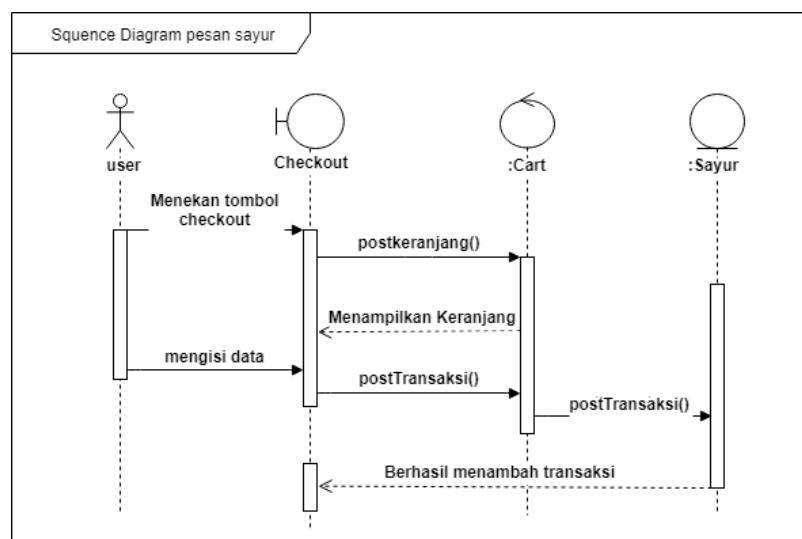
Sequence Diagram search sayur merupakan interaksi antara object untuk mencari data sayur berdasarkan nama sayur. Sequence diagram ini merupakan penjabaran dari use case “Search sayur” yang terdapat pada gambar 5.6. Berikut Gambar 5.38 sequence diagram dari search sayur:



Gambar 5.38 Sequence Diagram “search sayur”

## 6. Sequence Diagram Pesan Sayur

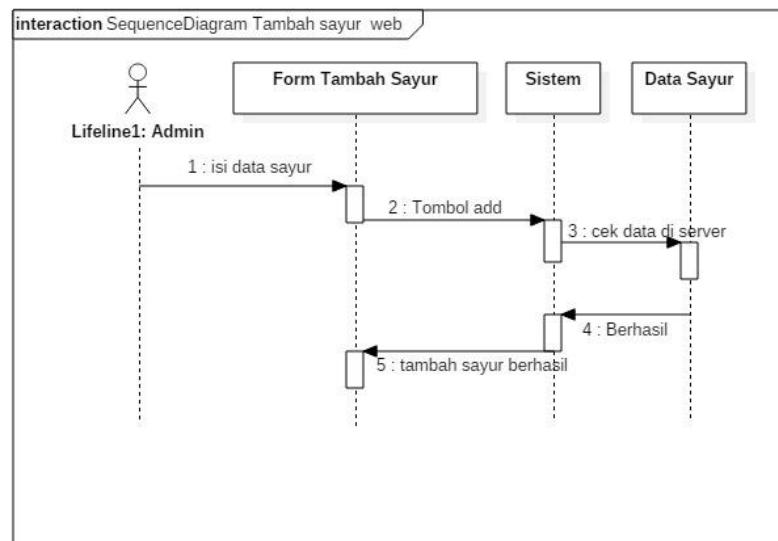
Sequence Diagram pesan sayur merupakan interaksi antara objek untuk memesan sayur yang telah dijual di sayur online. Sequence diagram ini merupakan penjabaran dari use case “Melakukan checkout” yang terdapat pada gambar 5.5. Berikut Gambar 5.39 sequence diagram dari pesan sayur:



Gambar 5.39 Sequence Diagram “pesan sayur”

## 7. Sequence Diagram Tambah Sayur Web

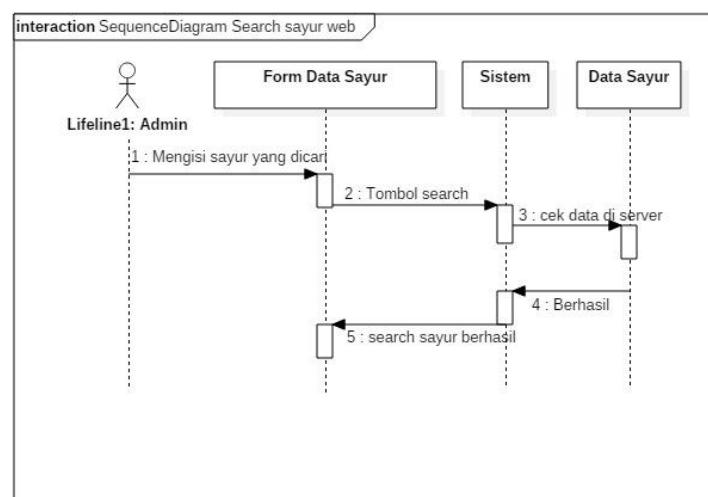
Sequence Diagram Tambah sayur web merupakan interaksi antara objek untuk menambah sayur digudang. Sequence diagram ini merupakan penjabaran dari use case “Tambah sayur” yang terdapat pada gambar 5.7. Berikut Gambar 5.40 sequence diagram dari tambah sayur web:



**Gambar 5.40 Sequence Diagram “Tambah sayur web”**

### 8. Sequence Diagram Search Sayur Web

Sequence Diagram Search sayur web merupakan interaksi antara objek untuk mencari sayur digudang. Sequence diagram ini merupakan penjabaran dari use case “Search sayur” yang terdapat pada gambar 5.7. Berikut Gambar 5.41 sequence diagram dari search sayur web:

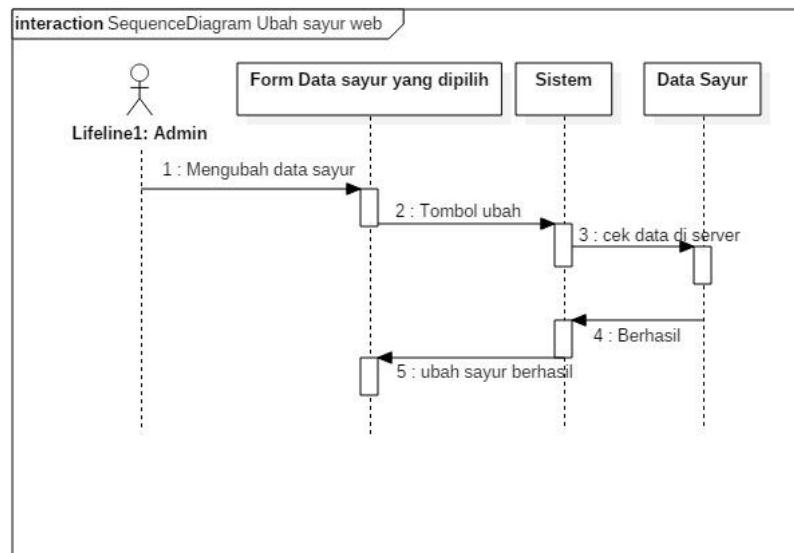


**Gambar 5.41 Sequence Diagram “Search sayur web”**

## 9. Sequence Diagram Ubah Sayur Web

Sequence Diagram Ubah sayur web merupakan interaksi antara objek untuk mengubah sayur digudang. Sequence diagram ini merupakan penjabaran dari use case “Edit data sayur” yang terdapat pada gambar 5.7.

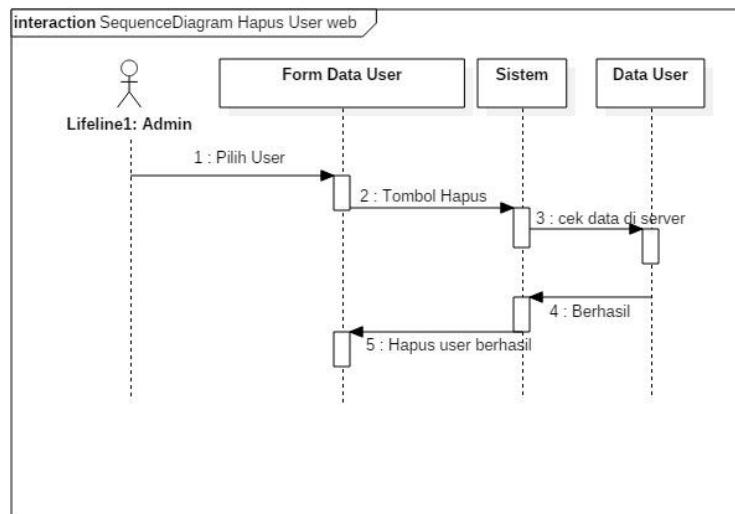
Berikut Gambar 5.42 sequence diagram dari search sayur web:



Gambar 5.42 Sequence Diagram “ubah sayur web”

## 10. Sequence Diagram Hapus User Web

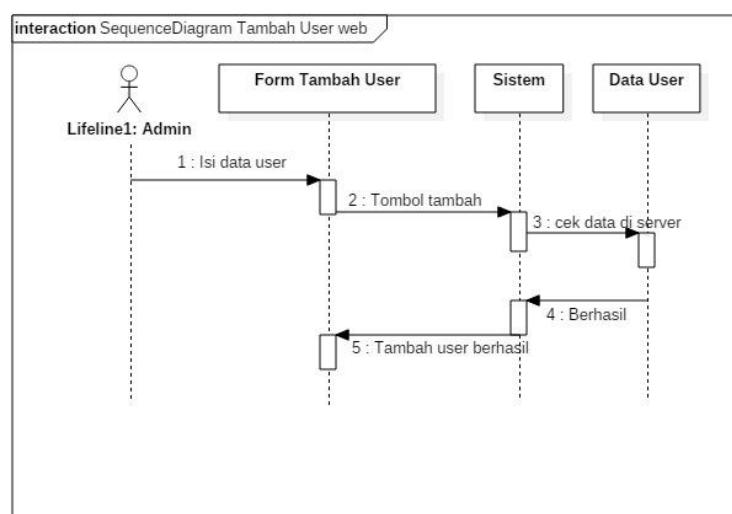
Sequence Diagram Hapus user web merupakan interaksi antara objek untuk menghapus user. Sequence diagram ini merupakan penjabaran dari use case “Hapus pengguna” yang terdapat pada gambar 5.7. Berikut Gambar 5.43 sequence diagram dari hapus user web:



**Gambar 5.43 Sequence Diagram “hapus user web”**

### 11. Sequence Diagram Tambah User Web

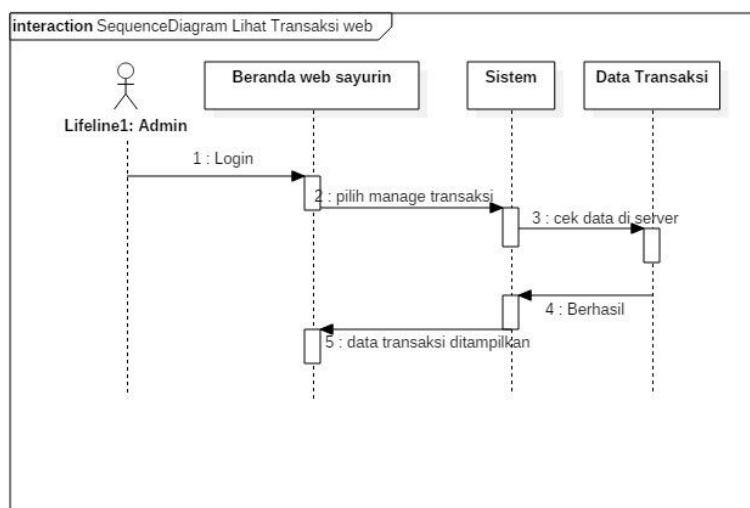
Sequence Diagram Tambah user web merupakan interaksi antara objek untuk menambah user. Sequence diagram ini merupakan penjabaran dari use case “Tambah pengguna” yang terdapat pada gambar 5.7. Berikut Gambar 5.44 sequence diagram dari tambah user web:



**Gambar 5.44 Sequence Diagram “tambah user web”**

## 12. Sequence Diagram Lihat Transaksi Web

Sequence Diagram Lihat Transaksi web merupakan interaksi antara objek untuk melihat transaksi yang dilakukan oleh *user*. Sequence diagram ini merupakan penjabaran dari use case “Menejemen transaksi” yang terdapat pada gambar 5.7. Berikut Gambar 5.45 sequence diagram dari lihat transaksi web:



Gambar 5.45 Sequence Diagram “lihat transaksi web”

## 5.3 Perancangan Perangkat Lunak

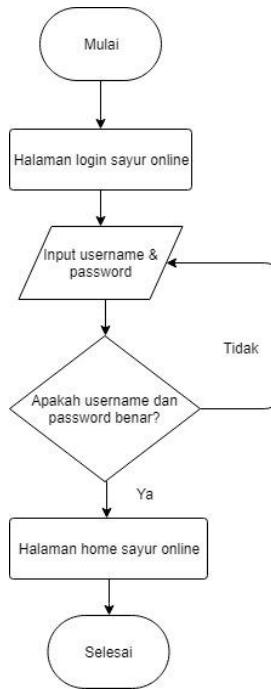
Dalam pembuatan Sistem Jual Beli Sayur *Online* ini diperlukan tahap perancangan perangkat lunak yang terdiri dari perancangan algoritma, perancangan data dan perancangan antrarmuka *user*. Masing-masing tahap tersebut dijelaskan pada bagian perancangan berikut ini.

### 5.3.1 Perancangan Algoritma

Pada perancangan algoritma ini dijelaskan mengenai alur algoritma dalam bentuk diagram alir yang merupakan salah satu cara penyajian algoritma yang lebih mudah dipahami untuk menjelaskan alur-alur proses *login*, lihat data sayur, tambah data sayur, *edit* data sayur dan hapus data sayur.

### 5.3.1.1 Diagram Alir Proses Login

Diagram alir proses login sebagai *user* ini merupakan gambaran secara visual dari algoritma login, Admin melakukan *input username* dan *password*. Diagram alir ini merupakan penjabaran dari *use case* “*Sign in*” yang terdapat pada gambar 5.4. Berikut gambar 5.46 diagram proses alir proses *login*.



Gambar 5.46 Diagram Alir Proses *Login* Sebagai Admin

### 5.3.1.2 Diagram Alir Proses Tambah Data Sayur

Diagram alir proses lihat data Sayur ini merupakan gambaran secara visual dari algoritma proses tambah data Sayur. Diagram alir ini merupakan penjabaran dari *use case* “Tambah sayur dari sayur gudang” yang terdapat pada gambar 5.6. Berikut gambar 5.47 diagram alir proses tambah data sayur.



**Gambar 5.47 Diagram Alir Proses Tambah Data Sayur**

### 5.3.1.3 Diagram Alir Proses Lihat Data Sayur

Diagram alir proses lihat data ibu hamil ini merupakan gambaran secara visual dari algoritma proses lihat data Sayur. Diagram alir ini merupakan penjabaran dari *use case* “Lihat daftar sayur pada *home*” yang terdapat pada gambar 5.4, 5.5, dan 5.6. Berikut gambar 5.48 diagram alir proses lihat data sayur.



**Gambar 5.48 Diagram Alir Proses Lihat Data Sayur**

#### 5.3.1.4 Diagram Alir Proses Perbarui Data Sayur

Diagram alir proses perbaru data sayur ini merupakan gambaran secara visual dari algoritma proses perbarui data sayur, admin melakukan perbarui data sayur pada form menejemen sayur. Diagram alir ini merupakan penjabaran dari *use case* “Perbaharui data sayur” yang terdapat pada gambar 5.6. Berikut gambar 5.49 diagram proses alir proses perbaharui data sayur.



**Gambar 5.49 Diagram Alir Proses Perbarui Data Sayur**

### 5.3.1.5 Diagram Alir Proses Hapus Data Sayur

Diagram alir proses hapus data sayur ini merupakan gambaran secara visual dari algoritma proses hapus data sayur, Admin melakukan hapus data sayur pada menu *home*. Diagram alir ini merupakan penjabaran dari *use case* “Hapus data sayur” yang terdapat pada gambar 5.6. Berikut gambar 5.50 diagram proses alir proses hapus data sayur.



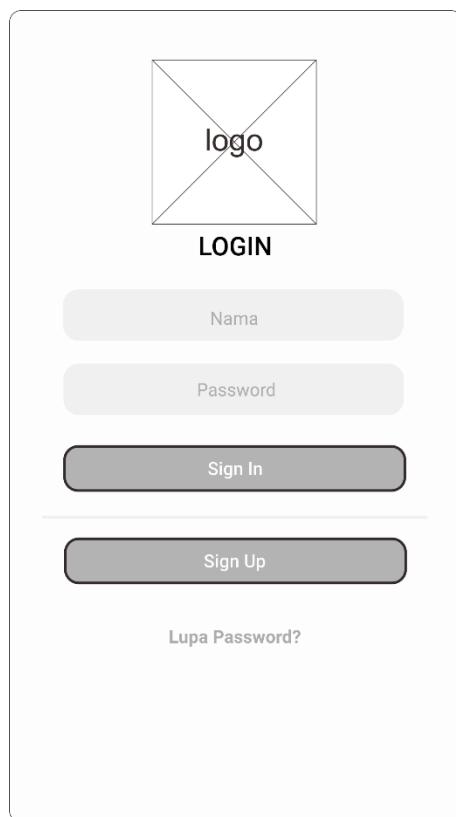
Gambar 5.50 Diagram Alir Proses Hapus Data Sayur

### **5.3.2 Perancangan Antarmuka**

Pengguna menggunakan aplikasi melalui sistem antarmuka. Perancangan antarmuka akan di terangkan pada bagian ini. Berikut ini adalah rancangan tampilan dari sistem yang akan dibuat.

#### **5.3.3.1 Perancangan Tampilan Halaman *Login***

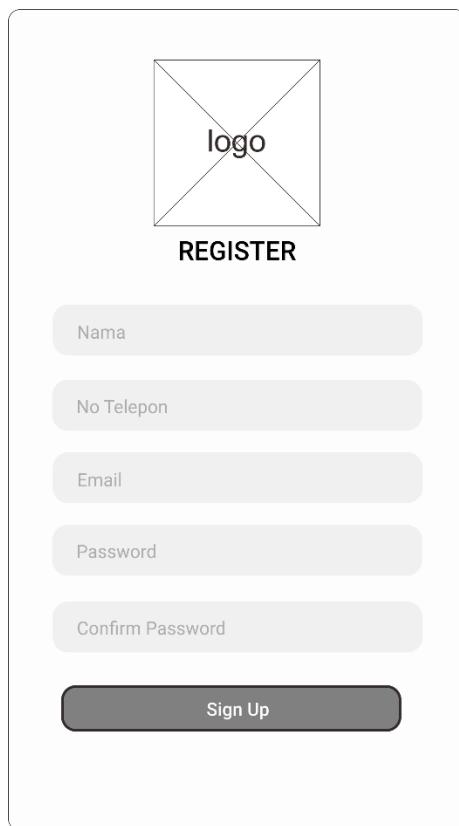
Tampilan halaman login berisi logo, judul sistem, *field email* dan *password* serta tombol *Sign In* yang digunakan *admin* dan *user* untuk masuk ke dalam sistem setelah menginputkan *email* dan *password*. Pada Gambar 5.27 berikut ini menunjukkan halaman login.



**Gambar 5.51 Perancangan Tampilan Halaman *Login***

### **5.3.3.2 Perancangan Tampilan Halaman *Register***

Tampilan halaman *register* berisi logo, judul sistem, *field* nama, no telp, email, password, konfirmasi password serta tombol *Sign Up* yang digunakan *user* untuk mendaftar ke dalam sistem setelah menginputkan *field* yang tersedia. Pada Gambar 5.28 berikut ini menunjukkan halaman *register*.

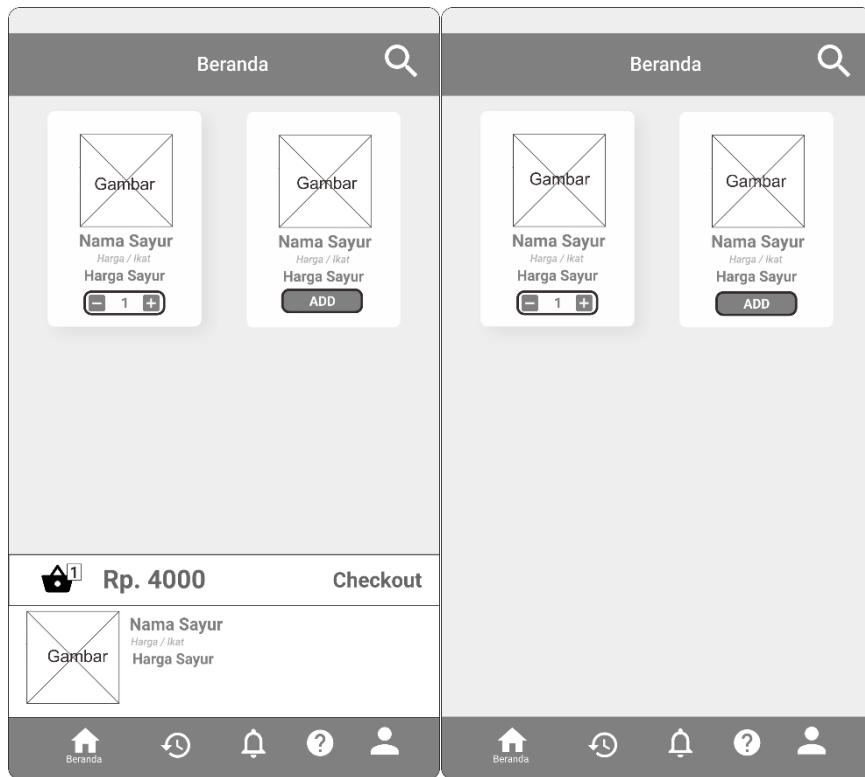


**Gambar 5.52 Perancangan Tampilan Halaman *Register***

### **5.3.3.3 Perancangan Tampilan Halaman *Home User***

Tampilan halaman *Home User* berisi tampilan yang muncul ketika *user* berhasil login. *User* dapat memilih dan melihat sayur sesuai yang dijual oleh admin, di halaman *home user* memiliki fitur *bottom sheet layout* berfungsi untuk menyimpan sayur yang akan dibeli dan menentukan jumlah sayur yang akan dibeli. Terdapat Button add digunakan untuk menyimpan sayur yang akan dibeli ke *bottom sheet layout*, dan button checkout berfungsi sebagai menyimpan sayur yang akan dibeli ke database dan berpindah halaman ke keranjang belanja untuk

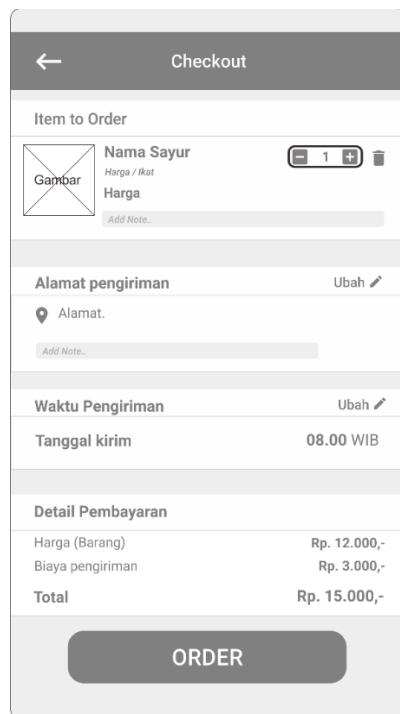
melengkapi data pengiriman. Yang terakhir terdapat tombol *search* yang berfungsi untuk mencari data sayur secara spesifik. Pada Gambar 5.29 berikut ini menunjukkan halaman *Home User*



**Gambar 5.53 Perancangan Tampilan Halaman *Home User***

#### 5.3.3.4 Perancangan Tampilan Halaman Keranjang Belanja

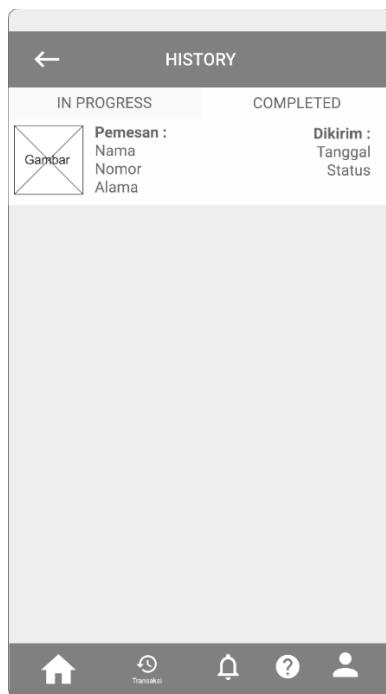
Tampilan halaman keranjang belanja berisi tampilan yang muncul ketika *user* menekan tombol *checkout*. Halaman ini akan menampilkan data sayur, total harga sayur yang akan dibeli dan untuk melengkapi data pengiriman, di halaman keranjang belanja ini terdapat *button order* untuk masuk ke proses pembelian sayur. Pada Gambar 5.30 berikut ini menunjukkan halaman Keranjang Belanja.



**Gambar 5.54 Perancangan Tampilan Halaman Keranjang Belanja**

### 5.3.3.5 Perancangan Tampilan Halaman Transaksi

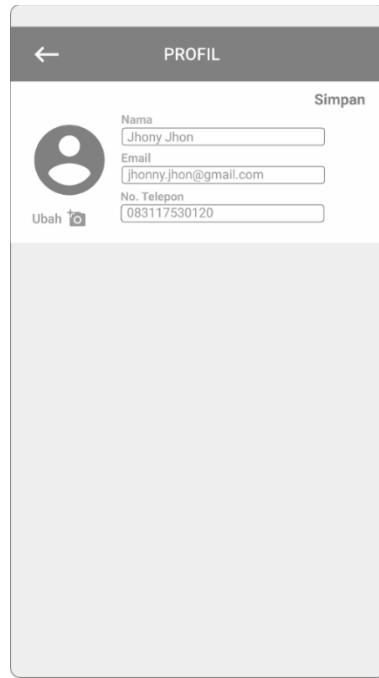
Tampilan halaman Transaksi belanja berisi tampilan yang muncul ketika *user* menekan tombol order. Halaman ini akan menampilkan data sayur, total sayur yang sebelumnya telah di order,jika barang sudah terkirim makan data akan berpindah ke tab *completed*. Pada Gambar 5.31 berikut ini menunjukkan halaman transaksi.



**Gambar 5.55 Perancangan Tampilan Halaman Transaksi**

#### **5.3.3.6 Perancangan Tampilan Halaman Profil**

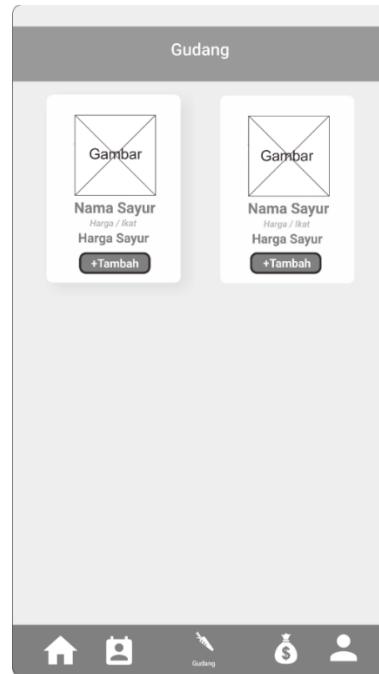
Tampilan halaman Profil muncul ketika *user* menekan tombol yang ada dipojok kanan bawah yaitu tombol *profile*. Halaman ini akan menampilkan data *user* yang berhasil login. Dihalaman ini *user* dapat merubah nama , *email* , no telepon dan foto. Pada Gambar 5.32 berikut ini menunjukkan halaman Profil.



**Gambar 5.56 Perancangan Tampilan Halaman Profil**

#### **5.3.3.7 Perancangan Tampilan Gudang Sayur**

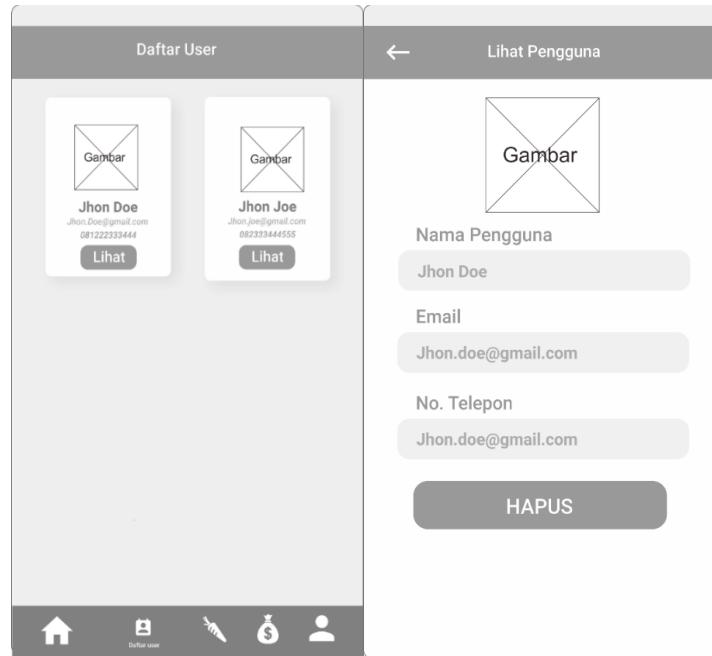
Tampilan halaman gudang sayur adalah menu ketika admin berhasil login. Halaman ini akan menampilkan data sayur yang ada digudang. Dihalaman ini terdapat tombol *add* yang berfungsi untuk menambahkan sayur yang ingin dijual dari gudang ,dan diatas juga terdapat tombol search untuk mencari sayur di gudang dengan spesifik. Pada Gambar 5.33 berikut ini menunjukkan halaman Gudang Sayur.



Gambar 5.57 Perancangan Tampilan Halaman Gudang Sayur

#### 5.3.3.8 Perancangan Tampilan *Manage User*

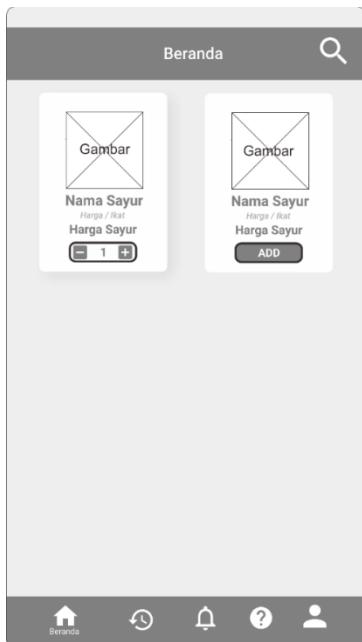
Tampilan halaman *Manage User* adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data *user* yang telah mendaftar. Dihalaman ini terdapat tombol lihat yang berfungsi untuk *memanage user*, dan diatas juga terdapat tombol search untuk mencari *user* dengan spesifik. Pada Gambar 5.34 berikut ini menunjukkan halaman *Manage User*.



**Gambar 5.58 Perancangan Tampilan Halaman *Manage User***

### 5.3.3.9 Perancangan Tampilan *Manage Sayur*

Tampilan halaman *Manage Sayur* adalah menu ketika admin berhasil login. Halaman ini akan menampilkan data sayur yang telah dijual. Dihalaman ini terdapat tombol edit yang berfungsi untuk memanage data sayur, dan diatas juga terdapat tombol search untuk mencari sayur yang dijual dengan spesifik. Pada Gambar 5.35 berikut ini menunjukkan halaman *Manage Sayur*.



**Gambar 5.59 Perancangan Tampilan Halaman Manage Sayur**

### 5.3.3.10 Perancangan Tampilan *Manage Transaksi*

Tampilan halaman *Manage Transaksi* adalah menu ketika admin berhasil login. Halaman ini akan menampilkan data sayur yang dibeli oleh *user*. Dihalaman ini admin dapat mengedit transaksi yang telah dilakukan oleh *user*. Pada Gambar 5.36 berikut ini menunjukkan halaman *Manage Transaksi*.

The wireframe illustrates the 'Detail transaksi' screen. At the top, there's a back arrow and the title 'Detail transaksi'. Below this, there are sections for 'Alamat pengiriman', 'Nama Pemesan', and 'Nomor Pemesan'. The next section is 'Item to Order', which includes a placeholder for 'Gambar' (Image) and a row for 'Nama Sayur' (Vegetable Name), showing 'Harga / ikat' (Price /束) and 'Harga' (Price). There's also a note input field labeled 'Add Note...'. Following this is a 'Alamat pengiriman' section with a location pin icon and a note input field. A 'Ubah' (Change) button with a checkmark is present. Next is a 'Waktu Pengiriman' section with a date input field for 'Tanggal kirim' (Delivery Date) set to '08.00 WIB' and a 'Ubah' button. Below these are 'Detail Pembayaran' (Payment Details) showing 'Harga (Barang)' (Item Price) as 'Rp. 12.000,-', 'Biaya pengiriman' (Delivery Cost) as 'Rp. 3.000,-', and a 'Total' of 'Rp. 15.000,-'. At the bottom, there's an 'Ubah Status' (Change Status) section with three buttons: 'Dikonfirmasi' (Confirmed), 'Menunggu Pembayaran' (Waiting for Payment), and 'Dikirim' (Shipped).

**Gambar 5.60 Perancangan Tampilan Halaman Manage Transaksi**

### 5.3.3.11 Perancangan Tampilan *Login Web*

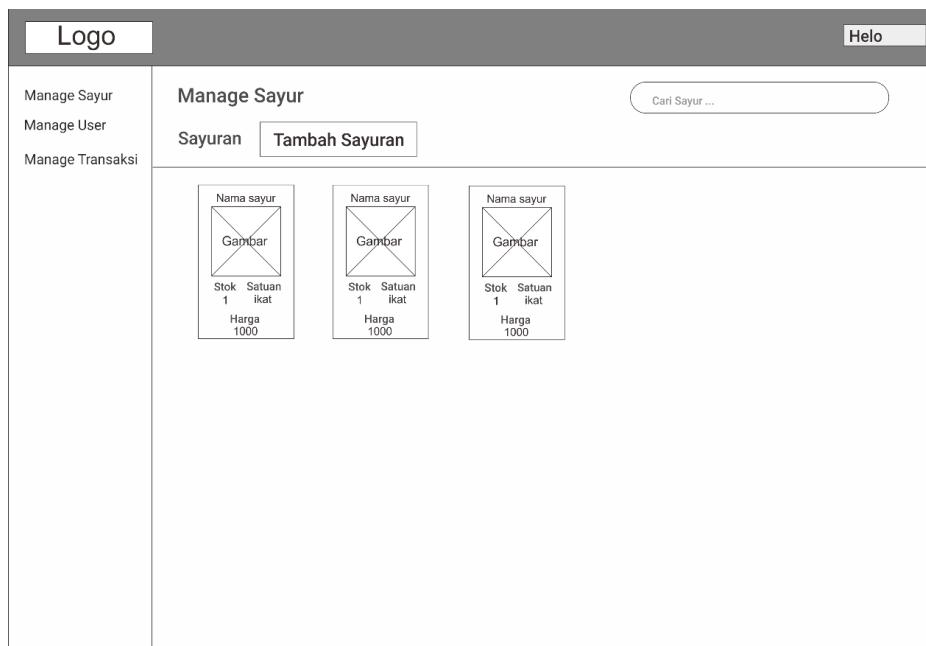
Tampilan halaman login web berisi logo, judul sistem, *field username* dan *password* serta tombol *Log In* yang digunakan *admin* untuk masuk ke dalam sistem setelah menginputkan *username* dan *password*. Pada Gambar 5.37 berikut ini menunjukkan halaman login Web.



**Gambar 5.61 Perancangan Tampilan Halaman *Login* Web**

#### 5.3.3.12 Perancangan Tampilan *Manage Sayur* Web

Tampilan halaman *Manage Sayur* web adalah menu ketika admin berhasi login. Halaman ini akan menampilkan data sayur yang telah dijual. Dihalaman ini admin dapat mengedit, menambah dan menghapus data sayur, dan disamping kanan juga terdapat tombol *search* untuk mencari sayur yang dijual dengan spesifik. Pada Gambar 5.38 berikut ini menunjukkan halaman *Manage Sayur* Web.



**Gambar 5.62 Perancangan Tampilan Halaman *Manage Sayur* Web**

### 5.3.3.13 Perancangan Tampilan *Manage Transaksi* Web

Tampilan halaman *Manage Transaksi* web adalah menu ketika admin berhasil login. Halaman ini akan menampilkan data transaksi yang dilakukan oleh user. Dihalaman ini admin dapat mengedit dan menghapus data transaksi. Pada Gambar 5.39 berikut ini menunjukkan halaman *Manage Transaksi* Web.

Manage Transaksi		
Transaksi		
Nama	Jumlah	Total Belanja
Jhon doe	4	20.000

**Gambar 5.63 Perancangan Tampilan Halaman *Manage Transaksi* Web**

#### **5.3.3.14 Perancangan Tampilan *Manage User* Web**

Tampilan halaman *Manage User* web adalah menu ketika admin berhasil login. Halaman ini akan menampilkan data *User* yang terdaftar. Dihalaman ini admin dapat menghapus data *user*, dan disamping kanan juga terdapat tombol search untuk mencari *User* yang terdaftar. Pada Gambar 5.40 berikut ini menunjukkan halaman *Manage User* Web.

Logo			HeLo
Manage Sayur Manage User Manage Transaksi	Manage Transaksi		
	Transaksi		
	Nama	Jumlah	Total Belanja
	Jhon doe	4	20.000
	Jhon doe	4	20.000
	Jhon doe	4	20.000
	Jhon doe	4	20.000
	Jhon doe	4	20.000
	Jhon doe	4	20.000

**Gambar 5.64 Perancangan Tampilan Halaman *Manage User* Web**

#### **5.3.3.14 Perancangan Tampilan *Manage User* Web**

Tampilan halaman *Manage User* web adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data *User* yang terdaftar. Dihalaman ini admin dapat menghapus data *user*, dan disamping kanan juga terdapat tombol search untuk mencari *User* yang terdaftar. Pada Gambar 5.40 berikut ini menunjukkan halaman *Manage User* Web.

Logo		Help																										
Manage Sayur Manage User Manage Transaksi	Manage Transaksi																											
	Transaksi																											
	<table border="1"> <thead> <tr> <th>Nama</th><th>Jumlah</th><th>Total Belanja</th></tr> </thead> <tbody> <tr><td>Jhon doe</td><td>4</td><td>20.000</td></tr> </tbody> </table>		Nama	Jumlah	Total Belanja	Jhon doe	4	20.000	Jhon doe	4																		
Nama	Jumlah	Total Belanja																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
<table border="1"> <thead> <tr> <th>Nama</th><th>Jumlah</th><th>Total Belanja</th></tr> </thead> <tbody> <tr><td>Jhon doe</td><td>4</td><td>20.000</td></tr> </tbody> </table>		Nama	Jumlah	Total Belanja	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000
Nama	Jumlah	Total Belanja																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
<table border="1"> <thead> <tr> <th>Nama</th><th>Jumlah</th><th>Total Belanja</th></tr> </thead> <tbody> <tr><td>Jhon doe</td><td>4</td><td>20.000</td></tr> </tbody> </table>		Nama	Jumlah	Total Belanja	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000
Nama	Jumlah	Total Belanja																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
<table border="1"> <thead> <tr> <th>Nama</th><th>Jumlah</th><th>Total Belanja</th></tr> </thead> <tbody> <tr><td>Jhon doe</td><td>4</td><td>20.000</td></tr> </tbody> </table>		Nama	Jumlah	Total Belanja	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000
Nama	Jumlah	Total Belanja																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
<table border="1"> <thead> <tr> <th>Nama</th><th>Jumlah</th><th>Total Belanja</th></tr> </thead> <tbody> <tr><td>Jhon doe</td><td>4</td><td>20.000</td></tr> </tbody> </table>		Nama	Jumlah	Total Belanja	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000
Nama	Jumlah	Total Belanja																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
<table border="1"> <thead> <tr> <th>Nama</th><th>Jumlah</th><th>Total Belanja</th></tr> </thead> <tbody> <tr><td>Jhon doe</td><td>4</td><td>20.000</td></tr> </tbody> </table>		Nama	Jumlah	Total Belanja	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000
Nama	Jumlah	Total Belanja																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
<table border="1"> <thead> <tr> <th>Nama</th><th>Jumlah</th><th>Total Belanja</th></tr> </thead> <tbody> <tr><td>Jhon doe</td><td>4</td><td>20.000</td></tr> </tbody> </table>		Nama	Jumlah	Total Belanja	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000	Jhon doe	4	20.000
Nama	Jumlah	Total Belanja																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										
Jhon doe	4	20.000																										

Gambar 5.65 Perancangan Tampilan Halaman *Manage User Web*

### 5.3.3.15 Perancangan Tampilan Detail Sayur Web

Tampilan halaman detail sayur web adalah menu ketika admin menekan sayur yang dipilih. Halaman ini akan menampilkan data sayur yang terdaftar. Dihalaman ini admin melihat detail sayur. Pada Gambar 5.41 berikut ini menunjukkan halaman Detail Sayur Web.

Logo		Help
Manage Sayur Manage User Manage Transaksi	Detail sayur	
	<div style="border: 1px solid black; height: 150px; width: 150px;"></div>	Nama <input type="text" value="Bayam"/>  Kuantitas <input type="text" value="Ikat"/> v
	Harga <input type="text" value="15.000"/>	<input type="button" value="Ubah"/>
	Stok <input type="text" value="20"/>	<input type="button" value="Hapus"/>

Gambar 5.66 Perancangan Tampilan Halaman Detail Sayur Web

### 5.3.3.16 Perancangan Tampilan Detail Transaksi Web

Tampilan halaman detail Transaksi web adalah menu ketika admin menekan transaksi yang dipilih. Halaman ini akan menampilkan data transaksi yang dilakukan oleh *user*. Dihalaman ini admin melihat dan mengedit transaksi.

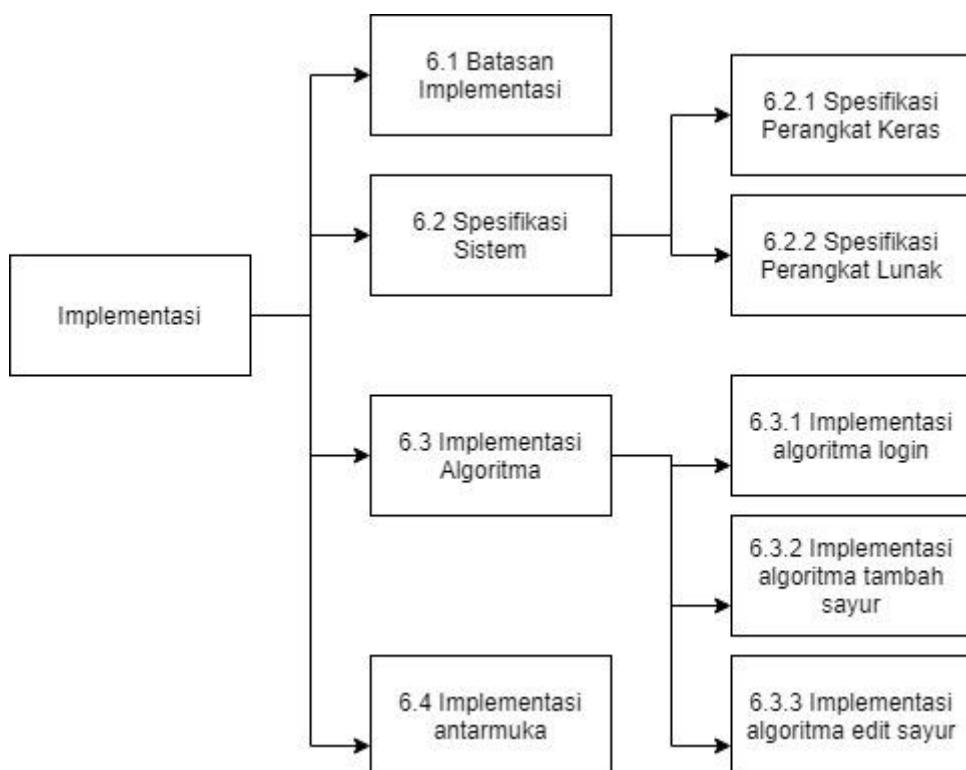
Pada Gambar 5.42 berikut ini menunjukkan halaman Detail Transaksi Web.

Logo	Hello												
<p>Manage Sayur Manage User Manage Transaksi</p>	<p><b>Detail Transaksi.</b></p> <p><b>Jhoni</b> 083111222333 Alamat kirim</p> <p><b>Tanggal Kirim</b> jhoni.email@mail.com</p> <p><b>Pesanan.</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th>Sayur</th><th>Jumlah</th><th>Total</th></tr></thead><tbody><tr><td>Bayam</td><td>3</td><td>15.000</td></tr><tr><td>Cabai</td><td>4</td><td>20.000</td></tr><tr><td></td><td><b>Total</b></td><td>35.000</td></tr></tbody></table> <p>Pembayaran : On The Spot.      Status Transaksi: Pengiriman ▾ <b>Simpan</b></p>	Sayur	Jumlah	Total	Bayam	3	15.000	Cabai	4	20.000		<b>Total</b>	35.000
Sayur	Jumlah	Total											
Bayam	3	15.000											
Cabai	4	20.000											
	<b>Total</b>	35.000											

**Gambar 5.67 Perancangan Tampilan Halaman Detail Transaksi Web**

## BAB 6 IMPLEMENTASI

Bab implementasi ini merupakan tahap pembuatan sistem yang telah dianalisa dan dirancang pada bab sebelumnya, seperti batasan implementasi, spesifikasi sistem, implementasi algoritma, dan implementasi antarmuka. Tujuan yang dicapai pada tahap ini adalah hasil perancangan sistem yang telah dibuat dapat dioperasikan. Pembahasan pada bab implementasi ditunjukkan dengan pohon implementasi seperti pada Gambar 6.1 berikut ini.



Gambar 6.1 Diagram Alir Implementasi

### 6.1 Batasan Implementasi

Batasan implementasi dalam mengimplementasikan Sistem Jual Beli Sayur Online Berbasis *Android* Menggunakan *Webservice* ini, diantaranya :

- *Input* sistem berupa database maupun data yang dilakukan input secara manual oleh *user* dan *Admin*.

- *Output* yang diterima oleh *user* dan admin berupa hasil dari database yang telah disimpan ke dalam sistem. Hasil database dapat ditampilkan berupa data.
- Parameter yang digunakan yaitu :
  1. Pesan Sayur
  2. Keranjang Belanja
  3. Data Sayur

## 6.2 Spesifikasi Sistem

Dari hasil analisa dan perancangan pada bab sebelumnya akan dijadikan acuan dalam melakukan implementasi. Pengembangan sistem dilakukan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak.

### 6.2.1 Spesifikasi Perangkat Keras

Sistem Jual Beli Sayur Online Berbasis Android Mengguanakan Webservice menggunakan komputer yang sesuai dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 6.1 berikut

**Tabel 6.1 Spesifikasi Perangkat Keras**

Nama Komponen	Spesifikasi
Processor	Intel(R) Core™ i5-4200u-1.8GHz Turbo 2.3GHz
Memori (RAM)	8.00 GB RAM
Harddisk	500 GB

### 6.2.2 Spesifikasi Perangkat Lunak

Implementasi Sistem Jual Beli Sayur Online Berbasis Android Mengguanakan Webservice menggunakan sebuah komputer yang sesuai dengan spesifikasi perangkat lunak yang dijelaskan pada Tabel 6.2 berikut.

**Tabel 6.2 Spesifikasi Perangkat Lunak**

<b>Nama Komponen</b>	<b>Spesifikasi</b>
Android Marshmello 6.0 ke atas	Sistem operasi yang digunakan untuk menjalankan Sayur Online
Visual Studio Code dan Android Studio	Sebagai editor untuk pemrograman Sayur Online
Android Emulator	Sebagai media untuk uji coba Sayur Online

### **6.3 Implementasi Algoritma**

Dalam implementasi Sistem Jual Beli Sayur Online Berbasis Android menggunakan Webservice ini memiliki beberapa proses, diantaranya tambah data sayur, lihat data sayur, dan lihat data user.

#### **6.3.1 Implementasi Algoritma Tambah Data Sayur**

Pada proses implementasi tambah data Sayur, Admin terlebih dahulu melakukan login untuk masuk ke dalam sistem. Kemudian Admin memilih menu data sayur. Sistem akan menampilkan form sayur gudang. Admin dapat memilih data yang akan ditambahkan atau dijual, jika sudah Admin dapat menyimpan data dengan mengklik tombol add. Data akan ditambahkan ke dalam database sistem pada webservice. Berikut ini tabel 6.3 merupakan algoritma kode program tambah data ibu hamil beserta penjelasannya.

**Tabel 6.3 Kode Program Tambah Sayur**

```
1 StringRequest stringRequest = new
2 StringRequest(Request.Method.POST, API_URL + "/tambah-
3 sayur", new Response.Listener<String>() {
4
5     @Override
6     public void
7     onResponse(String response) {
8
9         Intent mIntent = new
10        Intent(context,
11        AdminHome.class);
12        context.startActivity(mIntent);
13    },
14
15    Response.ErrorListener() {
16
17        @Override
18        public void
19        onErrorResponse(VolleyError error) {
20
21            error.printStackTrace();
22        }
23    } {
24
25        @Override
26        public Map<String, String>
27        getHeaders() throws AuthFailureError {
28
29            HashMap<String,
30            String> headers = new HashMap<>();
31
32            headers.put("Accept",
33            "application/json");
34
35            headers.put("Authorization", Server.TOKEN);
36
37            return headers;
38        }
39
40        @Override
41        protected Map<String,
42        String> getParams() throws AuthFailureError {
43
44            HashMap<String,
45            String> params = new HashMap<>();
46
47            params.put("id",
48            idSayur);
49
50            return params;
51        }
52    };
53
54
55    Singleton.getInstance(context).addToRequestQueue(strin-
56    gRequest);
57}
```

Penjelasan kode:

- Baris 1-3: Mengisikan *URL API* dan method yang digunakan untuk mengirimkan data
- Baris 4-6: Merupakan method yang digunakan jika data terkirim atau sukses
- Baris 7-10: Melakukan intent atau pindah halaman jika data berhasil terkirim
- Baris 11-20: Merupakan method yang digunakan jika data atau proses gagal
- Baris 21-43: Mengisikan atau membuat *header* untuk pengiriman data melalui api
- Baris 44-45: Memanggil method `addToRequestQueue` pada kelas Singleton untuk memproses data-data pengiriman data melalui api yang telah kita tentukan sebelumnya

### 6.3.2 Implementasi Algoritma Lihat Data Sayur

Pada implementasi proses lihat data sayur, admin terlebih dahulu melakukan *login* setelah itu memilih menu data sayur. Sistem akan menampilkan data sayur yang dijual yang datanya didapatkan dari *web service*. Berikut ini tabel 6.4 merupakan algoritma kode program lihat data sayur beserta penjelasannya.

**Tabel 6.4 Kode Program *Edit Data Sayur***

```
1 public void getSayurApi(String url, final View view, final
2 SayurGudangAdapter sayur) {
3     JsonArrayRequest jsonArrayRequest = new
4 JsonArrayRequest(Request.Method.GET, url, null, new
5 Response.Listener<JSONArray>() {
6     @Override
7         public void onResponse(JSONArray response) {
8             try{
9                 for (int i = 0; i < response.length();
10 i++) {
11                     JSONObject jsonObject =
12 response.getJSONObject(i);
13                     sayurGudangList.add(new
14 SayurGudangModel(jsonObject.getInt("id")
15 ,Server.URLIMAGE+jsonObject.getString("foto")
16 ,jsonObject.getString("nama"),
17 jsonObject.getInt("harga")));
18                     RecyclerView recyclerView =
19 (RecyclerView)
20 view.findViewById(R.id.recyclerviewsayurgudang);
21                     LinearLayoutManager gridLayoutManager =
22 new LinearLayoutManager(getContext(), 2);
23
24 recyclerView.setLayoutManager(gridLayoutManager);
25                     recyclerView.setAdapter(sayur);
26                 }
27             }catch (JSONException e){
28                 e.printStackTrace();
29             }
30         }
31     , new Response.ErrorListener() {
32         @Override
33             public void onErrorResponse(VolleyError error) {
34                 error.printStackTrace();
35             }
36     }) {
37         @Override
38             public Map<String, String> getHeaders() throws
39 AuthFailureError {
40                 HashMap<String, String> headers = new
41 HashMap<>();
42                 headers.put("Accept", "application/json");
43                 headers.put("Authorization", token);
44                 return headers;
45             }
46
47     };
48
49 Singleton.getInstance(this.getContext()).addToRequestQueue(
50 jsonArrayRequest); }
```

Penjelasan Kode :

- Baris 1-5: Mengisikan *URL API* dan *method* yang digunakan untuk mengambil data
- Baris 6-7: Merupakan *method* yang digunakan jika proses mengambil data sukses
- Baris 8-26: Melakukan *looping* untuk menyimpan data yang diterima pada *web service* pada array list dan data tersebut akan ditampilkan melalui *recyclerview*
- Baris 27-30: Merupakan proses yang akan dilakukan jika pada proses try sebelumnya terdapat error
- Baris 31-36: Merupakan proses yang akan dijalankan jika pada saat pengambilan data pada *web service* terjadi *error*
- Baris 37-48: Mengisikan atau membuat *header* untuk mengambil data melalui *api*
- Baris 49-50: Memanggil method *addToRequestQueue* pada kelas *Singleton* untuk memproses data-data pengiriman data melalui *api* yang telah kita tentukan sebelumnya

### 6.3.3 Implementasi Algoritma Lihat Data User

Pada implementasi proses lihat data user, admin terlebih dahulu melakukan *login* setelah itu memilih menu data user. Sistem akan menampilkan data user yang didapatkan melalui *webservice*. Berikut ini tabel 6.4 merupakan algoritma kode program lihat data user beserta penjelasannya.

**Tabel 6.5 Kode Program Lihat Data User**

```
1 public void getUserApi(String url, final View view, final
2 AdminuserAdapter sayur){
3     JsonArrayRequest jsonArrayRequest = new
4 JsonArrayRequest(Request.Method.GET, url, null, new
5 Response.Listener<JSONArray>() {
6         @Override
7         public void onResponse(JSONArray response) {
8             try{
9                 for (int i = 0; i < response.length(); i++) {
10                     JSONObject jsonObject = response.getJSONObject(i);
11
12                     if(jsonObject.getString("tipe").equals("user")){
13                         userdata.add(new
14 Userdata(jsonObject.getInt("id"),Server.URLIMAGE+jsonObject
15 .getString("foto"),
16 ,jsonObject.getString("name"),
17 ,jsonObject.getString("email"),
18 jsonObject.getString("nomor_telepon")));
19                         RecyclerView recyclerView =
20 (RecyclerView)
21 view.findViewById(R.id.recyclerviewadminhomeuser);
22                         LinearLayoutManager
23 layoutManager = new LinearLayoutManager(getApplicationContext(), 2);
24
25                         recyclerView.setLayoutManager(layoutManager);
26                         recyclerView.setAdapter(sayur);
27                     }
28                 }
29             } catch (JSONException e){
30                 e.printStackTrace();
31             }
32         }
33     }, new Response.ErrorListener() {
34         @Override
35         public void onErrorResponse(VolleyError error) {
36             error.printStackTrace();
37         }
38     });
39     @Override
40     public Map<String, String> getHeaders() throws
41 AuthFailureError {
42         HashMap<String, String> headers = new
43 HashMap<>();
44         headers.put("Accept", "application/json");
45         headers.put("Authorization", token);
46         return headers;
47     }
48 }
49 };
50
51 Singleton.getInstance(this.getApplicationContext()).addToRequestQueue(
52 jsonArrayRequest); }
```

Penjelasan Kode :

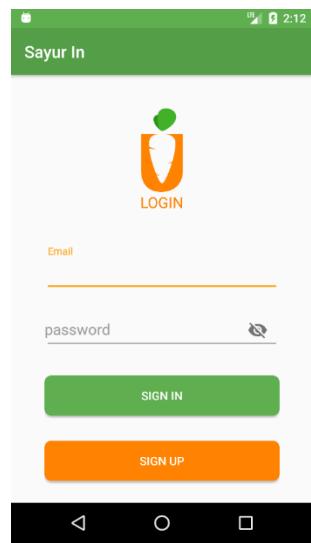
- Baris 1-5: Mengisikan *URL API* dan *method* yang digunakan untuk mengambil data
- Baris 6-7: Merupakan *method* yang digunakan jika proses mengambil data sukses
- Baris 8-29: Melakukan *looping* untuk menyimpan data yang diterima pada *web service* pada array list dan data tersebut akan ditampilkan melalui *recyclerview*
- Baris 30-33: Merupakan proses yang akan dilakukan jika pada proses *try* sebelumnya terdapat error
- Baris 34-39: Merupakan proses yang akan dijalankan jika pada saat pengambilan data pada *web service* terjadi *error*
- Baris 40-50: Mengisikan atau membuat *header* untuk mengambil data melalui *api*
- Baris 51-52: Memanggil method *addToRequestQueue* pada kelas *Singleton* untuk memproses data-data pengiriman data melalui *api* yang telah kita tentukan sebelumnya

## 6.4 Antarmuka

Implementasi antarmuka pada Sistem Jual Beli Sayur Online Berbasis Android Mengguanakan Webservice ini digunakan sebagai perantara antara pengguna dan sistem untuk berinteraksi. Berikut merupakan hasil implementasi antarmuka dari Sistem Jual Beli Sayur Online Berbasis Android Mengguanakan Webservice.

### a. Implementasi Halaman *Login*

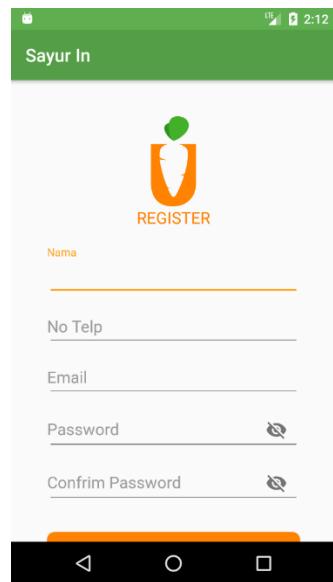
Tampilan halaman login sebagai *user* berisi logo, judul sistem, *field email* dan *password* serta tombol *Sign In* yang digunakan *admin* dan *user* untuk masuk ke dalam sistem setelah menginputkan *email* dan *password*. Pada Gambar 6.2 berikut ini menunjukkan halaman login.



**Gambar 6.2 Implementasi Halaman Login**

**b. Implementasi Halaman Register**

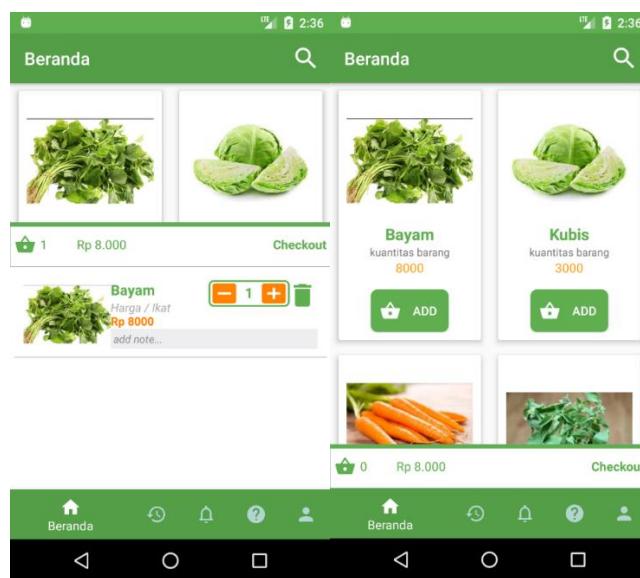
Tampilan halaman register berisi logo, judul sistem, *field* nama, no telp, email, password, konfirmasi password serta tombol *Sign Up* yang digunakan *user* untuk mendaftar ke dalam sistem setelah menginputkan *field* yang tersedia. Pada Gambar 6.3 berikut ini menunjukkan halaman *register*.



**Gambar 6.3 Implementasi Halaman Register**

### c. Implementasi Halaman *Home User*

Tampilan halaman *Home User* berisi tampilan yang muncul ketika *user* berhasil login. *User* dapat memilih dan melihat sayur sesuai yang dijual oleh admin, di halaman home *user* memiliki fitur *bottom sheet layout* berfungsi untuk menyimpan sayur yang akan dibeli dan menentukan jumlah sayur yang akan dibeli. Terdapat Button add digunakan untuk menyimpan sayur yang akan dibeli ke bottom sheet layout, dan button checkout berfungsi sebagai menyimpan sayur yang akan dibeli ke database dan berpindah halaman ke keranjang belanja untuk melengkapi data pengiriman. Yang terakhir terdapat tombol search yang berfungsi untuk mencari data sayur secara spesifik. Pada Gambar 6.4 berikut ini menunjukkan halaman *Home User*.

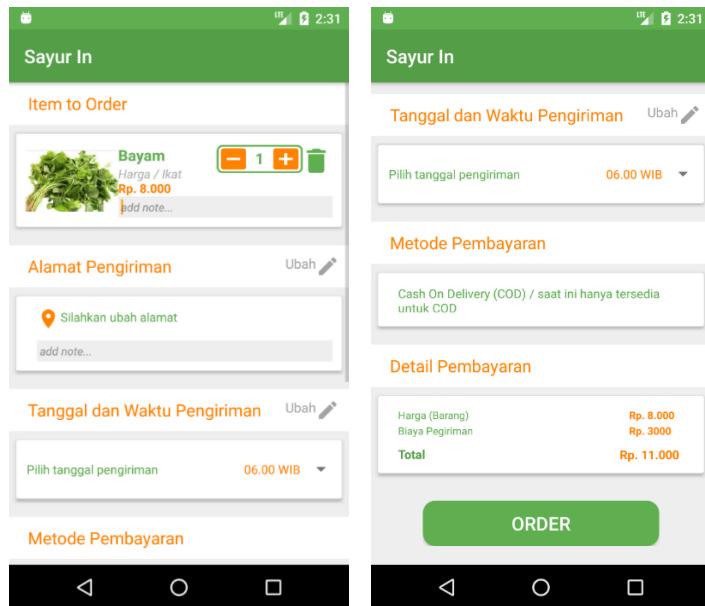


Gambar 6.4 Implementasi Halaman *Home User*

### d. Implementasi Halaman Keranjang Belanja

Tampilan halaman keranjang belanja berisi tampilan yang muncul ketika *user* menekan tombol checkout. Halaman ini akan menampilkan data sayur, total harga sayur yang akan dibeli dan untuk melengkapi data pengiriman, di halaman keranjang belanja ini terdapat button *order* untuk masuk ke proses

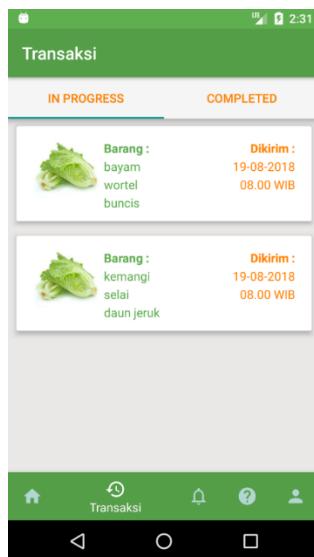
pembelian sayur. Pada Gambar 6.5 berikut ini menunjukkan halaman Keranjang Belanja.



Gambar 6.5 Implementasi Halaman Keranjang Belanja

#### e. Implementasi Halaman Transaksi

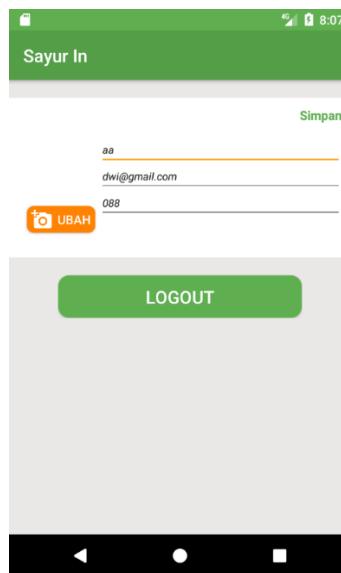
Tampilan halaman Transaksi belanja berisi tampilan yang muncul ketika *user* menekan tombol order. Halaman ini akan menampilkan data sayur, total sayur yang sebelumnya telah di order, jika barang sudah terkirim maka data akan berpindah ke tab *completed*. Pada Gambar 6.6 berikut ini menunjukkan halaman transaksi.



**Gambar 6.6 Implementasi Halaman Transaksi**

#### f. Implementasi Halaman Profil

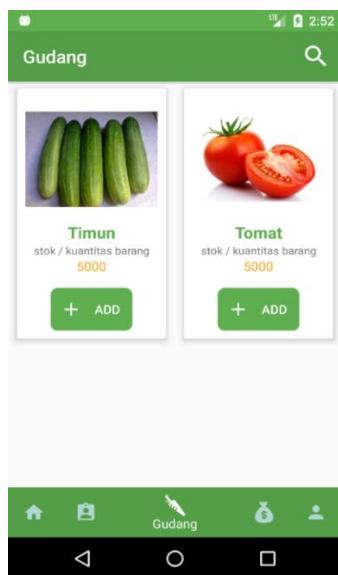
Tampilan halaman Profil muncul ketika *user* menekan tombol yang ada dipojok kanan bawah yaitu tombol *profile*. Halaman ini akan menampilkan data *user* yang berhasil login. Dihalaman ini *user* dapat merubah nama ,email ,no telepon dan foto. Pada Gambar 6.7 berikut ini menunjukkan halaman Profil.



**Gambar 6.7 Implementasi Halaman Profil**

### **g. Implementasi Halaman Gudang Sayur**

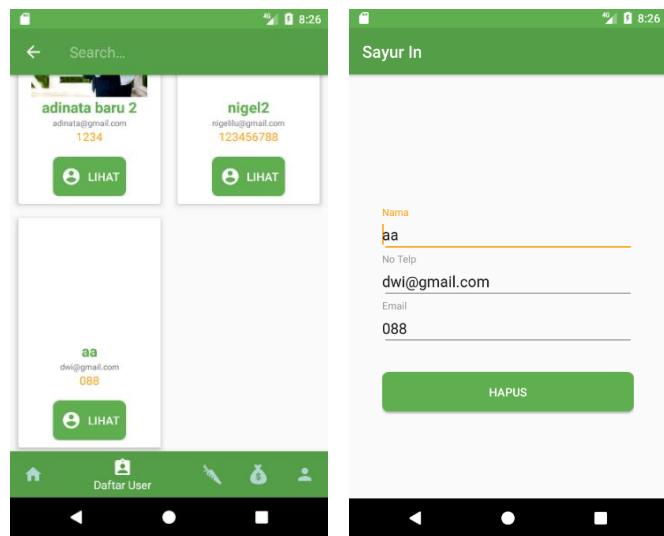
Tampilan halaman gudang sayur adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data sayur yang ada digudang. Dihalaman ini terdapat tombol add yang berfungsi untuk menambahkan sayur yang ingin dijual dari gudang ,dan diatas juga terdapat tombol search untuk mencari sayur di gudang dengan spesifik. Pada Gambar 6.8 berikut ini menunjukkan halaman Gudang Sayur.



**Gambar 6.8 Implementasi Halaman Gudang Sayur**

### **h. Implementasi Halaman *Manage User***

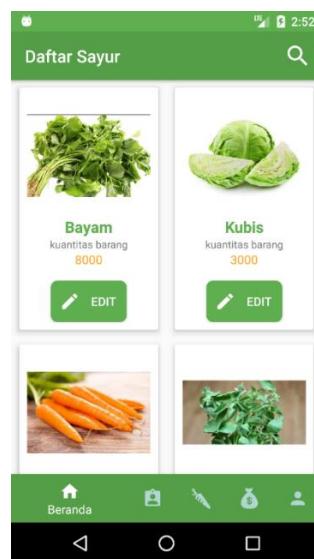
Tampilan halaman *Manage User* adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data *user* yang telah mendaftar. Dihalaman ini terdapat tombol lihat yang berfungsi untuk memanage *user*, dan diatas juga terdapat tombol search untuk mencari *user* dengan spesifik. Pada Gambar 6.9 berikut ini menunjukkan halaman *Manage User*.



Gambar 6.9 Implementasi Halaman *Manage User*

#### i. Implementasi Halaman *Manage Sayur*

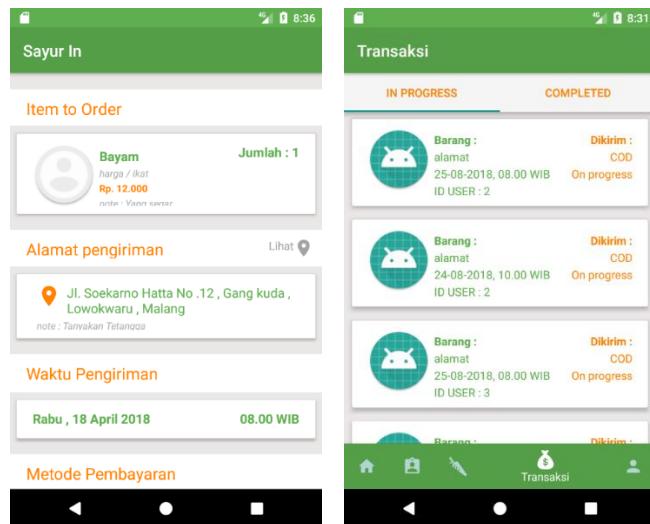
Tampilan halaman *Manage Sayur* adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data sayur yang telah dijual. Dihalaman ini terdapat tombol edit yang berfungsi untuk memanage data sayur, dan diatas juga terdapat tombol search untuk mencari sayur yang dijual dengan spesifik. Pada Gambar 6.10 berikut ini menunjukkan halaman *Manage Sayur*.



Gambar 6.10 Implementasi Halaman *Manage Sayur*

### j. Implementasi Halaman *Manage Transaksi*

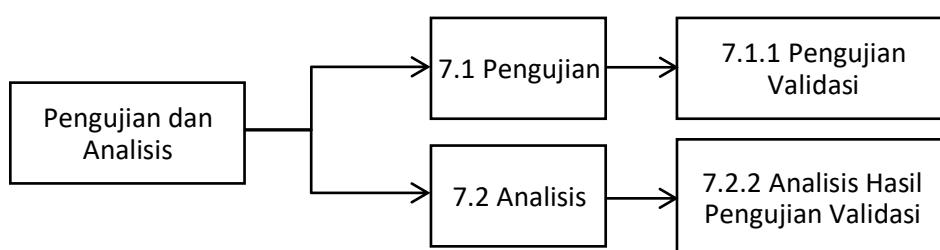
Tampilan halaman *Manage Transaksi* adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data sayur yang dibeli oleh *user*. Dihalaman ini admin dapat mengedit transaksi yang telah dilakukan oleh *user*. Pada Gambar 6.11 berikut ini menunjukkan halaman *Manage Transaksi*.



Gambar 6.11 Implementasi Halaman *Manage Transaksi*

## BAB 7 PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis akan menjelaskan tahapan dalam melakukan pengujian dan analisa terhadap Sistem Jual Beli Sayur Online Berbasis Android dengan *Web Service*. Pengujian dilakukan dengan validasi untuk tiap kasus menggunakan *Black – Box testing*. Kemudian dilakukan tahap analisis dari hasil pengujian yang telah dilakukan. Berikut Gambar 7.1 merupakan alur tahap pengujian dan analisis.



**Gambar 7.1 Diagram Alir Pengujian dan Analisis**

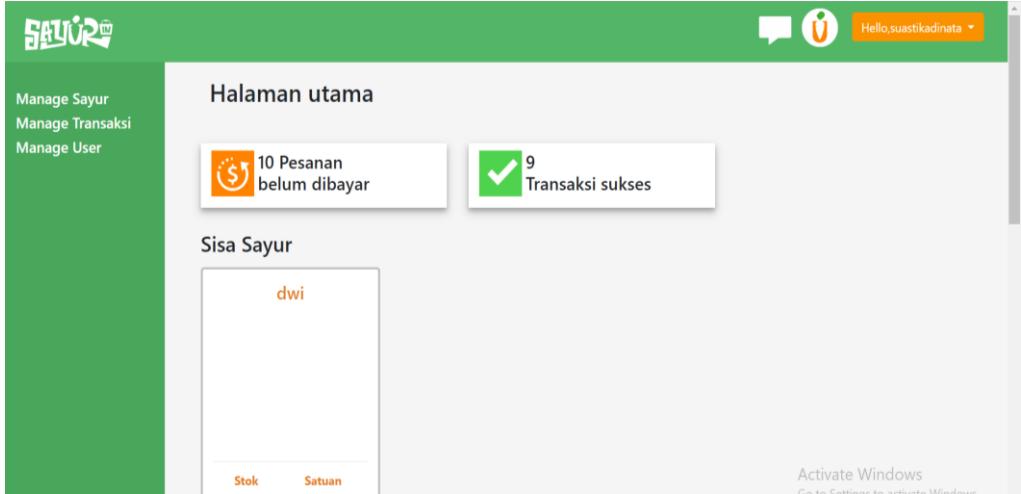
### 7.1 Pengujian

Pada pengujian sistem yang dilakukan menggunakan tahap pengujian validasi. Pengujian validasi berfungsi untuk menentukan apakah hasil dari pengembangan sistem telah sesuai dengan kebutuhan.

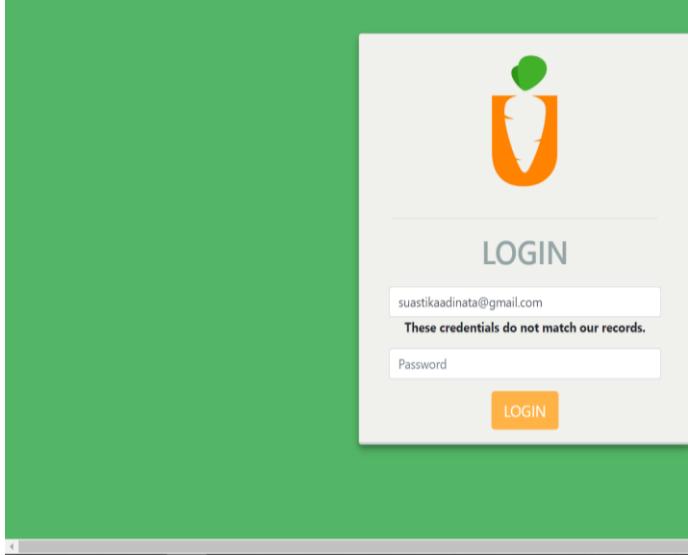
#### 7.1.1 Pengujian Validasi

Pengujian digunakan untuk memastikan bahwa sistem berfungsi sesuai perancangan sistem dan memenuhi kebutuhan *user*. Pengujian yang digunakan pada sistem ini adalah teknik pengujian *black box*. Pada pengujian *black box* digunakan metode pengujian validasi bertujuan untuk mengetahui apakah sistem yang dikembangkan sesuai dengan kebutuhan perangkat lunak yang sudah ditentukan sebelumnya. Berikut Tabel 7.1 merupakan pengujian validasi dari aplikasi Sistem Jual Beli Sayur Online Berbasis Android dengan *Web Service*.

**Tabel 7.1 Tabel Pengujian Validasi**

No	Test Name	Test Case	Expected Result	Result	Status
<b>Pengujian Login Web</b>					
1	Pengujian Login web	Pengujian dengan menggunakan <i>email</i> yang benar. Kemudian <i>user</i> melakukan klik tombol login	Login berhasil dan masuk ke halaman <i>home web</i> .	Login berhasil dan masuk ke halaman <i>home web</i> .	Valid
 					

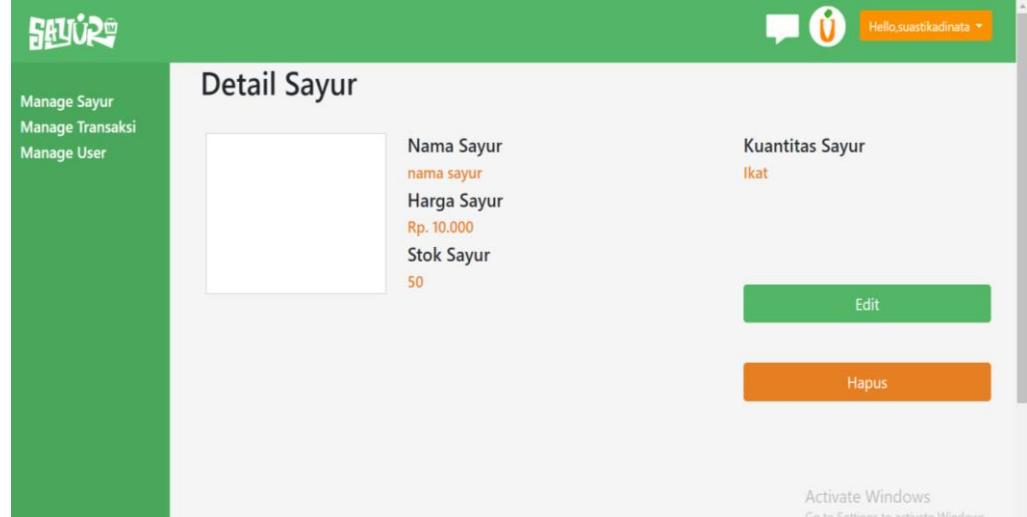
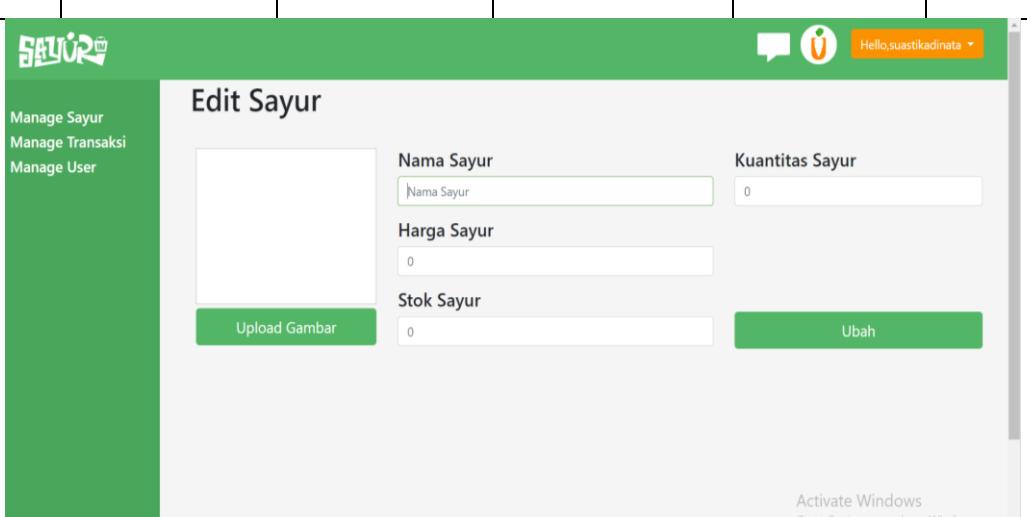
**Tabel 7.2 Tabel Pengujian Validasi (lanjutan)**

2	Pengujian Login web	Pengujian dengan menggunakan email yang salah. Kemudian user melakukan klik tombol login	Login gagal dan dikembalikan ke halaman <i>login</i> , menampilkan pesan "Email atau password tidak cocok"	Login gagal dan dikembalikan ke halaman <i>login</i> , menampilkan pesan "Email atau password tidak cocok"	Valid
					
<b>Pengujian tambah, edit, hapus data sayur web</b>					
3	Pengujian menambah data sayur, melihat data sayur yang ditambahkan.	Pengujian dengan mengklik menu tambah sayur, lalu user menambah data sayur dengan lengkap.	Data berhasil ditambahkan pada menu manage sayur. Dan melihat seluruh data sayur yang ditambahkan pada Menu manage sayur	Data berhasil ditambahkan pada menu manage sayur. Dan melihat seluruh data sayur yang ditambahkan pada Menu manage sayur	Valid

**Tabel 7.3 Tabel Pengujian Validasi (lanjutan)**

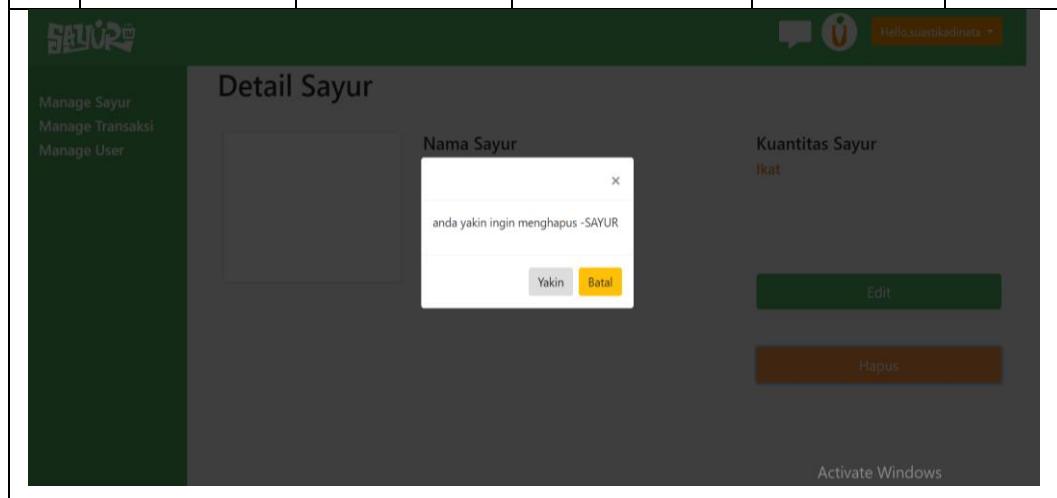
	
4 Pengujian Melihat data sayur.	Pengujian dengan mengklik sayur yang ingin dilihat. Menampilkan Data lengkap tentang sayur yang ditambahkan. Menampilkan Data lengkap tentang sayur yang ditambahkan. Valid

**Tabel 7.4 Tabel Pengujian Validasi (lanjutan)**

	 <table border="1"> <thead> <tr> <th>No</th><th>Pengujian</th><th>Pengujian</th><th>Berhasil diedit</th><th>Berhasil</th><th>Valid</th></tr> </thead> <tbody> <tr> <td>5</td><td>Mengedit data sayur.</td><td>dengan mengklik menu edit sayur pada detail sayur. Lalu user mengisi data yang diinginkan dengan lengkap.</td><td>dan tampil pada daftar sayur.</td><td>dan tampil pada daftar sayur.</td><td></td></tr> </tbody> </table>  <table border="1"> <thead> <tr> <th>No</th><th>Pengujian</th><th>Pengujian</th><th>Berhasil diedit</th><th>Berhasil</th><th>Valid</th></tr> </thead> <tbody> <tr> <td>5</td><td>Mengedit data sayur.</td><td>dengan mengklik menu edit sayur pada detail sayur. Lalu user mengisi data yang diinginkan dengan lengkap.</td><td>dan tampil pada daftar sayur.</td><td>dan tampil pada daftar sayur.</td><td></td></tr> </tbody> </table>	No	Pengujian	Pengujian	Berhasil diedit	Berhasil	Valid	5	Mengedit data sayur.	dengan mengklik menu edit sayur pada detail sayur. Lalu user mengisi data yang diinginkan dengan lengkap.	dan tampil pada daftar sayur.	dan tampil pada daftar sayur.		No	Pengujian	Pengujian	Berhasil diedit	Berhasil	Valid	5	Mengedit data sayur.	dengan mengklik menu edit sayur pada detail sayur. Lalu user mengisi data yang diinginkan dengan lengkap.	dan tampil pada daftar sayur.	dan tampil pada daftar sayur.	
No	Pengujian	Pengujian	Berhasil diedit	Berhasil	Valid																				
5	Mengedit data sayur.	dengan mengklik menu edit sayur pada detail sayur. Lalu user mengisi data yang diinginkan dengan lengkap.	dan tampil pada daftar sayur.	dan tampil pada daftar sayur.																					
No	Pengujian	Pengujian	Berhasil diedit	Berhasil	Valid																				
5	Mengedit data sayur.	dengan mengklik menu edit sayur pada detail sayur. Lalu user mengisi data yang diinginkan dengan lengkap.	dan tampil pada daftar sayur.	dan tampil pada daftar sayur.																					

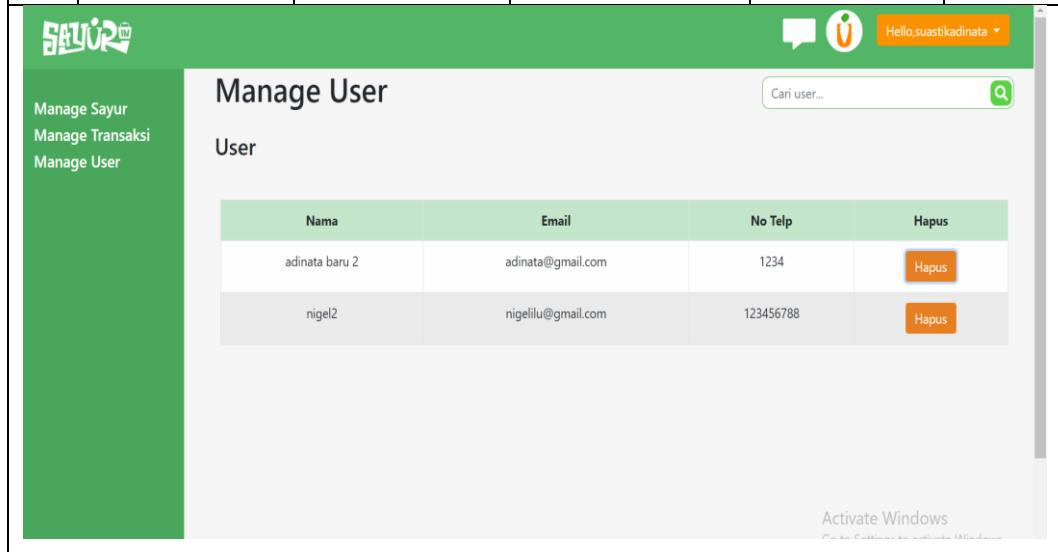
**Tabel 7.5 Tabel Pengujian Validasi (lanjutan)**

6	Pengujian Hapus data sayur	Pengujian dengan melakukan klik pada tombol hapus pada detail sayur.	Akan muncul modal yang berisi “apakah anda yakin ingin menghapus sayur?” jika dipilih “Yakin” maka data sayur akan dihapus.	Akan muncul modal yang berisi “apakah anda yakin ingin menghapus sayur?” jika dipilih “Yakin” maka data sayur akan dihapus.	Valid
---	----------------------------	--	---	---	-------

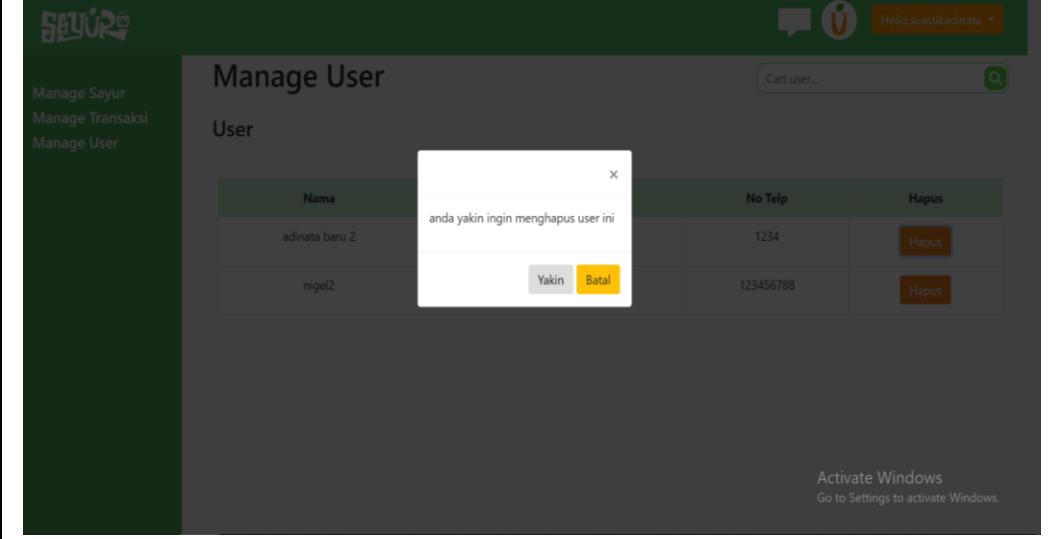
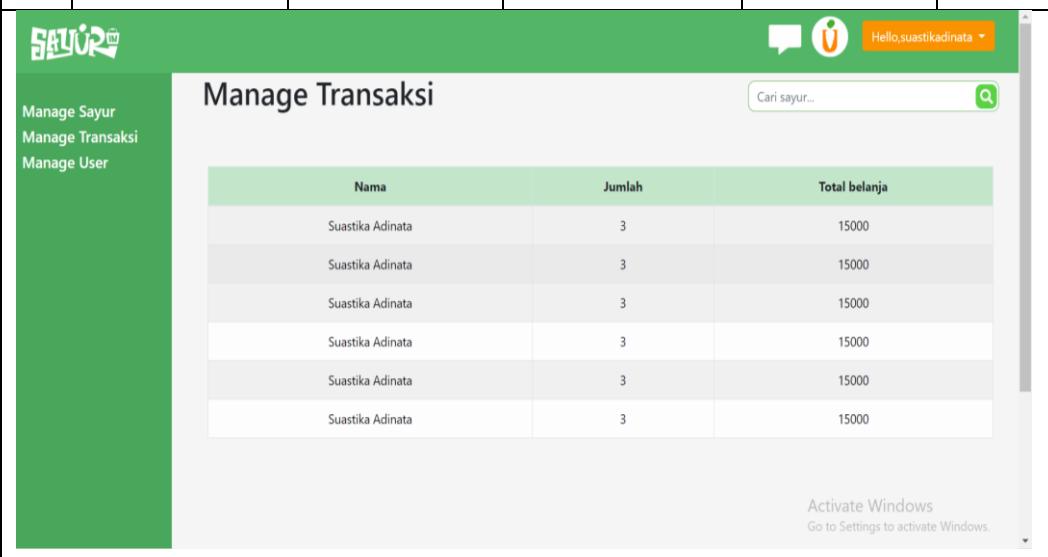


**Tabel 7.6 Tabel Pengujian Validasi (lanjutan)**

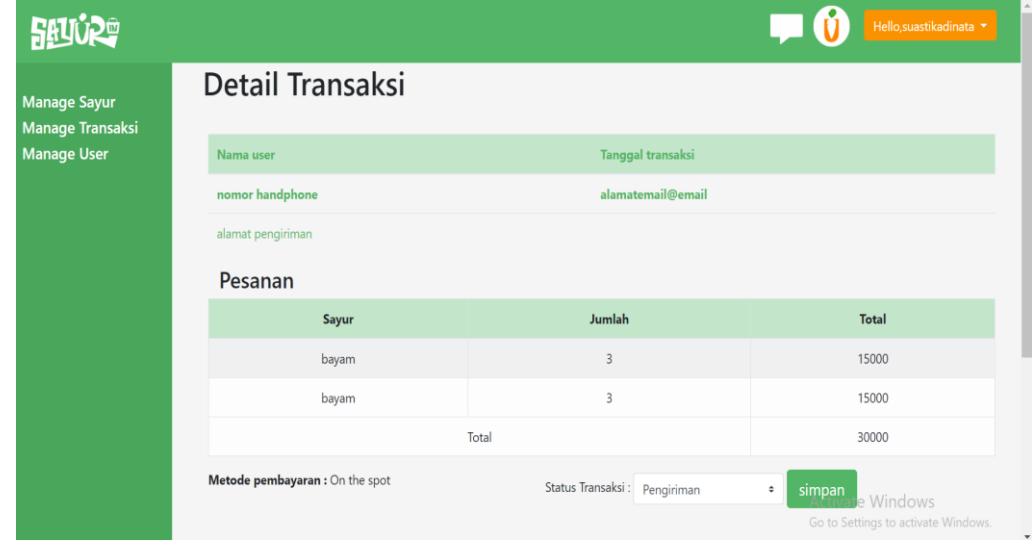
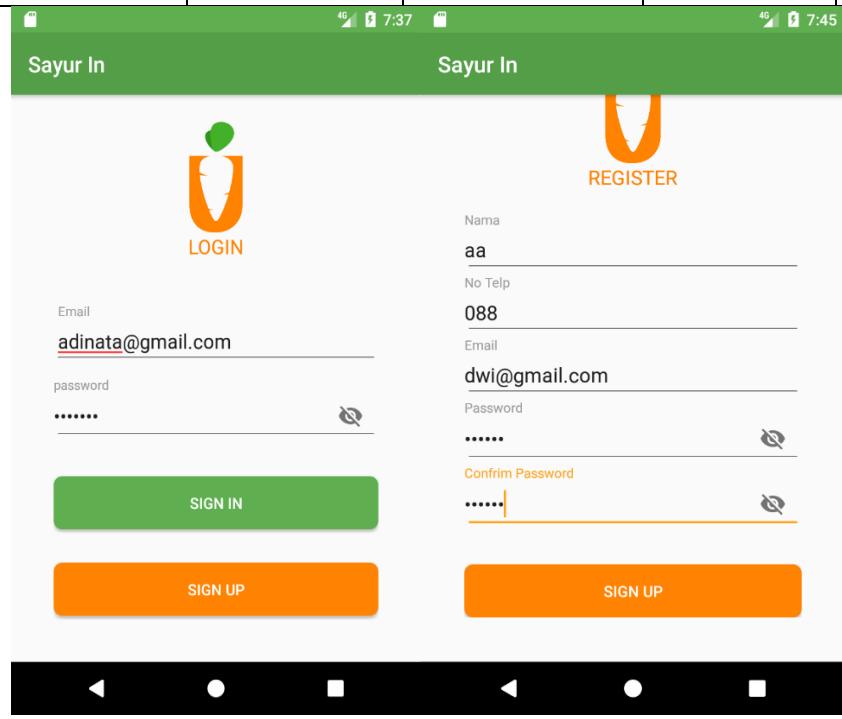
Melihat dan menghapus data <i>user</i> web					
7	Melihat data <i>user</i> dan menghapus data <i>user</i> .	Pengujian dengan memilih menu “Manage user” maka akan menampilkan seluruh data <i>user</i> . Data <i>user</i> dapat dihapus dengan menekan tombol hapus pada kolom <i>user</i> .	Menampilkan data <i>user</i> yang login. Jika menekan tombol hapus maka akan menampilkan modal “Apakah anda yakin ingin menghapus data <i>user</i> ?", jika dipilih yakin maka data <i>user</i> dihapus.	Menampilkan data <i>user</i> yang login. Jika menekan tombol hapus maka akan menampilkan modal “Apakah anda yakin ingin menghapus data <i>user</i> ?", jika dipilih yakin maka data <i>user</i> dihapus.	Valid



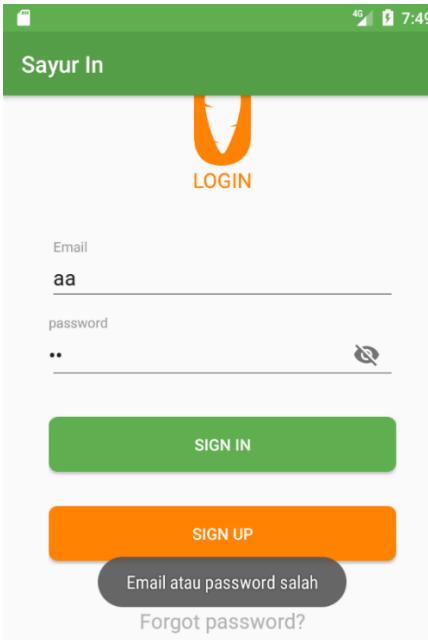
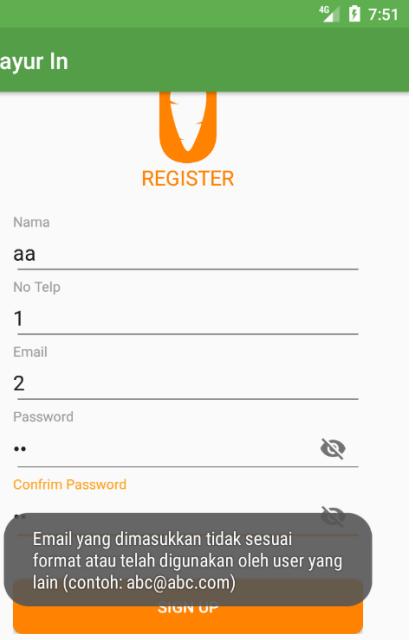
**Tabel 7.7 Tabel Pengujian Validasi (lanjutan)**

					
<b>Pengujian Lihat data transaksi web</b>					
8	Pengujian Lihat Data transaksi	Pengujian dengan mengklik menu “Manage transaksi”	Menampilkan halaman lihat data transaksi	Menampilkan halaman lihat data transaksi	Valid
					

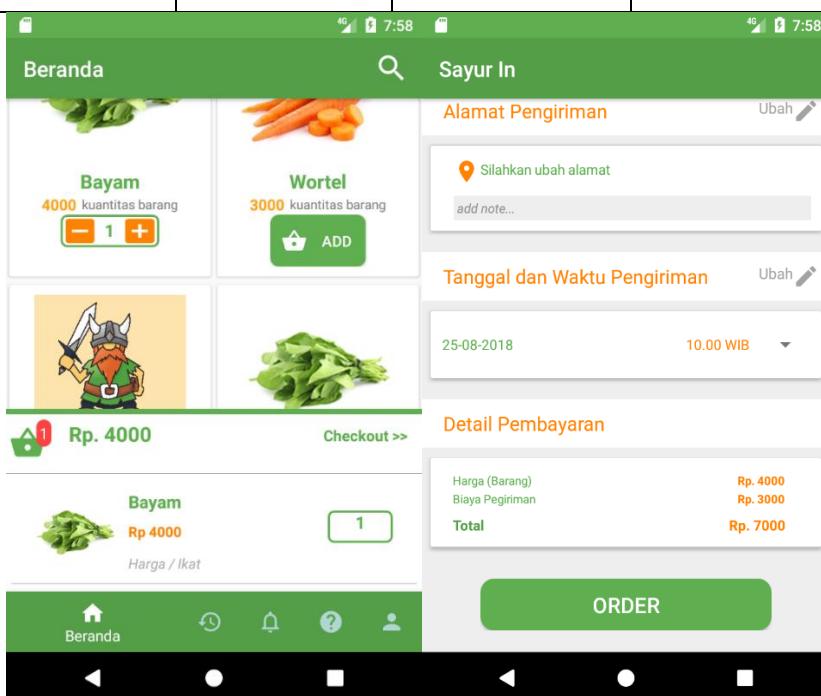
**Tabel 7.8 Tabel Pengujian Validasi (lanjutan)**

 <p>The screenshot shows the 'Detail Transaksi' (Transaction Detail) screen. At the top, there's a green header bar with the 'Sayur In' logo. Below it, a sidebar on the left lists 'Manage Sayur', 'Manage Transaksi', and 'Manage User'. The main content area has a table for user information:</p> <table border="1"> <thead> <tr> <th>Nama user</th> <th>Tanggal transaksi</th> </tr> </thead> <tbody> <tr> <td>nomor handphone</td> <td>alamatemail@email</td> </tr> </tbody> </table> <p>Below this is a section for delivery address. Then, there's a table for the order details:</p> <table border="1"> <thead> <tr> <th>Sayur</th> <th>Jumlah</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>bayam</td> <td>3</td> <td>15000</td> </tr> <tr> <td>bayam</td> <td>3</td> <td>15000</td> </tr> <tr> <td></td> <td>Total</td> <td>30000</td> </tr> </tbody> </table> <p>At the bottom, there are buttons for payment method ('On the spot'), transaction status ('Pengiriman'), and a 'simpan' (Save) button.</p>	Nama user	Tanggal transaksi	nomor handphone	alamatemail@email	Sayur	Jumlah	Total	bayam	3	15000	bayam	3	15000		Total	30000					
Nama user	Tanggal transaksi																				
nomor handphone	alamatemail@email																				
Sayur	Jumlah	Total																			
bayam	3	15000																			
bayam	3	15000																			
	Total	30000																			
<p align="center"><b>Pengujian Sign In, dan Sign Up pada mobile</b></p>																					
9	Pengujian Sign in, dan sign up.	Pengujian dengan mengisi data pada form sign in dan sign up.	Masuk ke halaman home.	Masuk ke halaman home.	Valid																
 <p>The screenshot shows two side-by-side login screens on a mobile device. Both screens have a green header with the 'Sayur In' logo. The left screen is for 'SIGN IN' and has fields for 'Email' (with value 'adinata@gmail.com') and 'password' (with value '*****'). It has a large green 'SIGN IN' button at the bottom. The right screen is for 'REGISTER' and has fields for 'Nama' (with value 'aa'), 'No Telp' (with value '088'), 'Email' (with value 'dwi@gmail.com'), 'Password' (with value '*****'), 'Confirm Password' (with value '*****'), and 'SIGN UP' buttons at the bottom. The device status bar shows the time as 7:37 on the left and 7:45 on the right.</p>																					

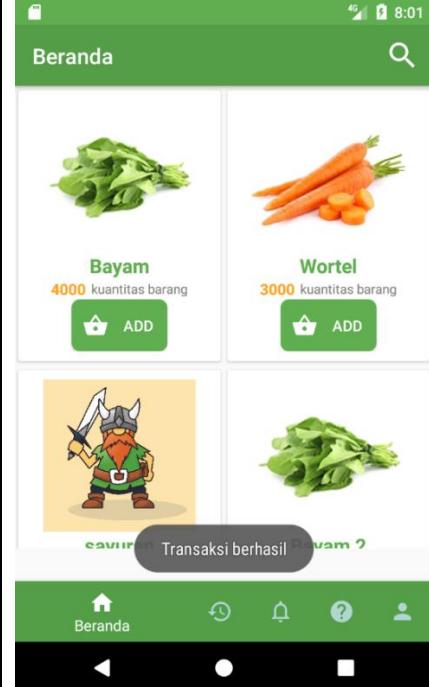
**Tabel 7.9 Tabel Pengujian Validasi (lanjutan)**

10	Pengujian Sign in, dan sign up.	Pengujian dengan mengisi form sign in dan sign up dengan data yang salah.	Menampilkan toast “password dan email salah” jika sign in gagal.	Menampilkan toast “password dan email salah” jika sign in gagal.	Valid
 					

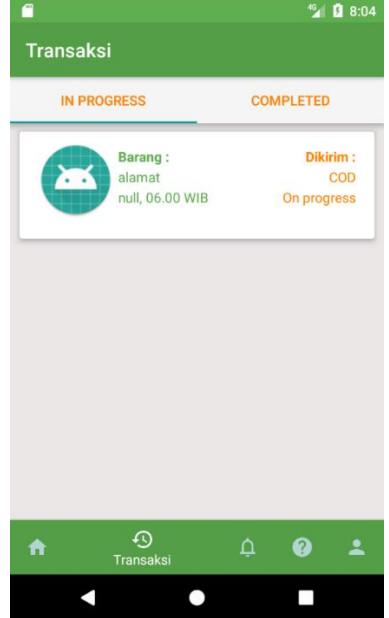
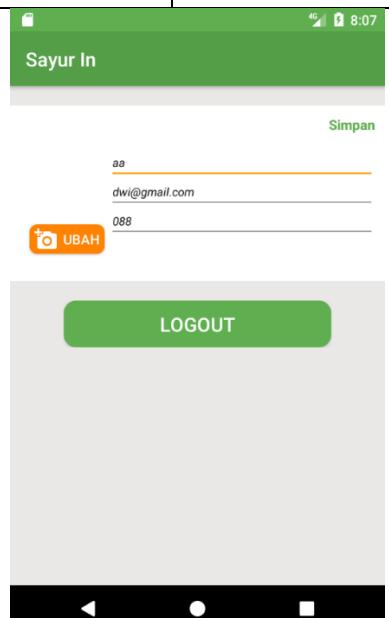
**Tabel 7.10 Tabel Pengujian Validasi (lanjutan)**

<b>Pengujian Lihat daftar sayur dan melakukan order</b>					
11	Pengujian Lihat daftar sayur dan melakukan order.	Pengujian dengan mengklik menu home dan menambah dengan klik tombol “Add” lalu menekan tombol checkout pada bottomsheet order dilakukan dengan mengisi data pada halaman checkout.	Menampilkan <i>bottom sheet</i> yang berisikan sayur yang ditambahkan. Setelah menekan tombol “checkout” maka akan kembali kepada tampilan home.	Menampilkan <i>bottom sheet</i> yang berisikan sayur yang ditambahkan. Setelah menekan tombol “checkout” maka akan kembali kepada tampilan home.	Valid
					

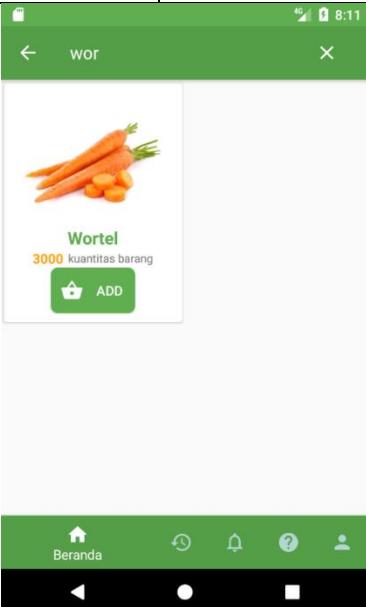
**Tabel 7.11 Tabel Pengujian Validasi (lanjutan)**

	
12 Melihat histori transaksi.	<p>Pengujian dengan memilih menu histori .</p> <p>Menampilkan data transaksi yang telah dilakukan yang sedang “on progress” atau sudah “completed”.</p> <p>Menampilkan data transaksi yang telah dilakukan yang sedang “on progress” atau sudah “completed”.</p> <p>Valid</p>

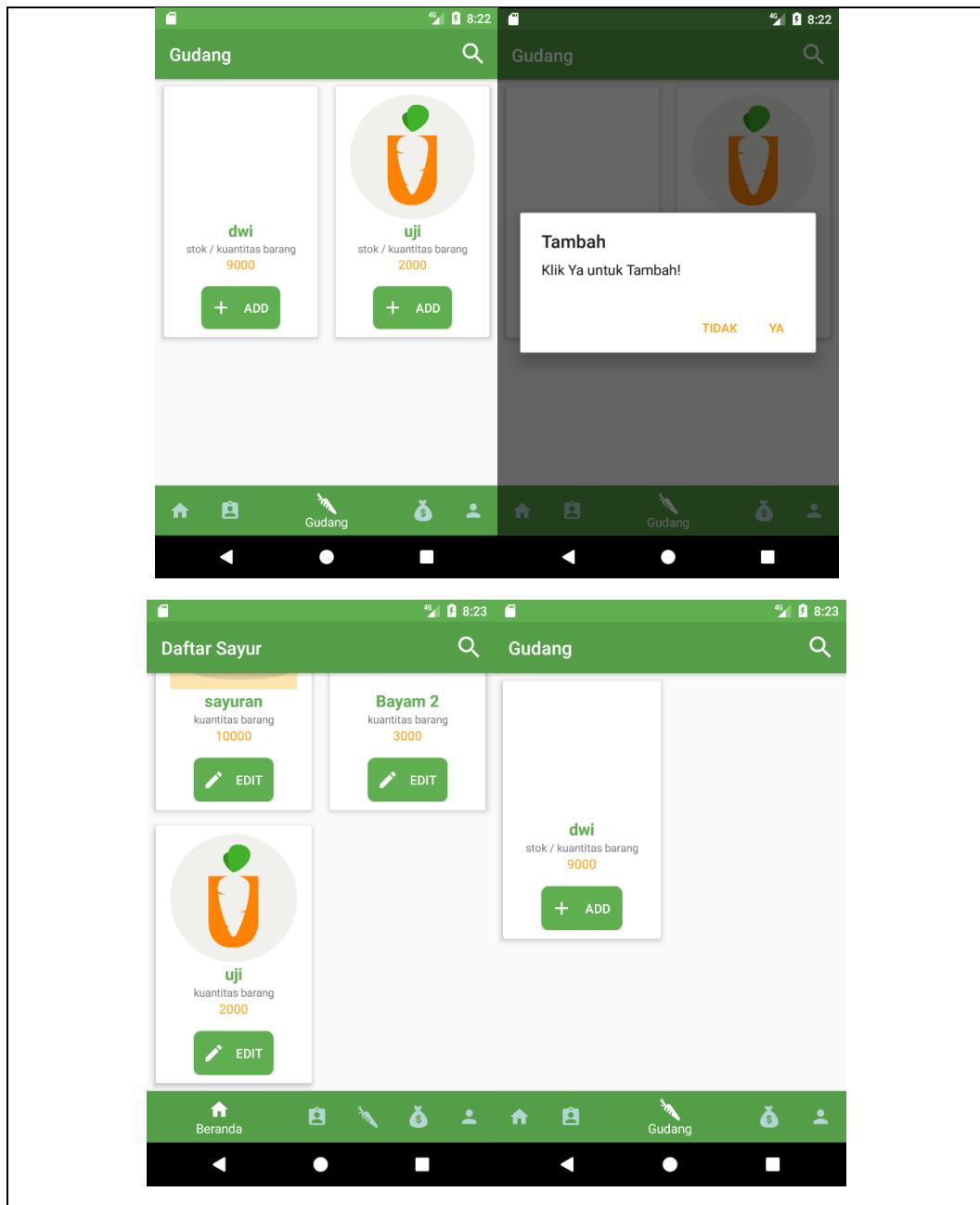
**Tabel 7.12 Tabel Pengujian Validasi (lanjutan)**

					
<b>Pengujian Edit profil</b>					
13	Pengujian edit profil.	Pengujian dengan memilih menu profil lalu mengisi data dengan data baru lalu di klik simpan.	Data profil akan disimpan dan muncul toast “Berhasil menyimpan data”	Data profil akan disimpan dan muncul toast “Berhasil menyimpan data”	Valid
					

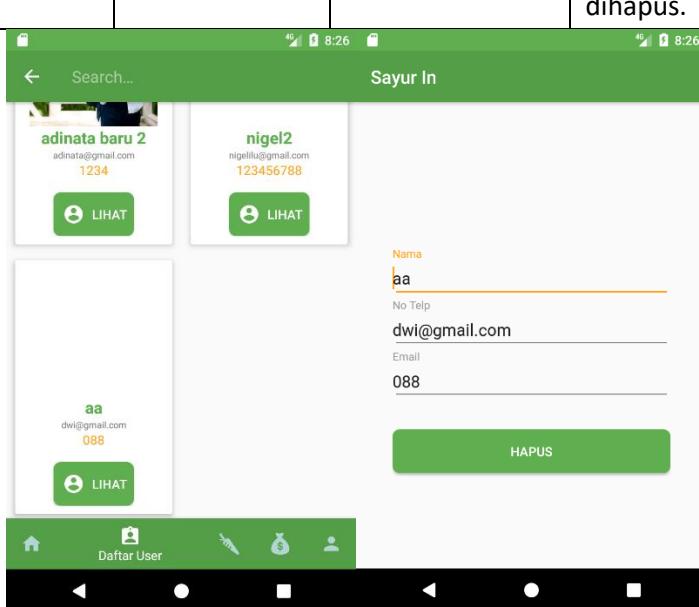
**Tabel 7.13 Tabel Pengujian Validasi (lanjutan)**

<b>Pengujian search sayur</b>					
14	Pengujian search sayur	Pengujian dengan menginput nama sayur yang ingin dicari pada menu search pada halaman home.	Mengetik nama sayur dan menampilkan sayur yang dicari.	Mengetik nama sayur dan menampilkan sayur yang dicari.	Valid
					
<b>Pengujian Menambah sayur yang dijual dari gudang</b>					
15	Pengujian Menambah sayur yang dijual dari gudang.	Pengujian dengan mengklik tombol "Add" pada menu gudang.	Menampilkan pop-up. Jika memilih "Ya" maka sayur akan hilang dari gudang dan berpindah ke halaman sayur yang dijual.	Menampilkan pop-up. Jika memilih "Ya" maka sayur akan hilang dari gudang dan berpindah ke halaman sayur yang dijual.	Valid

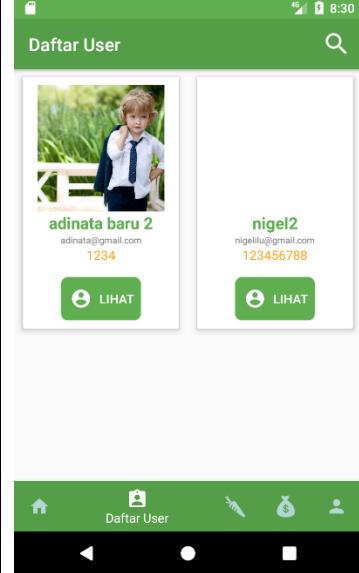
**Tabel 7.6 Tabel Pengujian Validasi (lanjutan)**



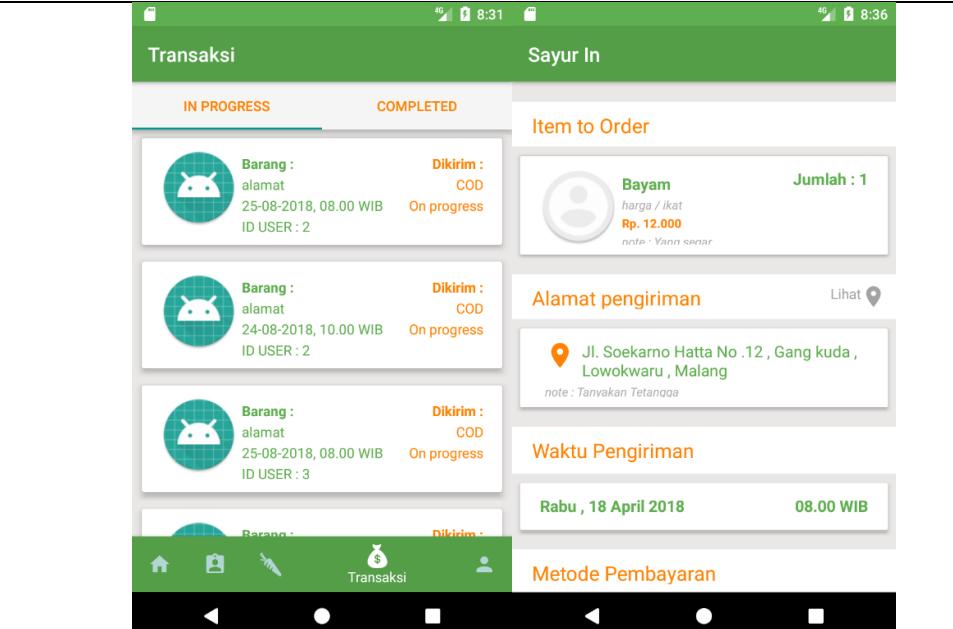
**Tabel 7.14 Tabel Pengujian Validasi (lanjutan)**

<b>Pengujian Melihat daftar user, Detail dan hapus user</b>					
16	Pengujian Daftar user , detail user, dan hapus user.	Pengujian dengan mengklik menu daftar user. Mengklik tombol lihat maka menampilkan detail user. Melakukan klik hapus jika ingin menghapus user.	Menampilkan daftar user yang terdaftar. Jika melakukan klik "lihat" maka menampilkan halaman detail user. Menampilkan pop-up jika ingin menghapus user jika memilih "Ya" maka data user akan dihapus.	Menampilkan daftar user yang terdaftar. Jika melakukan klik "lihat" maka menampilkan halaman detail user. Menampilkan pop-up jika ingin menghapus user jika memilih "Ya" maka data user akan dihapus.	Valid
					

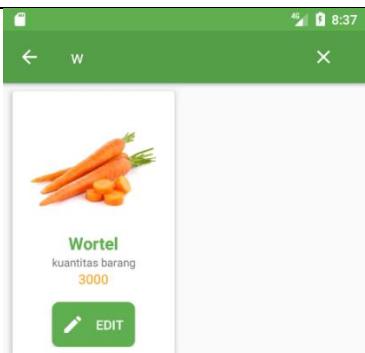
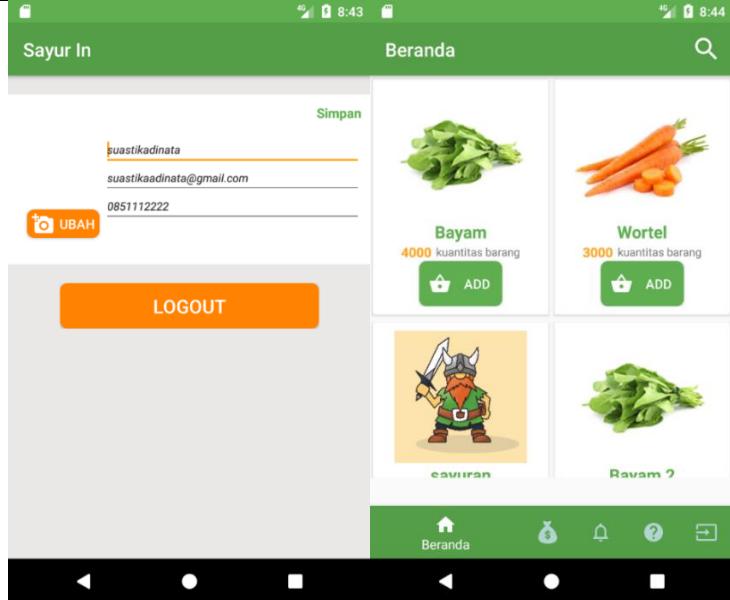
**Tabel 7.15 Tabel Pengujian Validasi (lanjutan)**

	<b>Melihat data transaksi dari user</b>				
17	Pengujian Melihat data transaksi yang dilakukan.	Pengujian dengan mengklik menu transaksi dan melakukan klik pada transaksi untuk melihat detail transaksi.	Menampilkan list transaksi yang sudah dilakukan. Menampilkan detail transaksi jika mengklik data transaksi.	Menampilkan list transaksi yang sudah dilakukan. Menampilkan detail transaksi jika mengklik data transaksi.	Valid

**Tabel 7.16 Tabel Pengujian Validasi (lanjutan)**

 <p>The screenshot shows a mobile application interface for a grocery store. At the top, there's a navigation bar with icons for home, user profile, and search. Below it, a header bar says "Transaksi" and "Sayur In". A sub-header indicates "IN PROGRESS" and "COMPLETED". The main content area displays three transaction cards, each with a small icon, item name, address, delivery method (COD), and status ("On progress"). To the right, there's a sidebar titled "Item to Order" showing a single item: "Bayam" (jumlah: 1). Further down, sections for "Alamat pengiriman" (Jl. Soekarno Hatta No.12, Gang kuda, Lowokwaru, Malang) and "Waktu Pengiriman" (Rabu, 18 April 2018, 08.00 WIB) are shown. At the bottom, a section for "Metode Pembayaran" is partially visible.</p>		<p><b>Pengujian search sayur (admin)</b></p> <table border="1"> <tr> <td>18</td><td>Pengujian Search sayur pada admin.</td><td>Pengujian dengan mengisi nama sayur yang diinginkan pada search.</td><td>Menampilkan data sayur yang diinginkan.</td><td>Menampilkan data sayur yang diinginkan.</td><td>Valid</td></tr> </table>	18	Pengujian Search sayur pada admin.	Pengujian dengan mengisi nama sayur yang diinginkan pada search.	Menampilkan data sayur yang diinginkan.	Menampilkan data sayur yang diinginkan.	Valid
18	Pengujian Search sayur pada admin.	Pengujian dengan mengisi nama sayur yang diinginkan pada search.	Menampilkan data sayur yang diinginkan.	Menampilkan data sayur yang diinginkan.	Valid			

**Tabel 7.17 Tabel Pengujian Validasi (lanjutan)**

 						
<b>Pengujian Logout</b>						
19	Pengujian Logout.	Pengujian dengan mengklik tombol logout pada bagian profil maka akan kembali kepada halaman home.	Menampilkan halaman <i>home</i> dan <i>user logout</i> . Ikon profil juga akan berubah.	Menampilkan halaman <i>home</i> dan <i>user logout</i> . Ikon profil juga akan berubah.	Valid	
						

## **7.2 Analisis**

Proses analisis dilakukan untuk mendapatkan hasil dari kesimpulan dari pengujian validasi yang telah dilakukan pada Sistem Jual Beli Sayur Berbasis Android dengan *Web Service*. Tahap analisis bertujuan sebagai perbaikan pada program yang telah dikembangkan apabila terdapat ketidaksesuaian fungsi pada sistem.

### **7.2.1 Analisis Hasil Pengujian Validasi**

Pengujian yang telah dilakukan dengan melihat daftar kebutuhan sistem dari bab analisis dan perancangan sebagai acuan pengujian. Dengan menggunakan teknik pengujian *black-box* menunjukkan validitas 100% terhadap hasil fungsi pada *expected result* dan hasil fungsi pada *result*, sehingga apa yang diharapkan sebelumnya sesuai dengan hasil apa yang didapatkan. Dengan demikian Sistem Jual Beli Sayur Berbasis Android dengan *Web Service* dapat dinyatakan sesuai dengan daftar kebutuhan yang telah diuraikan dalam dokumentasi dari tahap analisa, perancangan dan implementasi.

## BAB 8 PENUTUP

### 8.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi dan pengujian yang dilakukan, maka dapat diperoleh kesimpulan sebagai berikut :

1. Perancangan sayur online dimulai dengan melakukan pemodelan kebutuhan yang terdiri dari membuat *entity relationship diagram, use case diagram, use case skenario, class diagram, activity diagram, and sequence diagram*. Kemudian setelah melakukan pemodelan kebutuhan maka selanjutnya akan dilakukan perancangan algoritma dari aplikasi sayur online yang akan dipakai. Setelah melakukan pemodelan kebutuhan dan perancangan algoritma maka perancangan selanjutnya yang dilakukan adalah dengan melakukan perancangan antar muka dari aplikasi sayur online.
2. Setelah melakukan perencanaan sistem maka selanjutnya adalah melakukan implementasi. Implementasi terbagi menjadi dua bagian yaitu implementasi algoritma dan implementasi antar muka. Untuk implementasi algoritma digunakan tiga algoritma utama yang menggunakan webservice yaitu algoritma tambah data sayur, lihat data sayur, dan lihat data user.

### 8.2 Saran

Saran yang diberikan dalam pengembangan Sistem Jual Beli Sayur Berbasis Android dengan *Web Service* kedepannya adalah sebagai berikut :

1. Untuk pengembangan lebih lanjut, dapat ditambahkan fitur-fitur baru yaitu seperti transaksi dengan pembayaran menggunakan ATM atau transfer pembayaran melalui bank yang telah ditentukan, menambah fitur chatting dengan *user* dengan admin, penambahan *notifikasi*, penambahan fungsi agar pada saat keluar dari aplikasi data sayur yang tersimpan pada keranjang tidak

menghilang, penambahan fungsi untuk melakukan penambahan *user* pada bagian web sayurin dan juga dilakukan perbaikan-perbaikan pada beberapa fitur agar prosesnya dapat lebih sederhana dan *user friendly*.

2. Dapat dilanjutkan pengembangannya pada platform lain agar pelayanan dapat merata kepada seluruh konsumen atau pelanggan.

## DAFTAR PUSTAKA

- Android, 2018. *About the Android Open Source Project*. [online] Tersedia di : <  
<https://source.android.com/> > [Diakses 11 September 2018].
- Android Developer, 2018. *Meet Android Studio*.[online] Tersedia di : <[https://developer.android.com/studio/intro/#top\\_of\\_page](https://developer.android.com/studio/intro/#top_of_page)> [Diakses 10  
September 2018].
- Armel, J. 2014. *Web application development with Laravel PHP Framework version 4*. S1.Helsinki : Metropolia University. Tersedia di :  
<<http://www.theseus.fi/bitstream/handle/10024/74052/Author.pdf?sequence=1&isAllowed=y>> [Diakses pada 18 september 2018]
- Bell, D. 2003. *UML basics: An introduction to the Unified Modeling Language* [e-book]. IBM : Developer works. Tersedia di : IBM Developer Works <  
<https://www.ibm.com/developerworks/rational/library/769.html> >  
[Diakses pada 20 agustus 2018 ].
- Bittner, K.2006. Introduction To Writing Good Use Cases. *Development Conference 2006*.[Pdf] IBM Rational Software. Tersedia di : <[http://www-07.ibm.com/shared\\_downloads2/software/rsdc2006/ra\\_day\\_1/WritingGoodUseCases.pdf](http://www-07.ibm.com/shared_downloads2/software/rsdc2006/ra_day_1/WritingGoodUseCases.pdf)>[Diakses pada 18 September 2018]
- Chen , X., Ji , Z., Yu , F. & Zhan , Y. 2017. Restful API Architecture Based on Laravel Framework. *Journal of Physics: Conference Series*, [e-journal] 910 (012016).Tersedia di : IOP Publishing<<http://iopscience.iop.org/article/10.1088/1742-6596/910/1/012016>> [Diakses pada 18 september 2018] .
- Dennis, A., Wixom, B.H. & Roth , R.M., 2012. *System Analysis & Design*. 5th ed. USA : Jhon Wiley & Sons, Inc.
- Erinton, Ruli, Negara, Ridha Muldina, dan Sanjoyo, Danu Dwi. 2017. Analisis Performansi Framework Codeigniter dan Laravel Menggunakan Web Server Apache. *Jurnal e-Proceeding of Engineering*, Vol. 4: 3565-3572.
- Laravel , 2018. *Introduction*. [online] Tersedia di : <<https://laravel.com/docs/4.2/>

introduction /> [Diakses 11 September 2018].

Marlindawati, 2014. *Aplikasi Pemesanan Menu Makanan Di Rumah Makan Berbasis Web Service Menggunakan Mobile Android.* S1. Universitas binadarma. Tersedia di<[http://eprints.binadarma.ac.id/2002/1/Jurnalayu\\_inayah.pdf](http://eprints.binadarma.ac.id/2002/1/Jurnalayu_inayah.pdf)> [diakses pada 10 september 2018]

Rusdiana. H.A & Irfan, Moch., 2014. *Sistem informasi manajemen.* Bandung: Pustaka Setia.

Sommerville, I., 2011. *Software engineering.* 9th ed. London: Addison-Wesley.

Smyth, N. 2015. *Android Development Essential.* 6<sup>th</sup> ed [e-book]. Tersedia di : ebookfrenzy<[https://www.ebookfrenzy.com/pdf\\_prevviews/AndroidStudio6EssentialsPreview.pdf](https://www.ebookfrenzy.com/pdf_prevviews/AndroidStudio6EssentialsPreview.pdf)> [Diakses pada 5 Juli 2018].

UML Diagram, 2018. *The Unified Modeling Language.*[online] Tersedia di : <<https://www.uml-diagrams.org/>> [Diakses 11 September 2018].

W3C. 2004. *Web Service Glossary.*[online] Tersedia di : <<https://www.w3.org/TR/ws-gloss/>> [Diakses 3 Agustus 2018].

## **LAMPIRAN 1**

### **1. FOTO DOKUMENTASI KEGIATAN**



**Foto kegiatan pkl**



**Suasana tempat pkl**

## LAMPIRAN 2

### 1. USER MANUAL

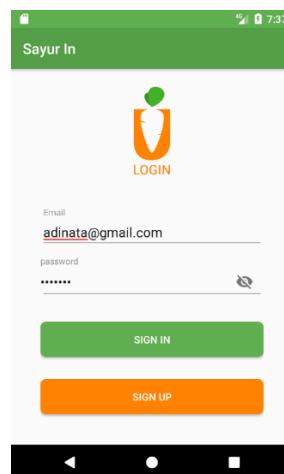
#### 1.1 Admin

##### 1.1.1 Bagian I – *Login*

*Login* pada android :

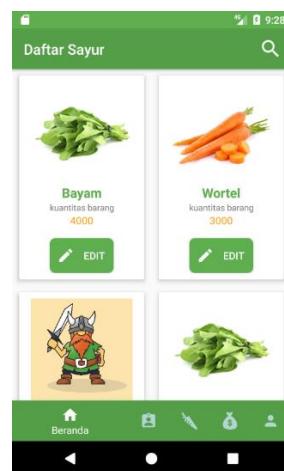
Pada bagian ini pengguna melakukan proses *login* pada aplikasi android.

Pengguna mengisi kolom *email* dan *password* yang telah di daftarkan sebelumnya. Berikut tampilan login pada android :



Tampilan *login android*

Jika proses *login* berhasil akan langsung dibawa ke halaman *home*.



Tampilan *home admin android*

Login pada web admin :

Pada halaman login web admin, admin diminta memasukkan email dan password yang telah didaftarkan sebelumnya. Berikut tampilan login pada web admin.



Tampilan *login* web

Jika proses berhasil akan ditampilkan halaman home pada web admin.

Berikut tampilan home web admin:

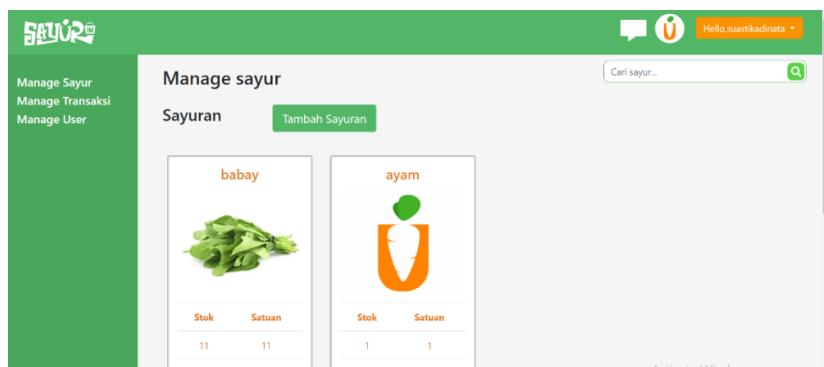
Tampilan *home* web

### 1.1.2 Bagian II – Menambahkan sayur.

Menambah sayur dari web admin.

Pada bagian ini admin dapat menambahkan sayur baru pada sistem dengan menekan menu **Manage Sayur** setelah itu akan muncul tampilan manage sayur. Di dalam halaman manage sayur kita akan ditampilkan daftar sayur yang telah dimasukkan, untuk menambahkan item sayur

admin dapat menekan tombol **Tambah Sayuran** . Setelah itu akan ditampilkan form untuk menambah sayur, di dalamnya admin mengisi kolom yang harus diisi , seperti foto , nama, harga, stok sayur, dan satuan. Setelah mengisi formnya dengan lengkap admin menekan tombol tambah, maka sayur akan ditambahkan dan akan muncul pada halaman *manage sayur*.



**Tampilan *manage sayur***

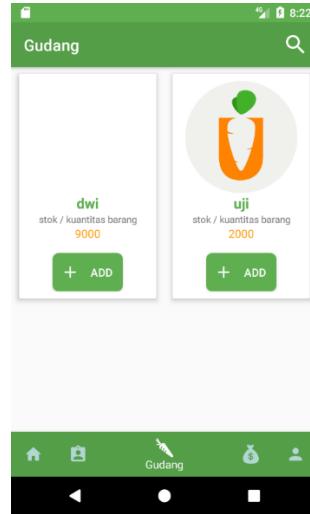
Nama Sayur	Kuantitas Sayur
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="button" value="Upload Gambar"/>	<input type="button" value="Tambah"/>

**Tampilan form tambah sayur.**

Menambah sayur dari gudang pada android.

Admin pada bagian android dapat menambahkan sayur yang telah ditambahkan ke gudang (web) ke dalam aplikasi android. Admin menekan menu dengan gambar di dalamnya ditampilkan sayur yang telah ditambahkan di web admin, untuk menambahkannya ke home admin

dapat menekan tombol  , maka akan muncul pop-up dan jika admin menekan pilihan “YA” sayur akan ditambahkan ke home sayur.

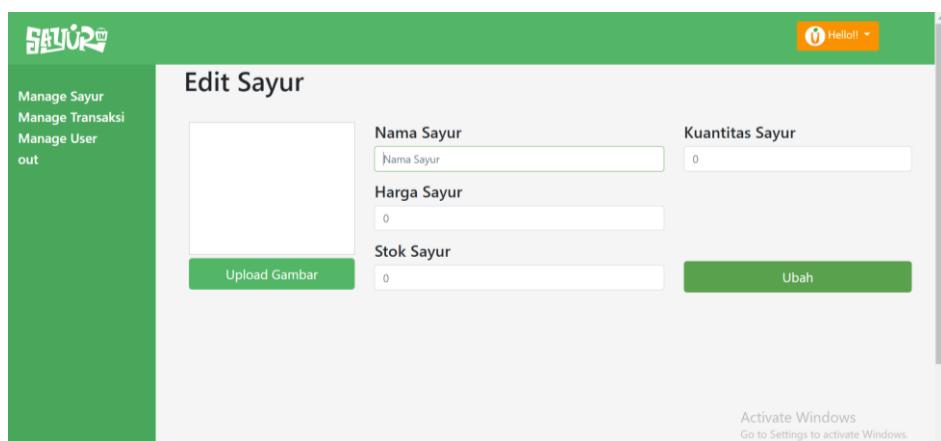


**Tampilan gudang sayur**

### **1.1.3 Bagian III – *Edit* sayur.**

Edit sayur pada web :

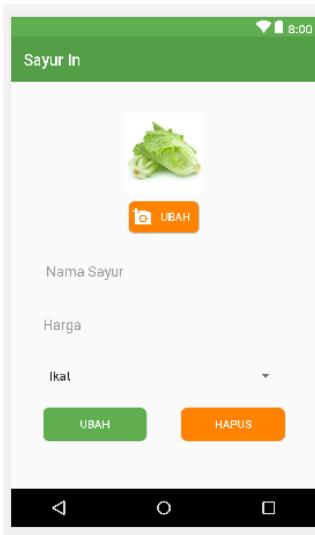
Admin dapat melakukan edit pada sayur yang telah ditambahkan pada web. Untuk melakukan edit sayur admin harus memilih sayur yang ingin diedit pada halaman manage sayur. Setelah itu akan ditampilkan halaman detail sayur yang telah dipilih. Admin dapat mengubah gambar, harga, stok dan satuan dengan menekan tombol  pada halaman detail sayur. Setelah menekan tombol edit maka akan ditampilkan form seperti form tambah sayur, admin diminta mengisi kolom yang ada dengan benar dan lengkap. Berikut tampilan edit sayur :



**Tampilan edit sayur.**

Edit sayur pada android :

Admin dapat melakukan edit sayur pada halaman home android. Pada halaman home admin dapat menekan tombol pada sayur yang ingin diubah. Setelah itu akan ditampilkan form untuk melakukan edit foto, nama, harga dan satuan dari sayur. Setelah admin mengisi formnya secara lengkap maka admin dapat menekan tombol untuk mengubah data sayur. Berikut tampilan ubah sayur android :



**Tampilan ubah sayur android.**

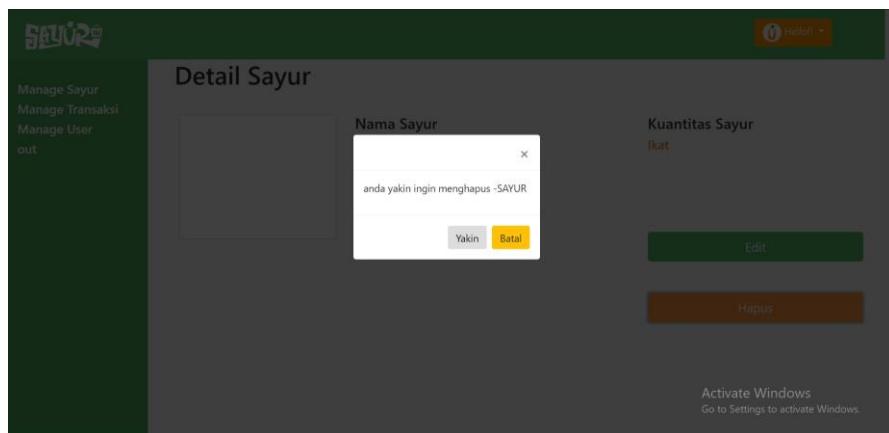
#### **1.1.4 Bagian IV – Hapus sayur**

Hapus sayur pada web :

Admin dapat menghapus sayur yang telah ditambahkan dengan menekan daftar sayur yang ingin dihapus, lalu akan ditampilkan tampilan detail sayur. Di dalam detail sayur admin hanya perlu menekan tombol

**Hapus**

lalu akan ditampilkan pop-up dan admin memilih “YAKIN” untuk melakukan penghapusan sayur. Berikut tampilan hapus sayur pada web :



**Tampilan hapus sayur.**

Hapus sayur pada android :

Untuk menghapus sayur pada android admin hanya perlu menekan tombol

**EDIT**

pada sayur yang ingin dihapus, lalu akan ditampilkan halaman edit sayur. Untuk menghapus sayur yang dipilih admin menekan tombol

**HAPUS**

dan sayur akan dihapus.

#### **1.1.5 Bagian V – Melihat dan menghapus daftar user.**

Melihat dan menghapus daftar *user* pada web :

Admin dapat melihat daftar *user* yang terdaftar di dalam sistem dengan memilih menu **Manage User**. Setelah itu akan ditampilkan tampilan manage *user* dengan bentuk tabel, di dalamnya admin dapat melihat daftar

*user* yang terdaftar. Untuk menghapus *user*, admin hanya diminta untuk

menekan tombol **Hapus** lalu akan muncul pop-up dan jika admin menekan pilihan “YAKIN” *user* akan dihapus dari sistem. Berikut tampilan manage user :

The screenshot shows a web-based application interface titled "Manage User". On the left, there is a sidebar with links: "Manage Sayur", "Manage Transaksi", and "Manage User". The main area is titled "User" and contains a table with two rows of data. The columns are labeled "Nama", "Email", "No Telp", and "Hapus". The first row has values "adinata baru 2", "adinata@gmail.com", "1234", and a "Hapus" button. The second row has values "Dwi", "dwi@gmail.com", "08123123123", and a "Hapus" button. At the bottom right of the table, there is a message: "Activate Windows Go to Settings to activate Windows.".

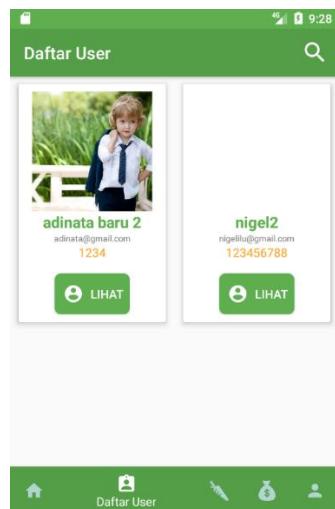
Nama	Email	No Telp	Hapus
adinata baru 2	adinata@gmail.com	1234	<b>Hapus</b>
Dwi	dwi@gmail.com	08123123123	<b>Hapus</b>

**Tampilan manage user.**

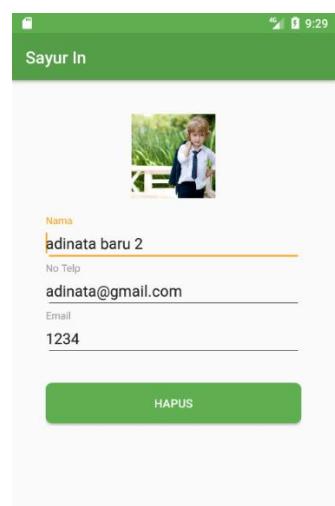
Melihat dan menghapus daftar *user* pada android :

Pada android untuk melihat daftar *user* admin memilih menu **Daftar User**, lalu akan ditampilkan tampilan daftar *user* yang terdaftar. Untuk menghapus *user*, admin perlu memilih *user* yang ingin dihapus lalu tekan tombol **LIHAT**. Setelah itu akan ditampilkan detail dari *user* dan admin menekan tombol **HAPUS** untuk menghapus *user* yang dipilih.

Berikut tampilan daftar *user* dan detail *user* :



Tampilan daftar user



Tampilan detail user

#### 1.1.6 Bagian VI – Melihat dan mengubah status transaksi.

Melihat dan mengubah status transaksi pada web :

Admin dapat melihat transaksi yang telah dilakukan *user* pada bagian menu **Manage Transaksi**. Di dalamnya akan ditampilkan daftar transaksi dalam bentuk tabel. Untuk melihat detail transaksinya, admin memilih daftar transaksi yang ingin dilihat. Setelah itu akan ditampilkan detail transaksinya. Untuk mengubah status transaksinya dari web, admin hanya

perlu mengubah pilihan pada bagian  dan setelah memilih status yang diinginkan admin perlu menekan tombol  untuk menyimpan kondisi atau status terbaru dari transaksi. Berikut tampilan daftar transaksi dan detail transaksi pada web :

Nama	Jumlah	Total belanja
Suastika Adinata	3	15000

Tampilan daftar transaksi

Nama user	Tanggal transaksi	
nomor handphone	alamatemail@email	
alamat pengiriman		
Pesanan		
Sayur	Jumlah	Total
bayam	3	15000
bayam	3	15000
Total		30000

Tampilan detail transaksi.

Melihat dan mengubah status transaksi pada android :

Admin dapat melihat transaksi yang dilakukan *user* dengan menekan menu

 , di dalamnya akan langsung ditampilkan daftar transaksi yang sedang “on progress” dan untuk melihat transaksi yang telah berhasil dapat dilihat pada tab “completed”. Untuk melihat detail transaksinya admin perlu menekan daftar yang ingin dipilih. Setelah detail transaksi muncul, untuk mengubah status dari transaksi tersebut admin hanya perlu

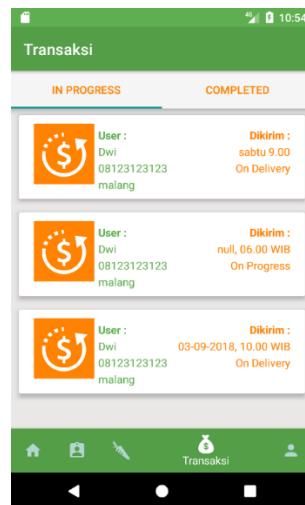
menekan salah satu dari status yang diinginkan. Jika memilih

ON DELIVERY

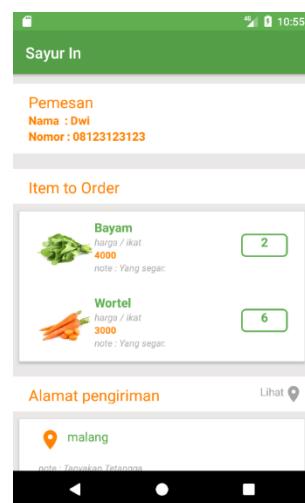
maka artinya transaksi sedang proses pengiriman, jika

COMPLETE

memilih maka artinya transaksi telah selesai dan daftar transaksi tersebut akan tampil pada bagian “completed”. Berikut tampilan daftar transaksi dan detail transaksi :



Tampilan daftar transaksi

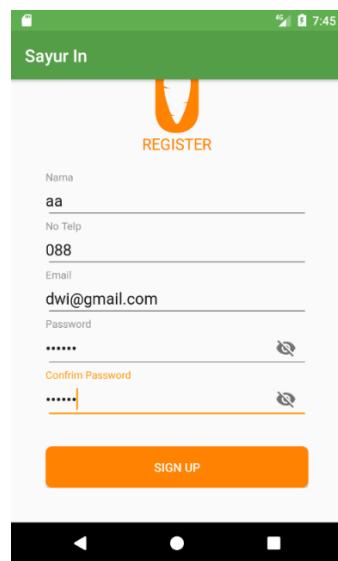


Tampilan detail transaksi

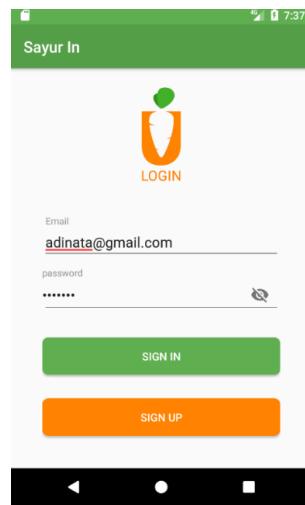
## 2.2. User

### 1.2.1 Bagian I – Login dan Register.

User yang belum terdaftar pada sistem dianjurkan melakukan *register* terlebih dahulu agar dapat menggunakan fitur dari aplikasi dengan menekan menu  , maka akan ditampilkan halaman login. Untuk register user harus menekan tombol  maka akan ditampilkan form register yang harus diisi, yaitu nama, nomor telepon,email, dan password. Jika pengguna telah mengisi form dengan lengkap dan format yang benar maka user perlu menekan tombol  untuk menyelesaikan proses register. Lalu, untuk melakukan *login* user hanya perlu memasukkan email dan password yang telah terdaftar pada form *login*, setelah itu menekan tombol  untuk melakukan proses *login*. Berikut tampilan halaman login dan register:

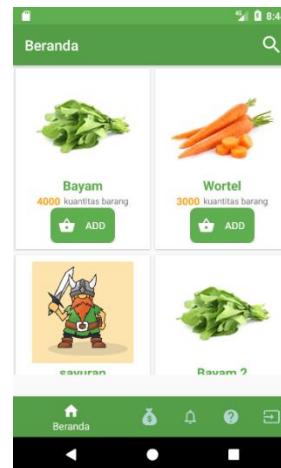


Tampilan halaman *register*



Tampilan halaman *login*

Setelah melalui proses register dan *user* berhasil *login*, maka akan ditampilkan halaman *home* yang berisi daftar sayur yang dapat dibeli. Berikut tampilan halaman *home* :



Tampilan halaman *home android*

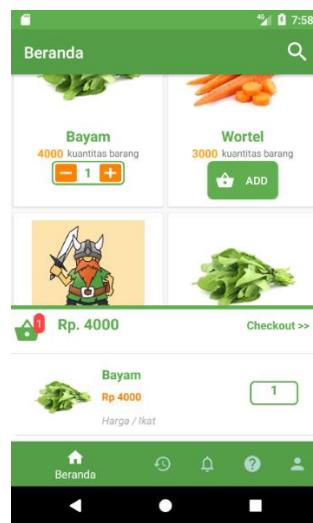
### 1.2.2 Bagian II – Memesan sayur

Jika *user* ingin melakukan pemesanan sayur pada aplikasi *user* dapat menekan tombol

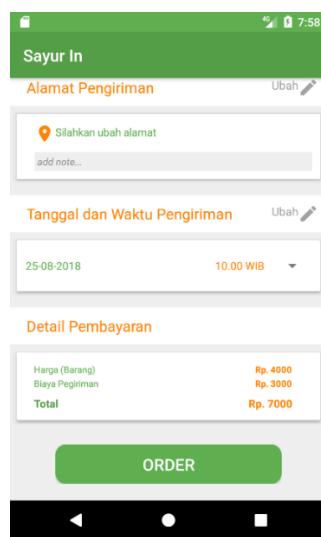


pada daftar sayur yang ingin dibeli, lalu akan muncul keranjang pada bagian bawah. *User* dapat mengatur jumlah sayur yang ingin dibeli dengan

mengatur jumlah pada bagian . Setelah proses pemilihan dan pengaturan sayur selesai untuk menuju proses selanjutnya untuk melengkapi data pengiriman, *user* harus menekan tombol [Checkout >>](#) pada bagian keranjang. Kemudian, *user* diminta mengisi data pengiriman seperti alamat dan waktu pengiriman dengan lengkap. Setelah *user* mengisi data tersebut dengan lengkap *user* menekan tombol [ORDER](#) untuk menyelesaikan proses pemesanan sayur. Berikut tampilan keranjang sayur dan halaman checkout :



**Tampilan keranjang sayur**

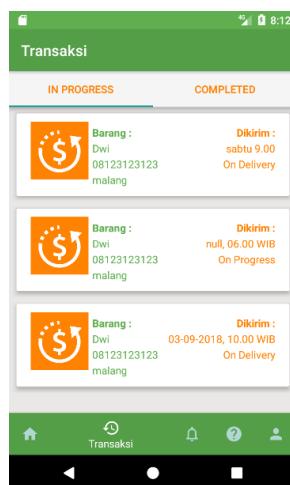


**Tampilan halaman *checkout***

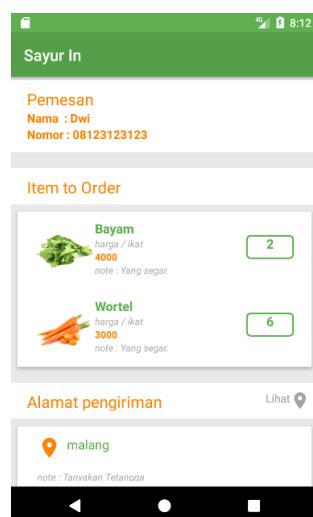
### 1.2.3 Bagian III – Melihat transaksi

Jika *user* telah melakukan transaksi, *user* dapat melihat transaksi yang telah

dilakukan dan memantau statusnya pada menu  Transaksi. Di dalamnya akan dibagi menjadi dua tab yaitu tab *on progress* berisi transaksi yang sedang dalam proses untuk dikirim dan tab *completed* berisi transaksi yang telah selesai dikirim. Untuk melihat detail dari transaksi yang telah dilakukan, *user* dapat menekan transaksi yang ada lalu akan ditampilkan detail transaksinya seperti sayur, jumlah sayur, alamat, waktu pengiriman dan total pembayannya. Berikut tampilan menu transaksi dan detail transaksi :



Tampilan menu transaksi

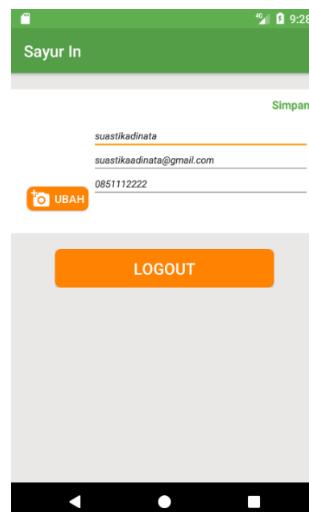


Tampilan detail transaksi

#### 1.2.4 Bagian IV – *Edit profil*

Untuk dapat melakukan proses ubah profil *user* harus menuju menu profil

yang bergambar  . Setelah itu *User* dapat langsung melakukan perubahan pada nama, email dan fotonya secara langsung, untuk merubah foto *user* perlu menekan tombol  UBAH lalu *user* memilih foto yang ingin disimpan. Jika telah melakukan perubahan pada bagian profil *user* dapat menekan tombol  Simpan untuk menyimpan perubahan yang telah dilakukan *user*. Berikut tampilan menu profil :



Tampilan menu profil

#### 1.2.5 Bagian V – *Logout*

Untuk keluar dari sistem *user* dapat menekan tombol  LOGOUT pada menu *profile*. Setelah proses *logout* selesai tampilan akan berubah ke halaman *home*.