

LAPORAN PRAKTIK KERJA LAPANG (PKL)

PERUSAHAAN

CV. SARANA UTAMA SOLUSINDO

SISTEM JUAL BELI SAYUR *ONLINE* BERBASIS *ANDROID*

MENGGUNAKAN *WEB SERVICE*

Diajukan untuk memenuhi sebagian persyaratan Kurikulum Sarjana



Disusun oleh:

Kadek Dwi Aryasa 155150201111124

Gusti Agung Suastika Adinata 155150201111273

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2018

PENGESAHAN

LAPORAN PRAKTIK KERJA LAPANGAN (PKL)
PERUSAHAAN
CV. SARANA UTAMA SOLUSINDO

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

Diajukan untuk memenuhi sebagian persyaratan Kurikulum Sarjana
Program Studi Teknik Informatika
Bidang Multimedia, Game dan Mobile

Disusun oleh:

Kadek Dwi Aryasa	155150201111124
Gusti Agung Suastika Adinata	1551502011111273

Praktik Kerja Lapangan ini dilaksanakan pada
1 Juli sampai dengan 30 Agustus 2017
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing PKL

Komang Candra Brata, S.Kom., M.T., M.Sc.
NIP: 201607 890711 1 000

Mengetahui,
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

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*

MENGUNAKAN *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 :

Pihak CV. SARANA UTAMA SOLUSINDO

Mengetahui,

General Manager

Menyetujui,

Pembimbing Lapangan

Sony Winarto

Selvina Yustriasanti

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam laporan PKL ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain dalam kegiatan akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam laporan PKL ini terbukti terdapat unsur-unsur plagiasi, saya bersedia PKL ini digugurkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 November 2018
Ketua Kelompok,

Gusti Agung Suastika Adinata
NIM: 155150201111273

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan PKL yang berjudul “Sistem Jual Beli Sayur *Online* Berbasis *Android* Menggunakan *Web Service*” ini dapat terselesaikan.

Penulis menyadari bahwa laporan ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Komang Candra Brata, S.Kom., M.T., M.Sc. selaku dosen pembimbing PKL yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan laporan ini.
2. Bapak Agus Wahyu Widodo, S.T, M.Cs. selaku ketua Program Studi Teknik Informatika.
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku ketua Jurusan Teknik Informatika.
4. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya laporan ini.
5. Seluruh civitas akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penyelesaian laporan PKL ini.

Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap PKL ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 26 November 2018
Ketua Kelompok,

Gusti Agung Suastika Adinata
Email: suastikaadinata@student.ub.ac.id

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 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 source* 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
PERNYATAAN ORISINALITAS.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vi
DAFTAR ISI	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah.....	2
1.6 Sistematika pembahasan	3
1.7 Pelaksanaan Praktek Kerja Lapang	4
BAB 2 PROFIL PERUSAHAAN.....	5
2.1 Sejarah dan Perkembangan CV Sarana Utama Solusindo (SASINDO).....	5
2.2 Visi, Misi dan Komitmen Perusahaan.	5
2.2.1 Visi.....	5
2.2.2 Misi	5
2.2.3 Komitmen.....	6
2.3 Struktur Organisasi CV SARANA UTAMA SOLUSINDO	6
2.4 Ruang Lingkup CV SARANA UTAMA SOLUSINDO	6
2.4.1 Solusi Bisnis TI (IT Business Solution) dan Konsultan IT (IT Consultant) 7	
2.4.2 Pendidikan dan Pelatihan (IT Training)	8

BAB 3 LANDASAN KEPUSTAKAAN	9
3.1 Sistem Penjualan	9
3.1.1 Sistem Jual Beli <i>Online</i>	9
3.2 <i>Android</i>	10
3.2.1 <i>Android Studio</i>	12
3.3 <i>Web Service</i>	13
3.4 <i>Unified Modeling Language (UML)</i>	14
3.4.1 <i>Structured Diagram</i>	14
3.4.2 <i>Behavior Diagram</i>	17
BAB 4 METODOLOGI	21
4.1 Penentuan Objek Penelitian	21
4.2 Studi Literatur	22
4.3 Analisa Kasus	22
4.4 Perancangan Konsep dan Model	22
4.5 Implementasi	23
4.6 Pengujian dan Analisis	23
4.7 Kesimpulan	23
BAB 5 ANALISIS DAN PERANCANGAN	24
5.1 Analisa Kebutuhan Sistem	25
5.1.1 Deskripsi Umum Sistem	25
5.1.2 Deskripsi Umum Perangkat Lunak	26
5.1.3 Fungsi Perangkat Lunak	26
5.1.4 Karakteristik Pengguna	26
5.2 Analisa Kebutuhan Perangkat Lunak	27
5.2.1 Kebutuhan Fungsional dan Non-Fungsional	28
5.2.2 Pemodelan Kebutuhan	30
5.3 Perancangan Perangkat Lunak	55
5.3.1 Perancangan Algoritma	55
5.3.2 Perancangan Antarmuka	59
5.3.2.1 Perancangan Tampilan Halaman <i>Login</i>	59

5.3.2.2 Perancangan Tampilan Halaman <i>Register</i>	60
5.3.2.3 Perancangan Tampilan Halaman <i>Home User</i>	60
5.3.2.4 Perancangan Tampilan Halaman Keranjang Belanja	61
5.3.2.5 Perancangan Tampilan Halaman Transaksi	62
5.3.2.6 Perancangan Tampilan Halaman Profil	63
5.3.2.7 Perancangan Tampilan Gudang Sayur	64
5.3.2.8 Perancangan Tampilan <i>Manage User</i>	65
5.3.2.9 Perancangan Tampilan <i>Manage Sayur</i>	66
5.3.2.10 Perancangan Tampilan <i>Manage Transaksi</i>	67
BAB 6 IMPLEMENTASI	68
6.1 Batasan Implementasi	68
6.2 Spesifikasi Sistem	69
6.2.1 Spesifikasi Perangkat Keras	69
6.2.2 Spesifikasi Perangkat Lunak	70
6.3 Implementasi Algoritma	70
6.3.1 Implementasi Algoritma Tambah Data Sayur	70
6.3.2 Implementasi Algoritma Lihat Data Sayur	72
6.3.3 Implementasi Algoritma Lihat Data <i>User</i>	74
6.4 Antarmuka	76
BAB 7 PENGUJIAN DAN ANALISIS	84
7.1 Pengujian	84
7.1.1 Pengujian Validasi	84
7.2 Analisis	94
7.2.1 Analisis Hasil Pengujian Validasi	94
BAB 8 PENUTUP	96
8.1 Kesimpulan	96
8.2 Saran	96
DAFTAR PUSTAKA	98
LAMPIRAN 1	99
LAMPIRAN 2	100

LOG BOOK..... 110

DAFTAR TABEL

Tabel 1.1 Kegiatan.....	4
Tabel 3.1 Notasi pada <i>Structured</i> diagram.....	16
Tabel 3.2 Notasi pada <i>Use Case Diagram</i>	17
Tabel 3.3 Notasi pada <i>Use Case Diagram</i> (lanjutan)	18
Tabel 3.4 Notasi pada <i>Sequence Diagram</i>	19
Tabel 3.5 Notasi pada <i>Sequence Diagram</i> (lanjutan).....	20
Tabel 5.1 Karakter Pengguna.....	27
Tabel 5.2 Kebutuhan Fungsional	28
Tabel 5.3 Kebutuhan Fungsional (lanjutan)	29
Tabel 5.4 Kebutuhan Fungsional (lanjutan)	30
Tabel 5.5 Kebutuhan Non-Fungsional Sistem	30
Tabel 5.6 Skenario <i>Use Case Sign in</i>	33
Tabel 5.7 Skenario <i>Use Case Sign up</i>	33
Tabel 5.8 Skenario <i>Use Case</i> Lihat daftar sayur pada <i>home</i>	34
Tabel 5.9 Skenario <i>Use Case</i> Tambah keranjang belanja.....	34
Tabel 5.10 Skenario <i>Use Case</i> hapus keranjang belanja	34
Tabel 5.11 Skenario <i>Use Case Search</i> sayur	35
Tabel 5.12 Skenario <i>Use Case</i> Lihat Keranjang belanja.....	35
Tabel 5.13 Skenario <i>Use Case Edit</i> jumlah sayur pada keranjang.	35
Tabel 5.14 Skenario <i>Use Case</i> Hapus daftar pembelian	36
Tabel 5.15 Skenario <i>Use Case</i> Melengkapi Data pengiriman	36
Tabel 5.16 Skenario <i>Use Case</i> Lihat transaksi.....	36
Tabel 5.17 Skenario <i>Use Case</i> Melihat detail transaksi.	37
Tabel 5.18 Skenario <i>Use Case</i> lihat bantuan	37
Tabel 5.19 Skenario <i>Use Case</i> lihat profil	37
Tabel 5.20 Skenario <i>Use Case Edit</i> Profil.....	38
Tabel 5.21 Skenario <i>Use Case Logout</i>	38

Tabel 5.22 Skenario <i>Use Case</i> lihat daftar sayur pada gudang.	38
Tabel 5.23 Skenario <i>Use Case</i> Menambah sayur.....	39
Tabel 5.24 Skenario <i>Use Case</i> Hapus Data sayur.....	39
Tabel 5.25 Skenario <i>Use Case Edit</i> data sayur.....	39
Tabel 5.26 Skenario <i>Use Case</i> Melihat daftar <i>user</i>	40
Tabel 5.27 Skenario <i>Use Case</i> Melihat data <i>user</i>	40
Tabel 5.28 Skenario <i>Use Case</i> Hapus data <i>user</i>	40
Tabel 5.29 Skenario <i>Use Case</i> Mengubah status transaksi.	41
Tabel 5.30 Skenario <i>Use Case</i> Lihat transaksi <i>On progress</i>	41
Tabel 5.31 Skenario <i>Use Case</i> Lihat transaksi <i>Completed</i>	41
Tabel 6.1 Spesifikasi Perangkat Keras	69
Tabel 6.2 Spesifikasi Perangkat Lunak	70
Tabel 6.3 Kode Program Tambah Data Sayur.....	71
Tabel 6.4 Kode Program Lihat Data Sayur.....	73
Tabel 6.5 Kode Program Lihat Data <i>User</i>	75
Tabel 7.1 Tabel Pengujian Validasi.....	85
Tabel 7.2 Tabel Pengujian Validasi (lanjutan)	86
Tabel 7.3 Tabel Pengujian Validasi (lanjutan)	87
Tabel 7.4 Tabel Pengujian Validasi (lanjutan)	88
Tabel 7.5 Tabel Pengujian Validasi (lanjutan)	89
Tabel 7.6 Tabel Pengujian Validasi (lanjutan)	90
Tabel 7.7 Tabel Pengujian Validasi (lanjutan)	91
Tabel 7.8 Tabel Pengujian Validasi (lanjutan)	92
Tabel 7.9 Tabel Pengujian Validasi (lanjutan)	93
Tabel 7.10 Tabel Pengujian Validasi (lanjutan).....	94

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi	6
Gambar 3.1. <i>Illustrasi Android Stack</i>	11
Gambar 3.2. <i>Class Diagram</i>	15
Gambar 3.3. <i>Package diagram</i>	15
Gambar 3.4. <i>Use Case Diagram</i>	17
Gambar 3.5. <i>Sequence Diagram</i>	19
Gambar 3.6. <i>Use Case Scenario</i>	20
Gambar 4.1 Diagram Alir Metodologi	21
Gambar 5.1 Diagram Alir Analisis dan Perancangan.....	24
Gambar 5.2 Deskripsi Sistem	25
Gambar 5.3 <i>Use Case Diagram</i> Dengan Aktor Guest.....	31
Gambar 5.4 <i>Use Case Diagram</i> Dengan Aktor User.....	31
Gambar 5.5 <i>Use Case Diagram</i> Dengan Aktor Admin.....	32
Gambar 5.6 <i>Class Diagram</i> Pada Bagian User	43
Gambar 5.7 <i>Class Diagram</i> Pada Bagian Admin	44
Gambar 5.8 <i>Class Diagram</i> Pada <i>Package</i> Admin Bagian 1	45
Gambar 5.9 <i>Class Diagram</i> Pada <i>Package</i> Admin Bagian 2	46
Gambar 5.10 <i>Class Diagram</i> Pada <i>Package</i> Admin Bagian 3	47
Gambar 5.11 <i>Class Diagram</i> Pada <i>Package Fragment</i> bagian 1	48
Gambar 5.12 <i>Detail Class Diagram</i> Pada <i>Package Fragment</i> bagian 2	49
Gambar 5.13 <i>Detail Class Diagram</i> Pada <i>Package Fragment</i> bagian 3	49
Gambar 5.14 <i>Detail Class Diagram</i> Pada <i>Package drawersayur</i>	50
Gambar 5.15 <i>Sequence Diagram</i> “menambah sayur”	51
Gambar 5.16 <i>Sequence Diagram</i> “perbarui data sayur”	52
Gambar 5.17 <i>Sequence Diagram</i> “hapus sayur”	52
Gambar 5.18 <i>Sequence Diagram</i> “hapus user”	53
Gambar 5.19 <i>Sequence Diagram</i> “search sayur”	54
Gambar 5.20 <i>Sequence Diagram</i> “pesan sayur”	54

Gambar 5.21 Diagram Alir Proses <i>Login</i> Sebagai Admin	55
Gambar 5.22 Diagram Alir Proses Tambah Data Sayur	56
Gambar 5.23 Diagram Alir Proses Lihat Data Sayur	57
Gambar 5.24 Diagram Alir Proses Perbarui Data Sayur	57
Gambar 5.25 Diagram Alir Proses Hapus Data Sayur	58
Gambar 5.26 Perancangan Tampilan Halaman <i>Login</i>	59
Gambar 5.27 Perancangan Tampilan Halaman <i>Register</i>	60
Gambar 5.28 Perancangan Tampilan Halaman <i>Home User</i>	61
Gambar 5.29 Perancangan Tampilan Halaman Keranjang Belanja	62
Gambar 5.30 Perancangan Tampilan Halaman Transaksi	63
Gambar 5.31 Perancangan Tampilan Halaman Profil	64
Gambar 5.32 Perancangan Tampilan Halaman Gudang Sayur	65
Gambar 5.33 Perancangan Tampilan Halaman <i>Manage User</i>	66
Gambar 5.34 Perancangan Tampilan Halaman Manage Sayur	66
Gambar 5.35 Perancangan Tampilan Halaman <i>Manage</i> Transaksi	67
Gambar 6.1 Diagram Alir Implementasi	68
Gambar 6.2 Implementasi Halaman <i>Login</i>	77
Gambar 6.3 Implementasi Halaman <i>Register</i>	77
Gambar 6.4 Implementasi Halaman <i>Home User</i>	78
Gambar 6.5 Implementasi Halaman Keranjang Belanja	79
Gambar 6.6 Implementasi Halaman Transaksi	80
Gambar 6.7 Implementasi Halaman Profil	80
Gambar 6.8 Implementasi Halaman Gudang Sayur	81
Gambar 6.9 Implementasi Halaman <i>Manage User</i>	82
Gambar 6.10 Implementasi Halaman <i>Manage</i> Sayur	82
Gambar 6.11 Implementasi Halaman <i>Manage</i> Transaksi	83
Gambar 7.1 Diagram Alir Pengujian dan Analisis	84

DAFTAR LAMPIRAN

LAMPIRAN 1.....	99
1. FOTO DOKUMENTASI KEGIATAN	99
LAMPIRAN 2.....	100
1. USER MANUAL	100
1.1 Admin	100
1.1.1 Bagian I – <i>Login</i>	100
1.1.2 Bagian II – Menambahkan sayur.....	101
1.1.3 Bagian III – <i>Edit</i> sayur.	101
1.1.4 Bagian IV – Hapus sayur	102
1.1.5 Bagian V – Melihat dan menghapus daftar <i>user</i>	102
1.1.6 Bagian VI – Melihat dan mengubah status transaksi.....	103
2.2. <i>User</i>	105
1.2.1 Bagian I – Login dan <i>Register</i>	105
1.2.2 Bagian II – Memesan sayur.....	106
1.2.3 Bagian III – Melihat transaksi.....	108
1.2.4 Bagian IV – <i>Edit</i> profil	109
1.2.5 Bagian V – <i>Logout</i>	109
LOG BOOK.....	110

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 penjual sayur belum mempunyai suatu media yang menyediakan layanan untuk menginformasikan tentang apa saja yang dijual secara langsung kepada konsumen. Tentunya sebagai konsumen harus mendatangi penjual untuk mengetahui sayur apa saja yang dijual dan untuk melakukan pembelian sayur. Dengan masalah tersebut akan menimbulkan ketidak-efisienan waktu dalam melakukan kegiatan berbelanja. Pembuatan sistem informasi dan penjualan sayur berbasis *android* ini merupakan solusi untuk menangani masalah tersebut.

Dalam perancangan proyek tim ini, penulis dan tim menggunakan bahasa pemrograman yang digunakan oleh *Android studio* yaitu *java* dan *xml*. Dengan menggunakan *Android Studio* akan dapat 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 pemrograman *java* , *xml* dan *php framework laravel* untuk *webservice*

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 serta saran-saran untuk pengembangan lebih lanjut.

1.7 Pelaksanaan Praktek Kerja Lapangan

- **Waktu**

PKL dilaksanakan mulai tanggal 1 April 2018 hingga tanggal 30 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 pengerjaan 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.

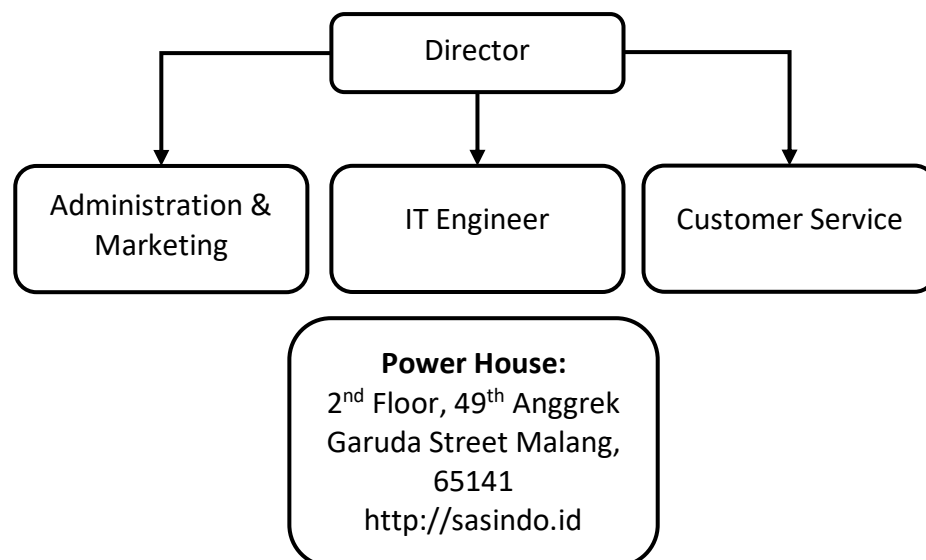
2. Memberikan solusi terbaik dalam setiap masalah yang dapat diselesaikan dengan TI.
3. Ikut serta memberikan kontribusi dalam perkembangan di dunia TI.

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*). Beberapa fokus dari CV. Sarana Utama Solusindo adalah:

1. Pengembangan Sistem Informasi Manajemen (*Management Information Sistem*)
 - 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*)
 - A0070likasi 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.

3.1.1 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;

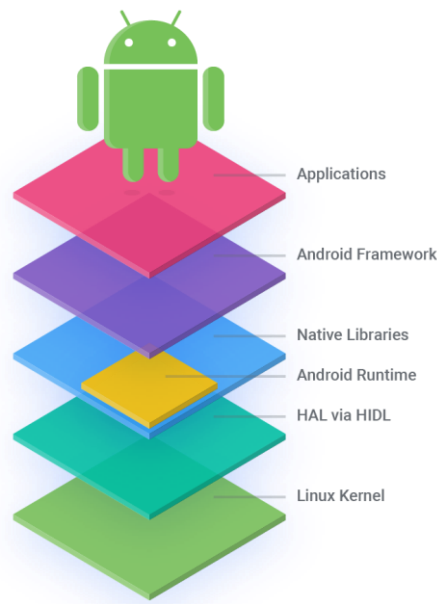
4. Memiliki kemampuan untuk menarik pelanggan baru;
5. Menyediakan satu titik lokasi kontak untuk bermacam-macam produk dan jasa;
6. Dapat digunakan sebagai media riset pasar.

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. Ilustrasi *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 Unified Modeling Language (UML)

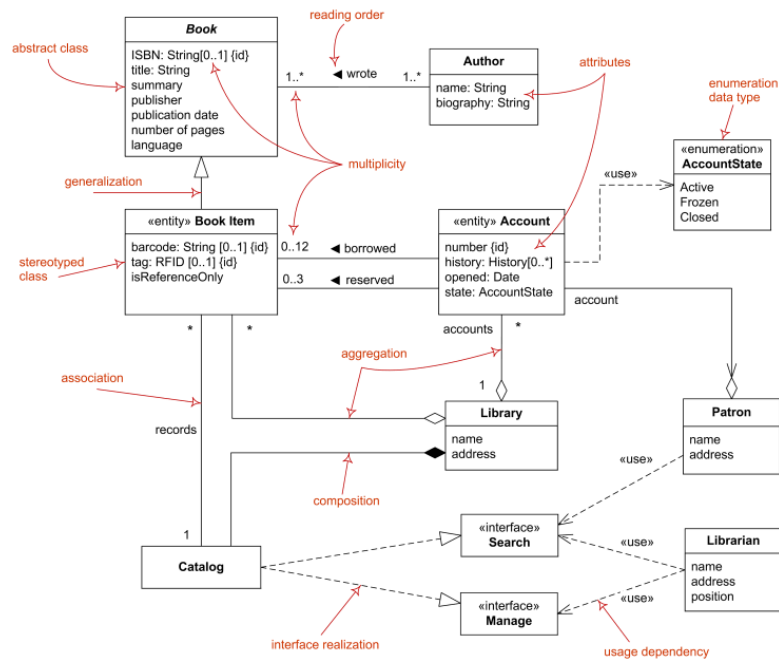
Modelling Language merupakan alat perancangan sistem yang berorientasi pada objek. *UML* diagram memiliki tujuan utama untuk membantu tim pengembangan proyek untuk dapat berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program (UML Diagrams,2018). *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.

3.4.1 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* :

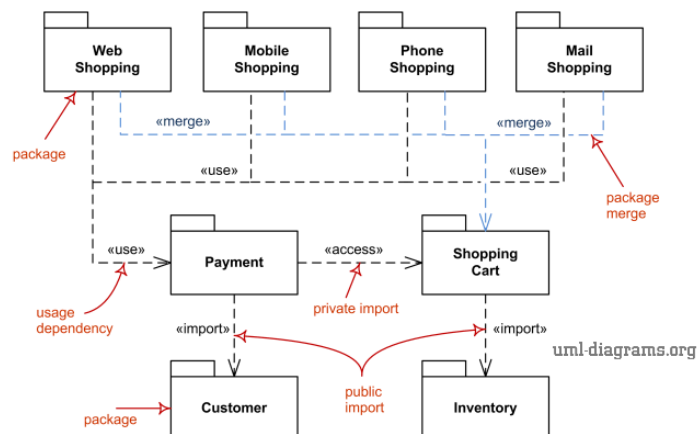


Gambar 3.2. Class Diagram

Sumber : UML Diagrams (2018)

2. Package Diagram.

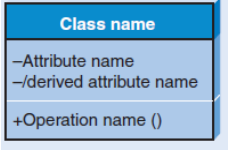
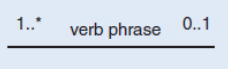


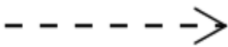

Package Diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Dalam penggunaan notasi pada *package diagram* memiliki persamaan dengan penggunaan pada *class diagram*. Berikut merupakan contoh dari *Pakcage Diagram* :



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.	
2	<i>Association</i>	Notasi yang digunakan untuk menunjukkan hubungan antar kelas, banyak kelas atau dengan kelas itu sendiri.	
3	<i>Package</i>	Notasi yang digunakan untuk menunjukkan package atau kumpulan 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.	
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.	

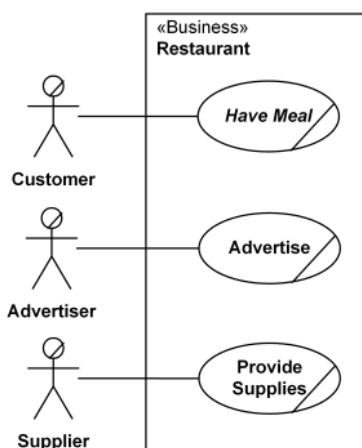
Sumber : Dennis (2012)

3.4.2 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* :



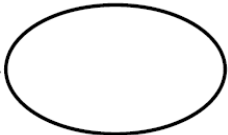

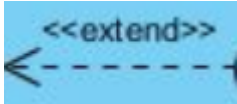
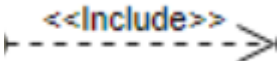
Gambar 3.4. Use Case Diagram

Sumber : UML Diagrams (2018)

Tabel 3.2 Notasi pada Use Case 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. Perlu dicatat bahwa <i>actor</i> berinteraksi dengan <i>use case</i> tetapi tidak memiliki kontrol atas <i>use case</i> .	

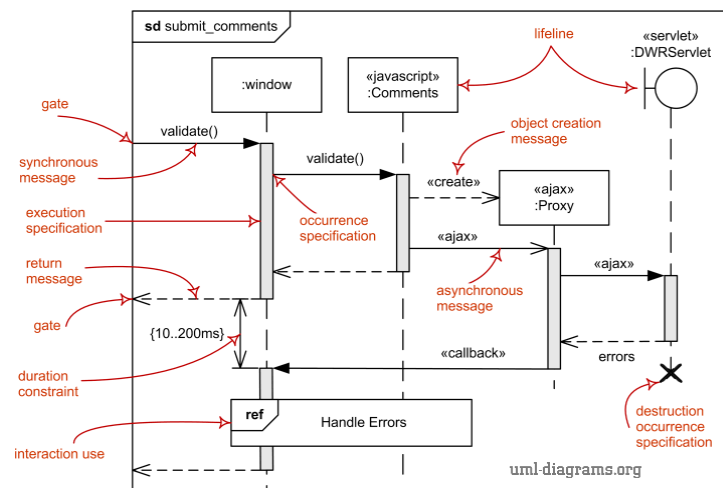
Tabel 3.3 Notasi pada *Use Case Diagram* (lanjutan)

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>Association</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* :





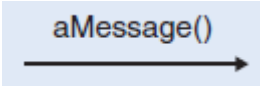
Gambar 3.5. Sequence Diagram

Sumber : UML Diagrams(2018)

Tabel 3.4 Notasi pada Sequence Diagram

No	Type	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.	
2	<i>Object</i>	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	
Brief Description	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"> • (Public) General Group: can only be created by an xxx User Administrator. • (Public) Article Group: will be created whenever a user is launched around an Article Available for User in the xxx environment.
Use Case Pre-Conditions	<ol style="list-style-type: none"> 1. The UGC has been identified by the system. 2. The UGC has the appropriate permission to create the user group.
Use Case Post-Conditions	<ol style="list-style-type: none"> 1. The user group has been created and is present and persistent in the system with the appropriate attributes.
List of Actors	<ul style="list-style-type: none"> • User Group Creator (UGC): <ol style="list-style-type: none"> 1. User Administrator 2. System
User Experience Links	
Basic Flow One: General Group	<ol style="list-style-type: none"> 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"> a. Title of user group (required) b. Description of user group (required)
Basic Flow Two: Article Group	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.
Alternate Flows	

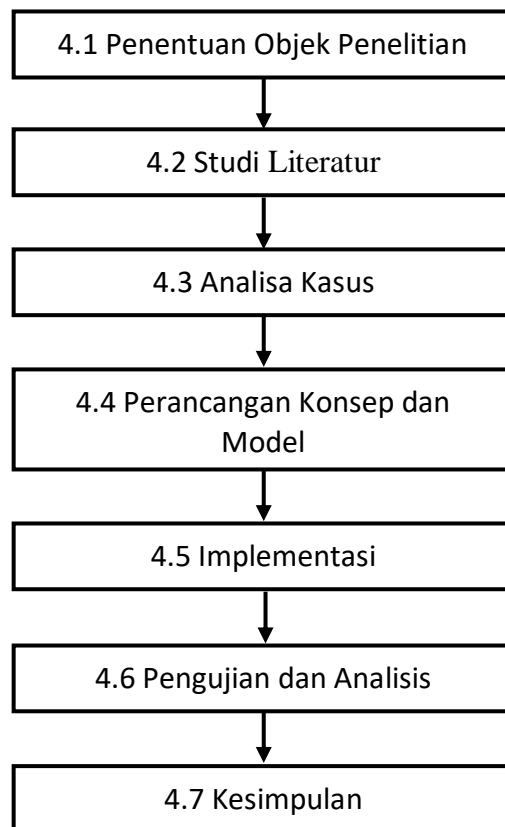
Gambar 3.6. *Use Case Scenario*.

Sumber : Bittner (2006)

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 penelitian 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

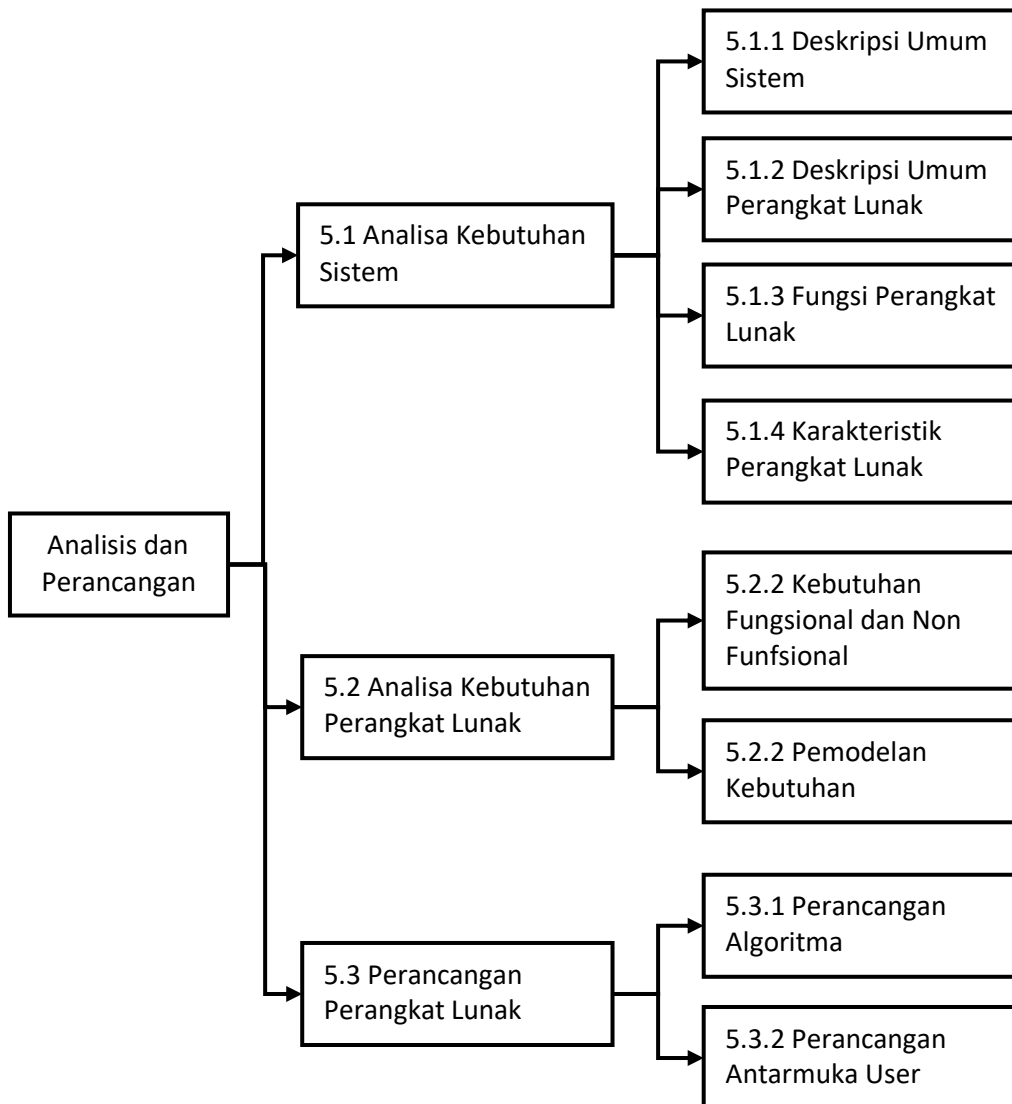
Pengujian yang akan digunakan yaitu pengujian *Black Box*. Pengujian *Black Box* merupakan pengujian fungsionalitas sistem untuk mengetahui apakah sistem telah berjalan dengan benar sesuai dengan parameter yang telah ditentukan. Uji coba ini dilakukan oleh tim internal dan juga dilakukan oleh calon pengguna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



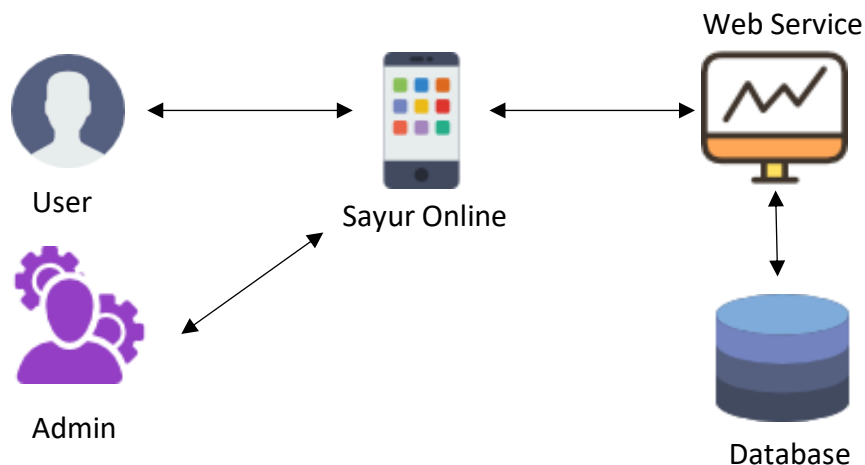
Gambar 5.1 Diagram Alir Analisis dan Perancangan

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 ditunjukkan pada tabel 5.1.

Tabel 5.1 Karakter Pengguna

No	Identifikasi Pengguna	Karakteristik
1	<i>User</i>	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 ditunjukkan pada tabel 5.2.

Tabel 5.2 Kebutuhan Fungsional

No	Aktor	Nama Fungsi	Deskripsi
1	User/Admin	Login	Untuk menyeleksi pengguna yang memiliki otorisasi 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 melihat 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 meruba 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 <i>User</i>	Admin dapat melakukan hapus terhadap data <i>user</i> yang tersimpan dalam sistem
12	<i>User</i> /Admin	<i>Edit</i> profil	<i>User</i> /admin dapat melakukan <i>edit profile</i> untuk kelengkapan data dari pengguna user ataupun admin
13	<i>User</i> /Admin	Lihat profil	<i>User</i> /Admin dapat melihat profil
14	<i>User</i>	<i>Register</i>	<i>User</i> dapat melakukan pendaftaran pada aplikasi sayur online untuk membeli sayur yang tersedia.
15	<i>User</i>	Order sayur	<i>User</i> dapat memilih dan melakukan penyimpanan data sayur yang akan dibeli ke dalam fitur <i>cart</i> .
16	<i>User</i>	Melengkapi data transaksi	<i>User</i> dapat melakukan pembelian sayur dengan melengkapi data transaksi pada keranjang.
17	<i>User</i>	<i>Edit</i> jumlah sayur pada keranjang	<i>User</i> dapat mengubah jumlah sayur yang terdapat di dalam keranjang.

Tabel 5.4 Kebutuhan Fungsional (lanjutan)

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

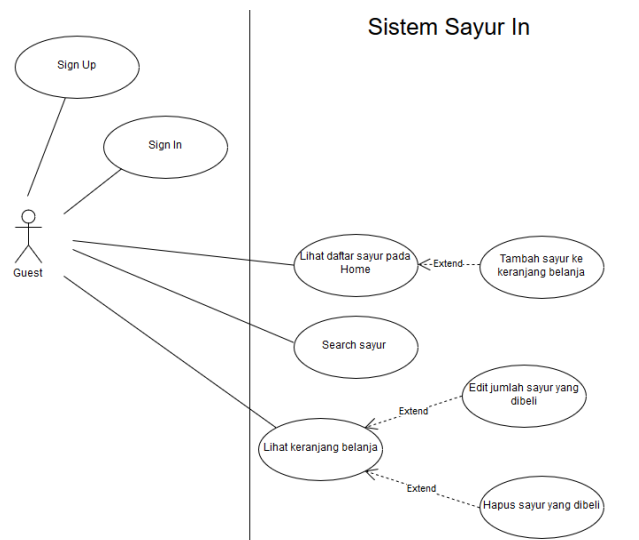
Tabel 5.5 Kebutuhan Non-Fungsional Sistem

No	Kebutuhan	Deskripsi
1	Availability	Dengan menguji ketersediaan 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 Use Case Diagram

Diagram *use case* digunakan untuk menggambarkan kebutuhan-kebutuhan dan fungsionalitas dari sistem serta digunakan untuk menunjukan aksi-aksi yang dilakukan oleh aktor terhadap sistem. Berikut adalah use case diagram dari aplikasi sayur online yang ditunjukkan pada gambar 5.3, 5.4, dan 5.5.



Gambar 5.3 Use Case Diagram Dengan Aktor Guest



Gambar 5.4 Use Case Diagram Dengan Aktor User



Gambar 5.5 Use Case Diagram Dengan Aktor Admin

5.2.2.2 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 yang ditunjukkan pada tabel 5.6 – 5.31.

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">1. Guest mengisi data pada kolom yang kosong,2. Data yang dimasukkan tervalidasi3. Menuju halaman home.
Skenario Alternatif	<ol style="list-style-type: none">1. Jika terdapat kolom kosong atau belum diisi sistem akan meminta melengkapi kolom sign in.
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">1. Menekan tombol <i>sign up</i>.2. Guest mengisi kolom dengan lengkap dan format yang benar.3. Guest menekan tombol sign up.4. Guest terdaftar dan dapat <i>login</i> pada halaman <i>sign in</i>.
Skenario Alternatif	<ol style="list-style-type: none">1. Jika terdapat kolom kosong atau belum diisi sistem akan meminta melengkapi kolom sign in.2. Jika Mengisi dengan format yang salah maka muncul pesan “format salah”
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	<i>User/Admin.</i>
Kondisi awal	Halaman <i>Home</i>
Skenario Utama	1. <i>Guest/User</i> Melihat daftar sayur pada <i>home</i> .
Skenario Alternatif	-
Kondisi akhir	<i>Guest/User</i> Melihat daftar sayur pada <i>home</i> .

Tabel 5.9 Skenario *Use Case* Tambah keranjang belanja

<i>Use case</i> Tambah keranjang belanja.	
Tujuan	Menambahkan Sayur pada keranjang belanja
Aktor	<i>User.</i>
Kondisi awal	Halaman <i>Home</i>
Skenario Utama	<ol style="list-style-type: none"> 1. <i>User</i> Menekan tombol “Add” pada sayur yang ingin dibeli lalu muncul keranjang pada bagian bawah halaman. 2. <i>User</i> 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	<i>User.</i>
Kondisi awal	Halaman <i>Home</i>
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	<i>Search Sayur</i>
Aktor	<i>Guest/User/Admin.</i>
Kondisi awal	Halaman <i>Home</i>
Skenario Utama	1. <i>Guest/User/Admin</i> 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	<i>Guest/User.</i>
Kondisi awal	Halaman <i>Home</i>
Skenario Utama	1. <i>Guest/User</i> 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	<i>User.</i>
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	<ol style="list-style-type: none"> 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	<i>User/Admin.</i>
Kondisi awal	Halaman <i>History</i> .
Skenario Utama	1. <i>User/Admin</i> 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	<i>User.</i>
Kondisi awal	Halaman <i>Home</i> .
Skenario Utama	1. <i>User</i> 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 <i>user</i> .
Aktor	<i>User/Admin.</i>
Kondisi awal	Halaman <i>Home</i> .
Skenario Utama	2. <i>User/Admin</i> 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	<i>User/Admin.</i>
Kondisi awal	Halaman Profil.
Skenario Utama	<ol style="list-style-type: none"> 1. <i>User</i> menekan icon camera untuk mengganti foto <i>user</i> 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 3. <i>User</i> menekan tombol simpan
Skenario Alternatif	-
Kondisi akhir	Profil berubah.

Tabel 5.21 Skenario Use Case Logout.

<i>Use case logout.</i>	
Tujuan	Keluar sistem.
Aktor	<i>User/Admin.</i>
Kondisi awal	Halaman Profil.
Skenario Utama	<ol style="list-style-type: none"> 1. <i>User</i> tombol logout.
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"> 1. Admin menekan tombol menu gudang.
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"> Admin menekan tombol menu gudang. Admin memilih sayur yang akan ditambahkan dengan menekan tombol "Add" Muncul pesan "Apakah anda yakin ingin menambahkan sayur?" Jika admin memilih "Iya" maka sayur ditambahkan ke halaman home.
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"> Admin menekan tombol lihat pada sayur. Admin menekan tombol hapus.
Skenario Alternatif	-
Kondisi akhir	Sayur berhasil dihapus.

Tabel 5.25 Skenario *Use Case* Edit data sayur

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

Tabel 5.26 Skenario Use Case Melihat daftar user

<i>Use case</i> Melihat daftar user.	
Tujuan	Melihat data user.
Aktor	Admin.
Kondisi awal	Halaman Home.
Skenario Utama	1. Admin menekan tombol daftar user pada menu.
Skenario Alternatif	-
Kondisi akhir	Menampilkan halaman daftar user.

Tabel 5.27 Skenario Use Case Melihat data user

<i>Use case</i> Melihat data user.	
Tujuan	Melihat detail data user.
Aktor	Admin.
Kondisi awal	Halaman daftar user.
Skenario Utama	1. Admin menekan tombol lihat pada daftar user.
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 user.
Aktor	Admin.
Kondisi awal	Halaman daftar user.
Skenario Utama	1. Admin menekan tombol lihat pada daftar user. 2. Admin menekan tombol hapus pada tampilan detail data user.
Skenario Alternatif	-
Kondisi akhir	Data user 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"> 1. Admin memilih daftar transaksi yang ingin diubah pada tab <i>On Progress</i>. 2. Memilih status transaksi yang tersedia pada bagian bawah halaman.
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	<i>User/Admin</i> .
Kondisi awal	Halaman <i>history</i> .
Skenario Utama	<ol style="list-style-type: none"> 1. Memilih tab <i>on progress</i> pada menu transaksi.
Skenario Alternatif	-
Kondisi akhir	Status transaksi berhasil berubah.

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

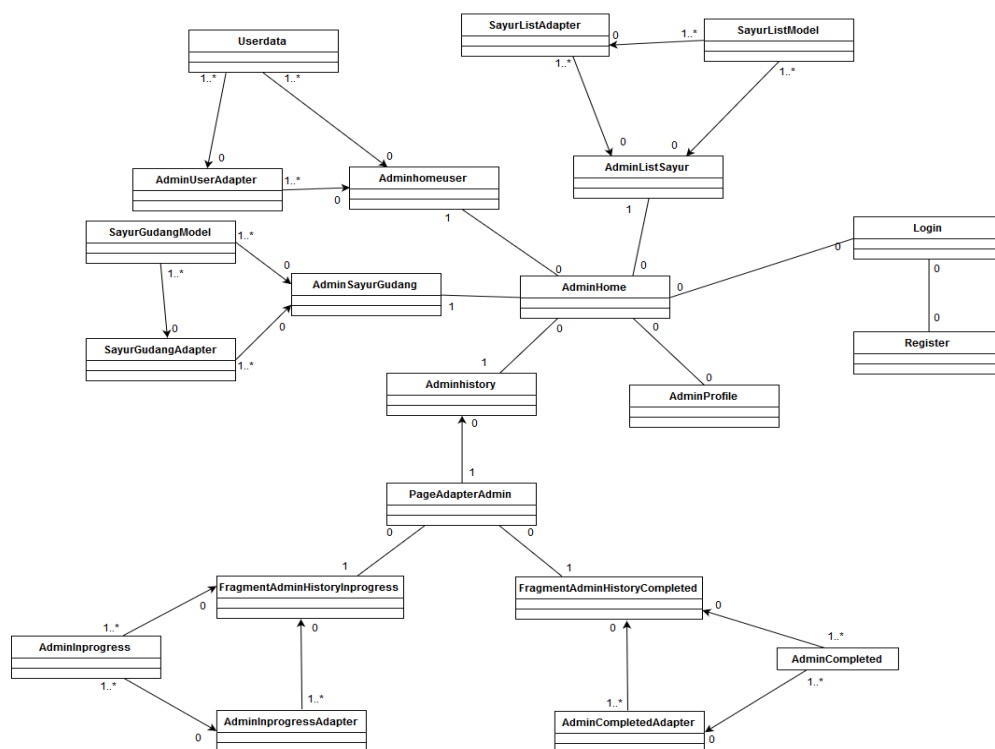
<i>Use case</i> Lihat transaksi <i>Completed</i> .	
Tujuan	Melihat transaksi yang berstatus <i>Completed</i>
Aktor	<i>User/Admin</i> .
Kondisi awal	Halaman <i>history</i> .
Skenario Utama	<ol style="list-style-type: none"> 1. Memilih tab <i>Completed</i> pada menu transaksi.
Skenario Alternatif	-
Kondisi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

5.2.2.3 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.

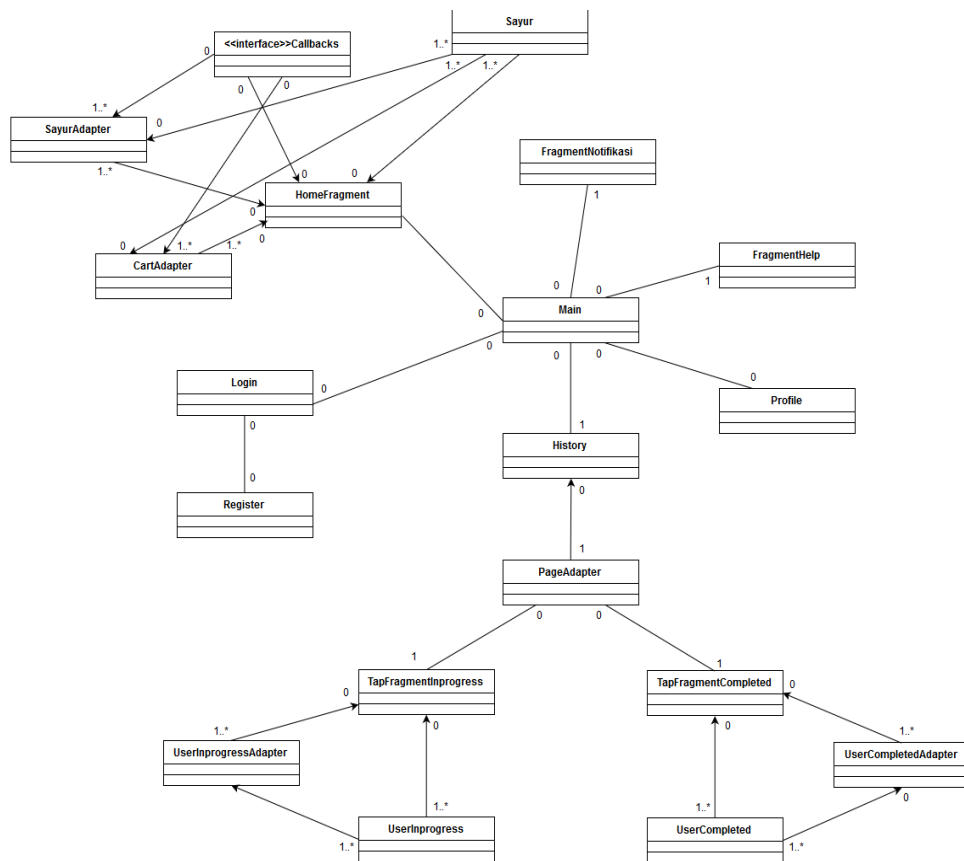
1. Class Diagram Pada Bagian User

Class diagram ini merupakan penjabaran dalam bentuk kelas dari use case pada gambar 5.4. 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 5.6 berikut ini.



2. *Class Diagram* Pada Bagian Admin

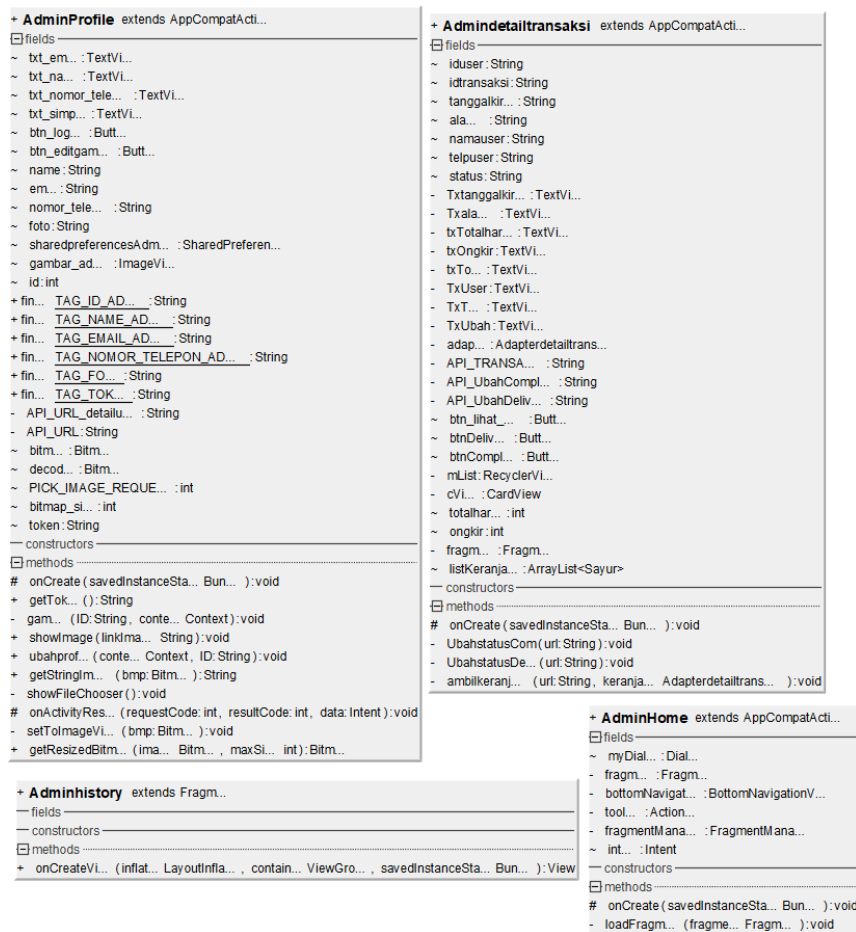
Class diagram ini merupakan penjabaran dalam bentuk kelas dari use case pada gambar 5.5. Pada *Class Diagram* ini terdapat 19 kelas yang saling berelasi dimana relasi antar kelas dapat dilihat dari tanda penghubung yang digunakan. Jika menggunakan tanda panah itu berarti kelas satu digunakan oleh kelas 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 5.7 berikut ini.



Gambar 5.7 Class Diagram Pada Bagian Admin

3. Class Diagram Pada Package Admin Bagian 1

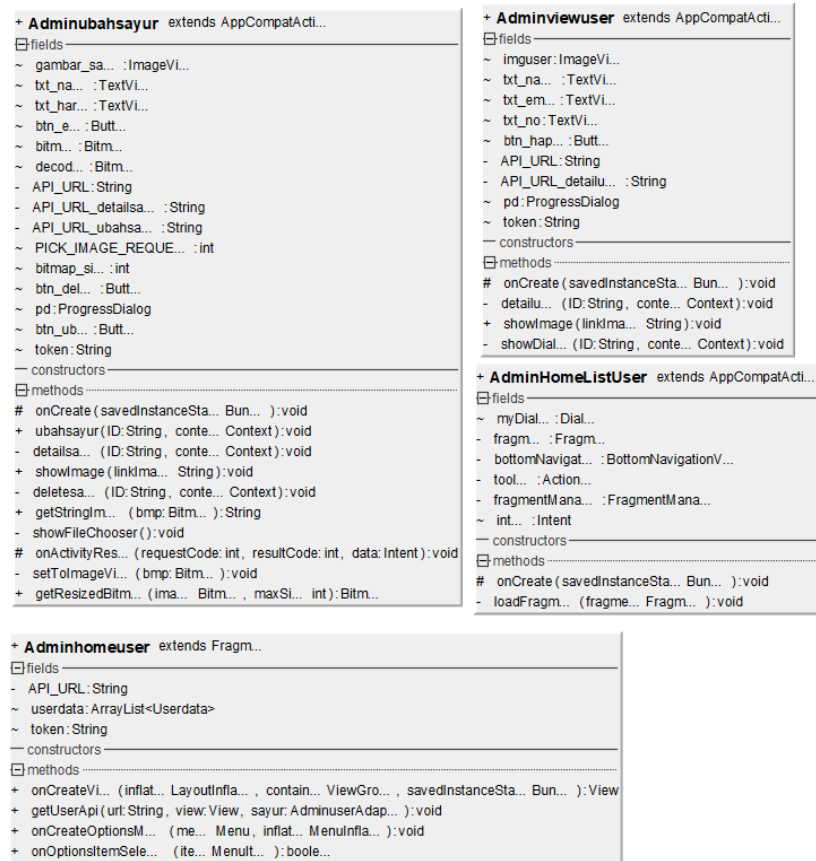
Pada Class Diagram pada package admin bagian 1 ini terdapat Class AdminProfile, AdminDetailtransaksi, Adminhistory dan AdminHome. Class Diagram ini merupakan detail Class Diagram dari Class Diagram yang ditunjukkan pada gambar 5.7. Gambar Class Diagram pada package admin bagian 1 dapat dilihat pada gambar 5.8 berikut ini.



Gambar 5.8 Class Diagram Pada Package Admin Bagian 1

4. Class Diagram Pada Package Admin Bagian 2

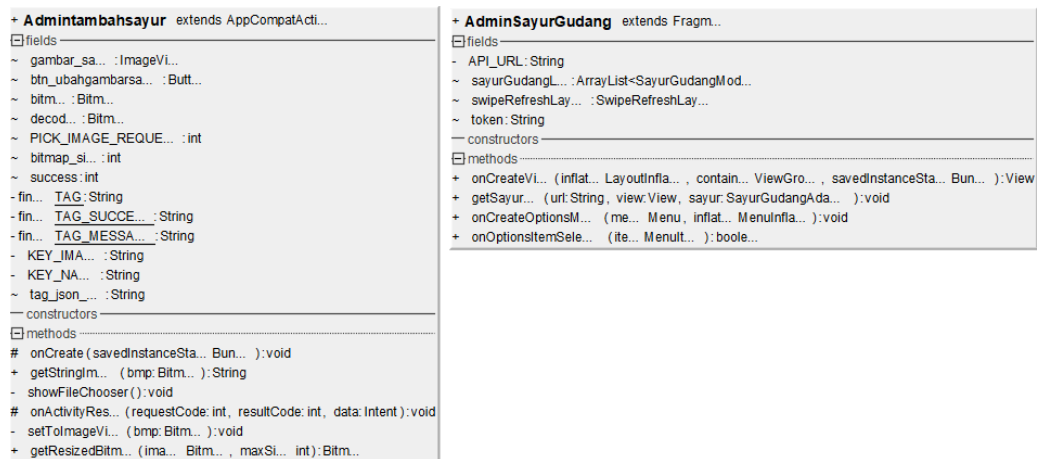
Pada Class Diagram pada package admin bagian 2 ini terdapat Class Adminubahsayur, Adminviewuser, AdminHomeListUser dan Admin-homeuser. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan pada gambar 5.7. Gambar *Class* pada *package* admin bagian 2 dapat dilihat pada gambar 5.9 berikut ini.



Gambar 5.9 Class Diagram Pada Package Admin Bagian 2

5. Class Diagram Pada Package Admin Bagian 3

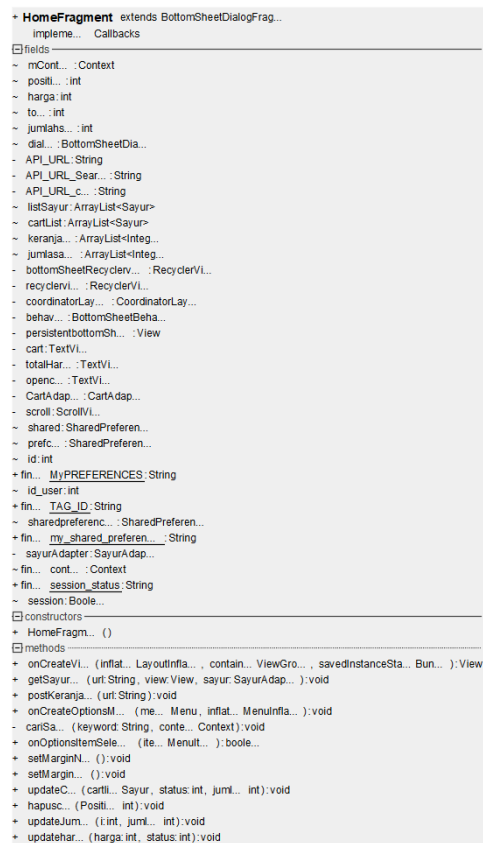
Pada Class Diagram pada package admin bagian 2 ini terdapat Class Admintambahsayur dan AdminSayurGudang. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan pada gambar 5.7. Gambar *Class Diagram* pada package admin bagian 3 dapat dilihat pada gambar 5.10 berikut ini.



Gambar 5.10 Class Diagram Pada Package Admin Bagian 3

6. Class Diagram Pada Package Fragment Bagian 1

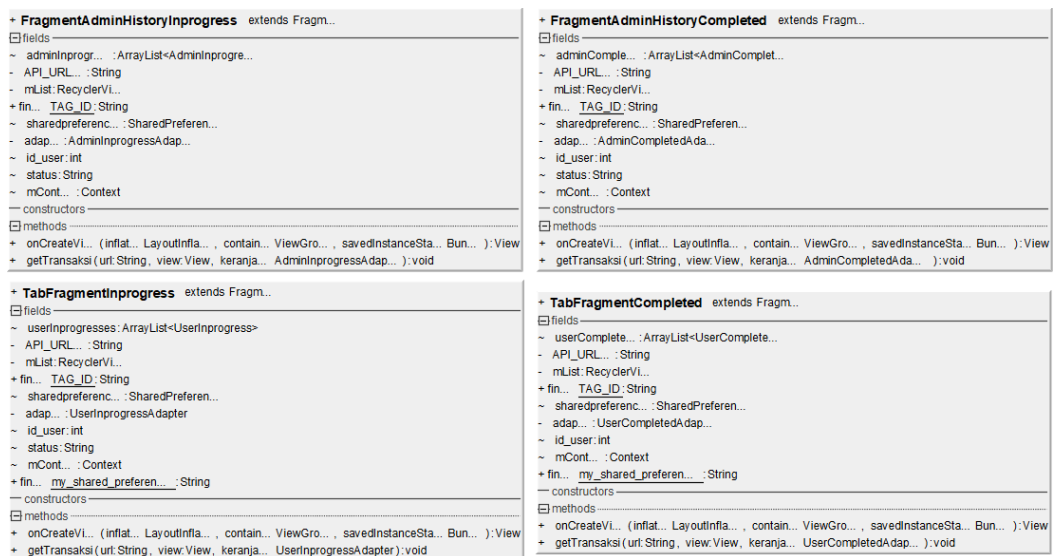
Pada Class Diagram pada package fragment bagian 1 ini terdapat Class HomeFragment. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan pada gambar 5.7. Gambar *Class Diagram* pada *package fragment* bagian 1 dapat dilihat pada gambar 5.11 berikut ini.



Gambar 5.11 Class Diagram Pada Package Fragment bagian 1

7. Class Diagram Pada Package Fragment Bagian 2

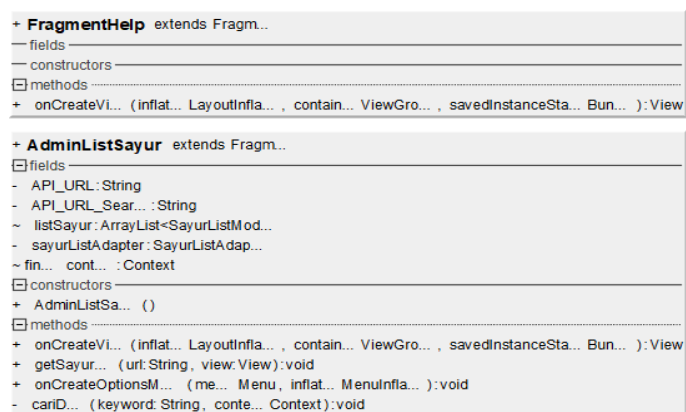
Pada Class Diagram pada package fragment bagian 2 ini terdapat Class FragmentAdminHistoryInProgress, FragmentAdminHistoryCompleted, TabFragmentInProgress, TabFragmentCompleted. Untuk kelas FragmentAdminHistoryInProgress dan FragmentAdminHistoryCompleted merupakan detail dari *Class Diagram* yang ditunjukkan pada gambar 5.7. Sedangkan untuk kelas TabFragmentInProgress dan TabFragmentCompleted merupakan detail dari *Class Diagram* yang ditunjukkan pada gambar 5.6. Gambar *Class Diagram* pada *package fragment* bagian 2 dapat dilihat pada gambar 5.12. berikut ini.



Gambar 5.12 Detail Class Diagram Pada Package Fragment bagian 2

8. Class Diagram Pada Package Fragment Bagian 3

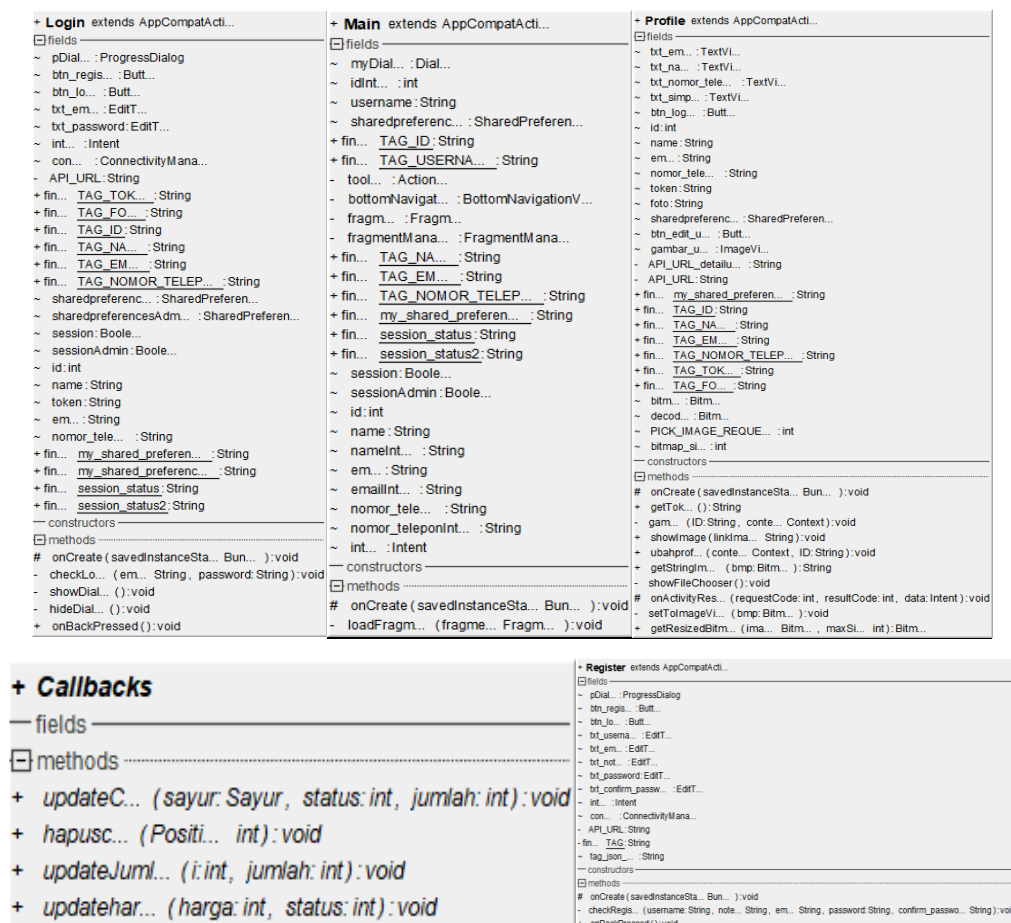
Pada Class Diagram pada package fragment bagian 3 ini terdapat Class FragmentHelp, AdminListSayur. *Class Diagram* ini merupakan detail *Class Diagram* dari *Class Diagram* yang ditunjukkan dari pada gambar 5.6. Gambar *Class Diagram* pada package *fragment* bagian 3 dapat dilihat pada gambar 5.13 berikut ini.



Gambar 5.13 Detail Class Diagram Pada Package Fragment bagian 3

9. Class Diagram 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 5.6, kemudian untuk kelas Login, Profile, Register merupakan detail Class Diagram dari Class Diagram yang ditunjukkan pada gambar 5.6 dan 5.7. Gambar *Class Diagram* pada *package drawersayur* dapat dilihat pada gambar 5.14 berikut ini.



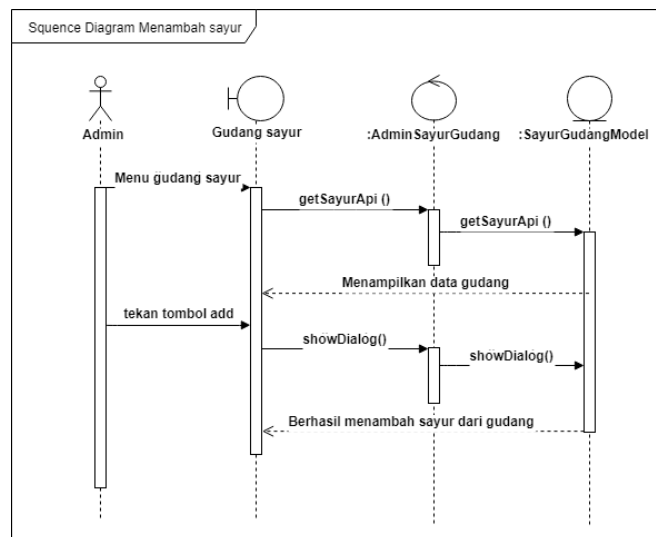
Gambar 5.14 Detail Class Diagram Pada Package drawersayur

5.2.2.4 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

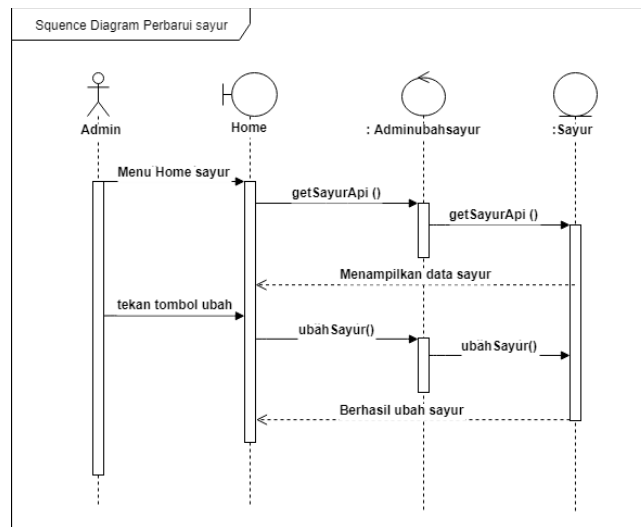
Sequence Diagram perbarui data sayur merupakan interaksi antara objek untuk memperbarui/meng-*edit* data sayur. *Sequence diagram* ini merupakan penjabaran dari use case “Tambah sayur dari gudang” yang terdapat pada gambar 5.4. Berikut Gambar 5.15 *sequence diagram* dari menambah sayur:



Gambar 5.15 *Sequence Diagram* “menambah sayur”

2. *Sequence Diagram* perbarui data sayur

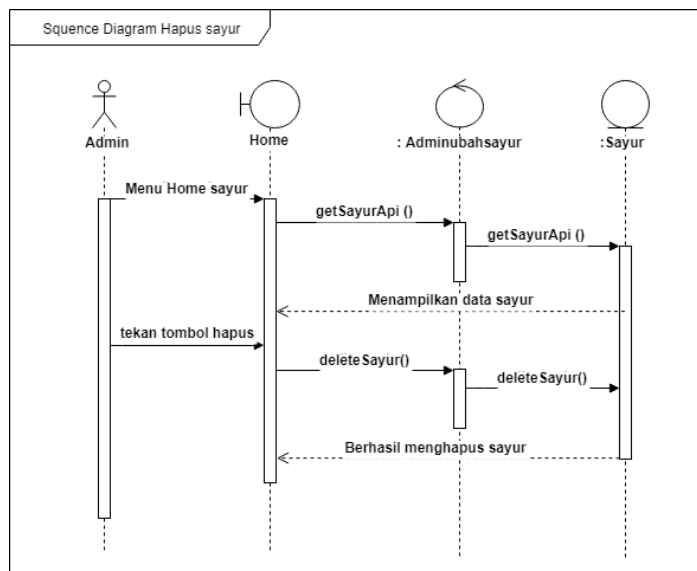
Sequence Diagram perbarui data sayur merupakan interaksi antara objek untuk memperbarui/meng-*edit* data sayur. *Sequence diagram* ini merupakan penjabaran dari use case “Perbaharui data sayur” yang terdapat pada gambar 5.4. Berikut Gambar 5.16 *sequence diagram* dari perbaharui data sayur:



Gambar 5.16 Sequence Diagram “perbarui data sayur”

3. Sequence Diagram hapus sayur

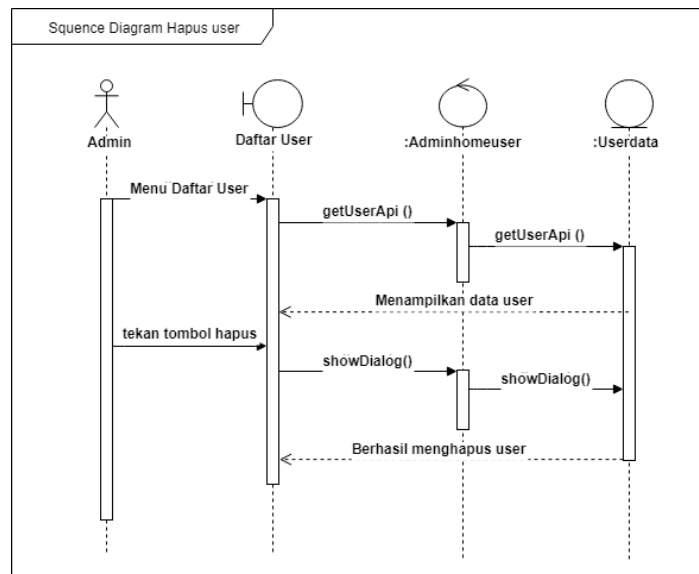
Sequence Diagram hapus sayur merupakan interaksi antara objek untuk menghapus data sayur. Sequence diagram ini merupakan penjabaran dari use case “Hapus data sayur” yang terdapat pada gambar 5.5. Berikut Gambar 5.17 sequence diagram dari hapus sayur:



Gambar 5.17 Sequence Diagram “hapus sayur”

4. Sequence Diagram Hapus User

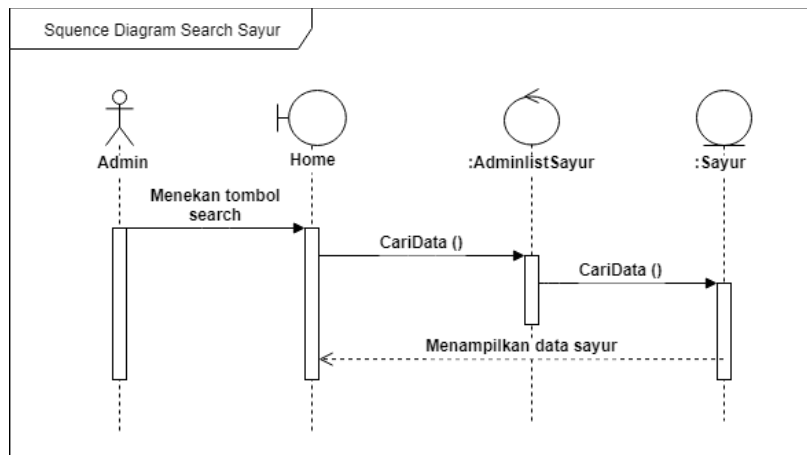
Sequence Diagram hapus user merupakan interaksi antara objek untuk menghapus *user* yang terdaftar. *Sequence diagram* ini merupakan penjabaran dari use case “Hapus data *user*” yang terdapat pada gambar 5.5. Berikut Gambar 5.18 *sequence diagram* dari hapus user:



Gambar 5.18 Sequence Diagram “hapus user”

5. Sequence Diagram Search Sayur

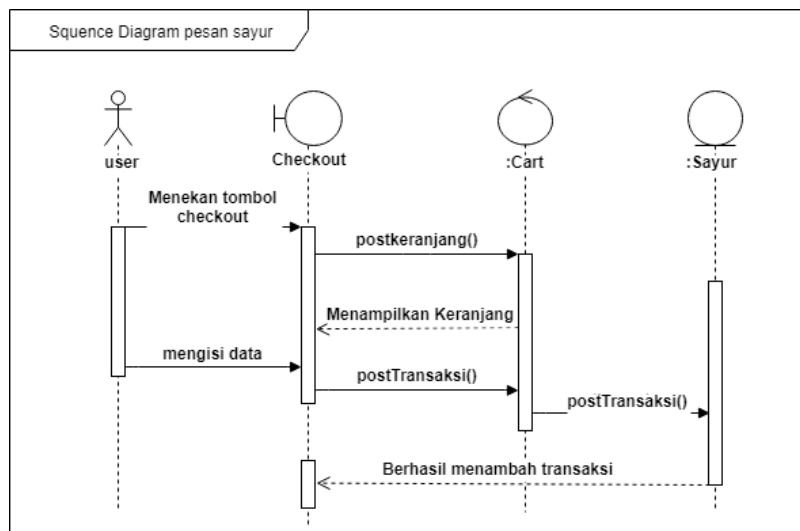
Sequence Diagram search sayur merupakan interaksi antara object untuk mencari data sayur berdasarkan nama sayur. *Sequence diagram* ini merupakan penjabaran dari use case “Search sayur” yang terdapat pada gambar 5.4 dan 5.6. Berikut Gambar 5.19 *sequence diagram* dari search sayur:



Gambar 5.19 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 5.4. Berikut Gambar 5.20 *sequence diagram* dari pesan sayur:



Gambar 5.20 Sequence Diagram “pesan sayur”

5.3 Perancangan Perangkat Lunak

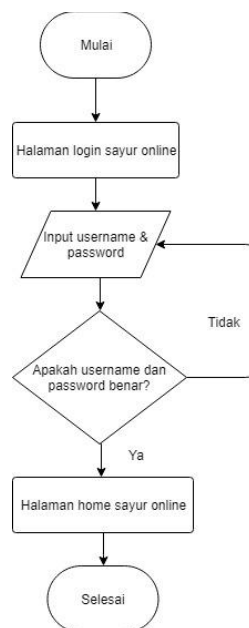
Dalam pembuatan Sistem Jual Beli Sayur *Online* ini diperlukan tahap perancangan perangkat lunak yang terdiri dari perancangan algoritma, perancangan data dan perancangan antarmuka *user*. Masing-masing tahap tersebut dijelaskan pada bagian perancangan berikut ini.

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*

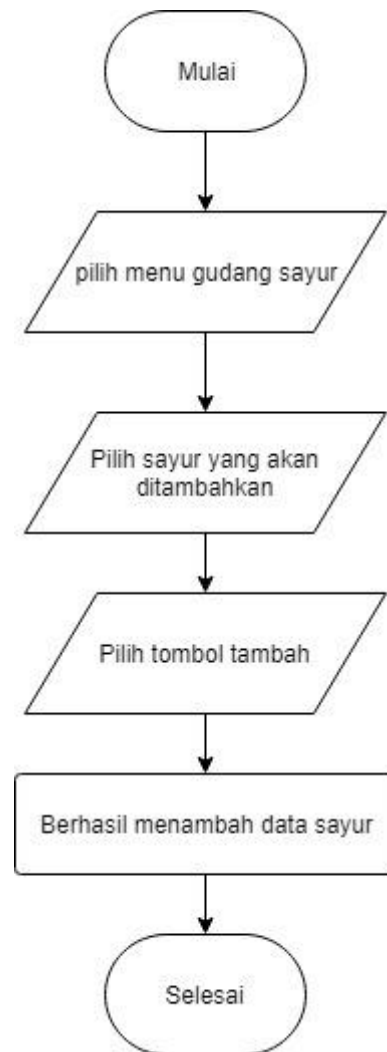
Diagram alir proses login sebagai *user* ini merupakan gambaran secara visual dari algoritma login, Admin melakukan *input username* dan *password*. Diagram alir ini merupakan penjabaran dari *use case "Sign in"* yang terdapat pada gambar 5.3. Berikut gambar 5.21 diagram proses alir proses *login*.



Gambar 5.21 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 5.4. Berikut gambar 5.22 diagram alir proses tambah data sayur.



Gambar 5.22 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 5.3, 5.4, dan 5.5. Berikut gambar 5.23 diagram alir proses lihat data sayur.



Gambar 5.23 Diagram Alir Proses Lihat Data Sayur

5.3.1.4 Diagram Alir Proses Perbarui Data Sayur

Diagram alir proses perbarui data sayur ini merupakan gambaran secara visual dari algoritma proses perbarui data sayur, admin melakukan perbarui data sayur pada form manajemen sayur. Diagram alir ini merupakan penjabaran dari *use case* “Perbaharui data sayur” yang terdapat pada gambar 5.5. Berikut gambar 5.24 diagram proses alir proses perbaharui data sayur.



Gambar 5.24 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 5.5. Berikut gambar 5.25 diagram proses alir proses hapus data sayur.



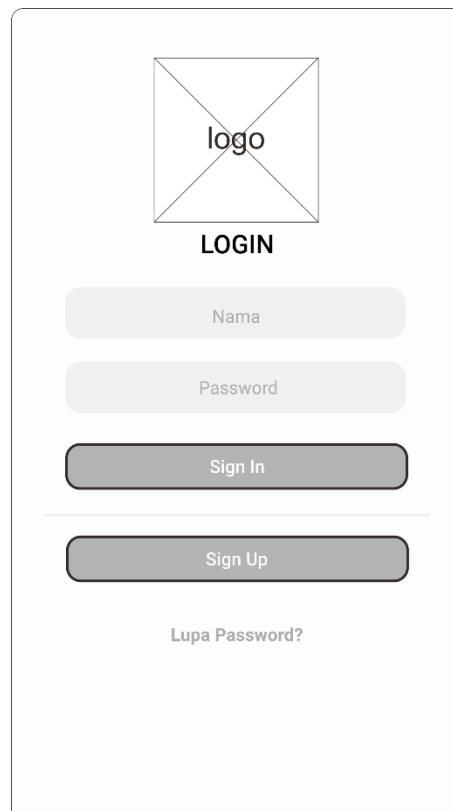
Gambar 5.25 Diagram Alir Proses Hapus Data Sayur

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.

5.3.2.1 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 5.26 berikut ini menunjukkan halaman login.

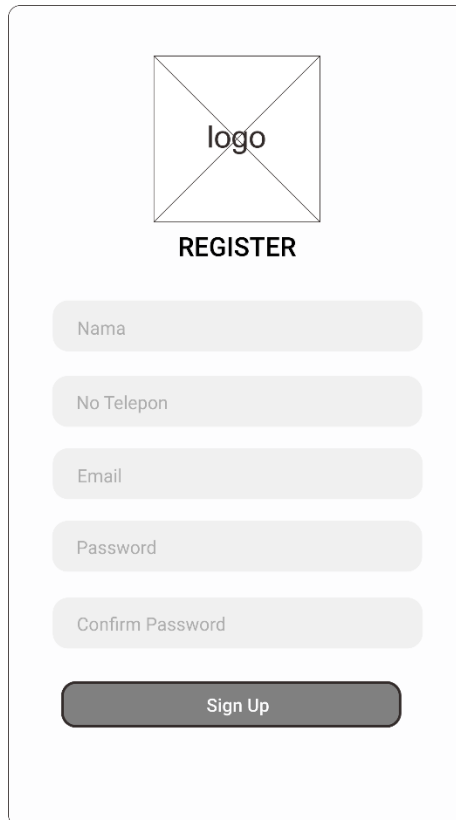


The image shows a wireframe of a login page. At the top, there is a square placeholder for a logo with the word "logo" in the center. Below the logo is the word "LOGIN" in bold. Underneath "LOGIN" are two input fields: one labeled "Nama" and another labeled "Password". Below these fields is a "Sign In" button. A horizontal line separates the "Sign In" button from a "Sign Up" button below it. At the bottom of the form is a link labeled "Lupa Password?".

Gambar 5.26 Perancangan Tampilan Halaman *Login*

5.3.2.2 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 5.27 berikut ini menunjukkan halaman *register*.

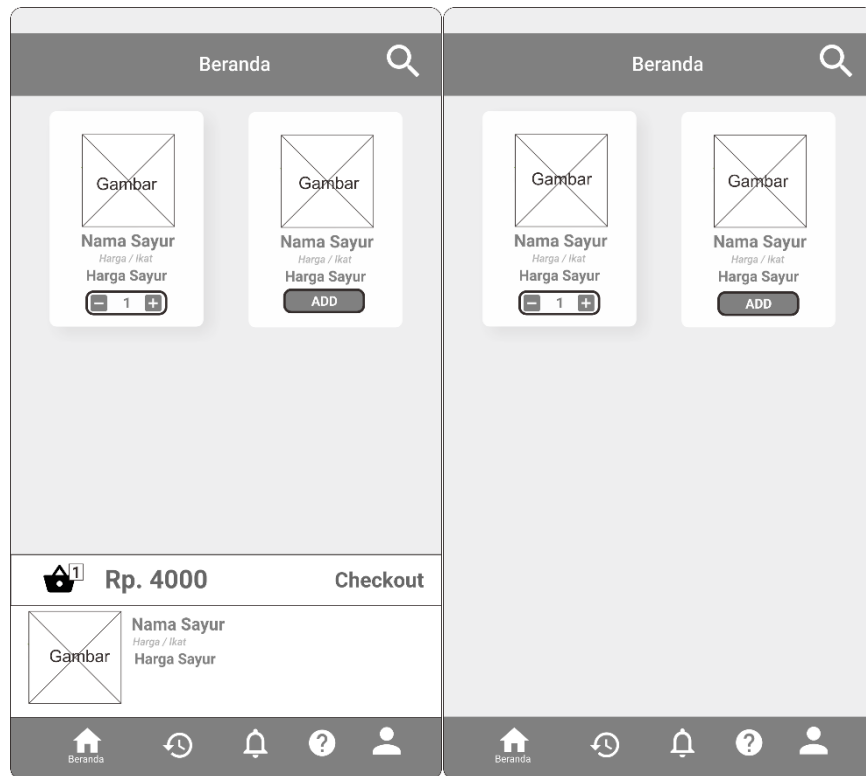


Gambar 5.27 Perancangan Tampilan Halaman *Register*

5.3.2.3 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 di jual 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 5.28 berikut ini menunjukkan halaman *Home User*



Gambar 5.28 Perancangan Tampilan Halaman *Home User*

5.3.2.4 Perancangan Tampilan Halaman Keranjang Belanja

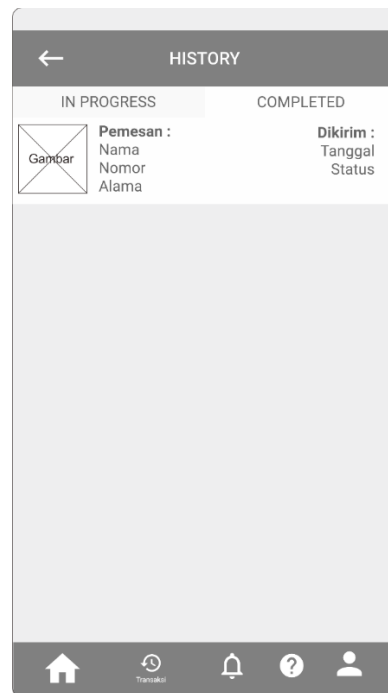
Tampilan halaman keranjang belanja berisi tampilan yang muncul ketika *user* menekan tombol checkout. Halaman ini akan menampilkan data sayur, total harga sayur yang akan dibeli dan untuk melengkapi data pengiriman, di halaman keranjang belanja ini terdapat *button order* untuk masuk ke proses pembelian sayur. Pada Gambar 5.29 berikut ini menunjukkan halaman Keranjang Belanja.

The screenshot shows a mobile application interface for a checkout process. The title bar at the top is dark gray with a back arrow on the left and the word 'Checkout' in the center. Below the title bar, there's a section titled 'Item to Order'. It contains a list item for 'Nama Sayur' (Vegetable Name) with a placeholder image labeled 'Gambar', a quantity of '1', and a price of 'Rp. 12.000,-'. Below the item name are fields for 'Harga / Berat' (Price / Weight) and 'Harga' (Price), with an 'Add Note...' button. The next section is 'Alamat pengiriman' (Shipping Address) with a location pin icon, a text field for 'Alamat.', and an 'Add Note...' button. Below that is 'Waktu Pengiriman' (Shipping Time) with a text field for 'Tanggal kirim' (Shipping Date) showing '08.00 WIB' and a 'Ubah' (Change) button. The 'Detail Pembayaran' (Payment Details) section shows a table with three rows: 'Harga (Barang)' (Price of Goods) at 'Rp. 12.000,-', 'Biaya pengiriman' (Shipping Fee) at 'Rp. 3.000,-', and 'Total' at 'Rp. 15.000,-'. At the bottom of the screen is a large, dark gray button with the word 'ORDER' in white capital letters.

Gambar 5.29 Perancangan Tampilan Halaman Keranjang Belanja

5.3.2.5 Perancangan Tampilan Halaman Transaksi

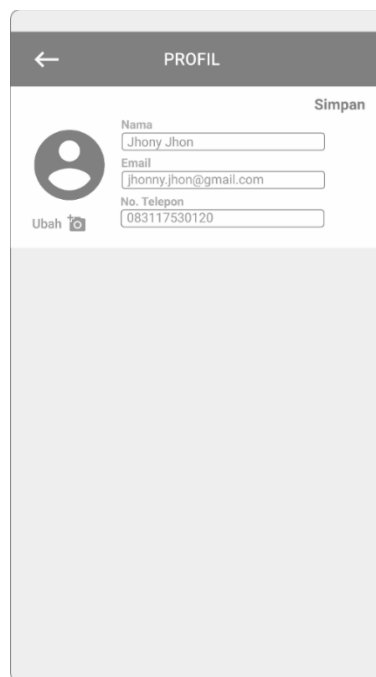
Tampilan halaman Transaksi belanja berisi tampilan yang muncul ketika *user* menekan tombol order. Halaman ini akan menampilkan data sayur, total sayur yang sebelumnya telah di order, jika barang sudah terkirim maka data akan berpindah ke tab *completed*. Pada Gambar 5.30 berikut ini menunjukkan halaman transaksi.



Gambar 5.30 Perancangan Tampilan Halaman Transaksi

5.3.2.6 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 yang berhasil login. Di halaman ini *user* dapat merubah nama , *email* , no telepon dan foto. Pada Gambar 5.31 berikut ini menunjukkan halaman Profil.

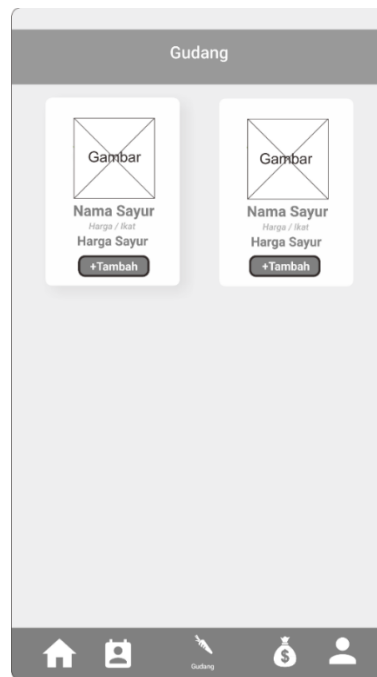


The image shows a mobile application interface for a user profile. At the top, there is a dark grey header bar with a white back arrow on the left and the word "PROFIL" in white text on the right. Below the header, the page has a light grey background. On the left side, there is a circular profile picture placeholder with a person icon. To the right of the profile picture, there are three input fields for user information: "Nama" (Name) with the value "Jhony Jhon", "Email" with the value "jhony.jhon@gmail.com", and "No. Telepon" (Phone Number) with the value "083117530120". Below the profile picture, there is a small "Ubah" (Change) button with a camera icon. To the right of the input fields, there is a "Simpan" (Save) button. The bottom half of the page is a large, empty light grey rectangular area.

Gambar 5.31 Perancangan Tampilan Halaman Profil

5.3.2.7 Perancangan Tampilan Gudang Sayur

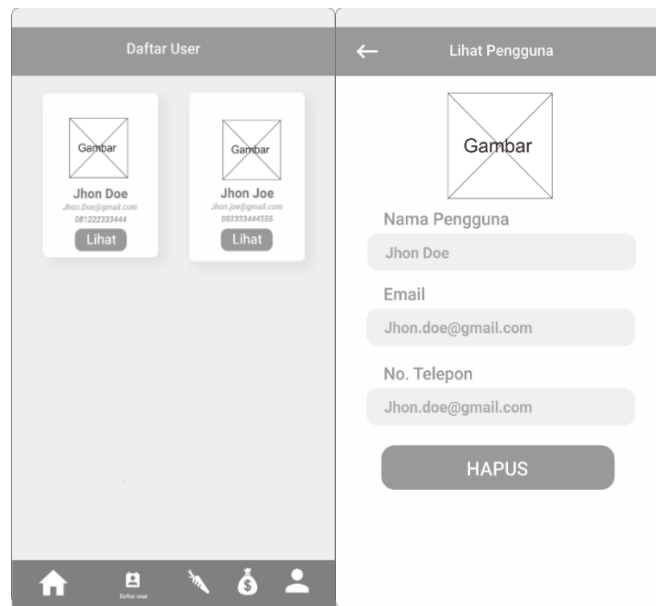
Tampilan halaman gudang sayur adalah menu ketika admin berhasil login. Halaman ini akan menampilkan data sayur yang ada digudang. Di halaman ini terdapat tombol *add* yang berfungsi untuk menambahkan sayur yang ingin dijual dari gudang ,dan diatas juga terdapat tombol search untuk mencari sayur di gudang dengan spesifik. Pada Gambar 5.32 berikut ini menunjukkan halaman Gudang Sayur.



Gambar 5.32 Perancangan Tampilan Halaman Gudang Sayur

5.3.2.8 Perancangan Tampilan *Manage User*

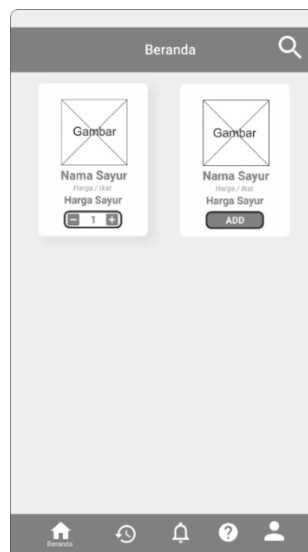
Tampilan halaman *Manage User* adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data *user* yang telah mendaftar. Di halaman ini terdapat tombol lihat yang berfungsi untuk *manage user*, dan diatas juga terdapat tombol search untuk mencari *user* dengan spesifik. Pada Gambar 5.33 berikut ini menunjukkan halaman *Manage User*.



Gambar 5.33 Perancangan Tampilan Halaman *Manage User*

5.3.2.9 Perancangan Tampilan *Manage Sayur*

Tampilan halaman *Manage Sayur* adalah menu ketika admin berhasil login. Halaman ini akan menampilkan data sayur yang telah dijual. Di halaman ini terdapat tombol edit yang berfungsi untuk manage data sayur, dan diatas juga terdapat tombol search untuk mencari sayur yang dijual dengan spesifik. Pada Gambar 5.34 berikut ini menunjukkan halaman *Manage Sayur*.



Gambar 5.34 Perancangan Tampilan Halaman *Manage Sayur*

5.3.2.10 Perancangan Tampilan *Manage* Transaksi

Tampilan halaman *Manage* Transaksi adalah menu ketika admin berhasil login. Halaman ini akan menampilkan data sayur yang dibeli oleh *user*. Di halaman ini admin dapat mengedit transaksi yang telah dilakukan oleh *user*. Pada Gambar 5.35 berikut ini menunjukkan halaman *Manage* Transaksi.

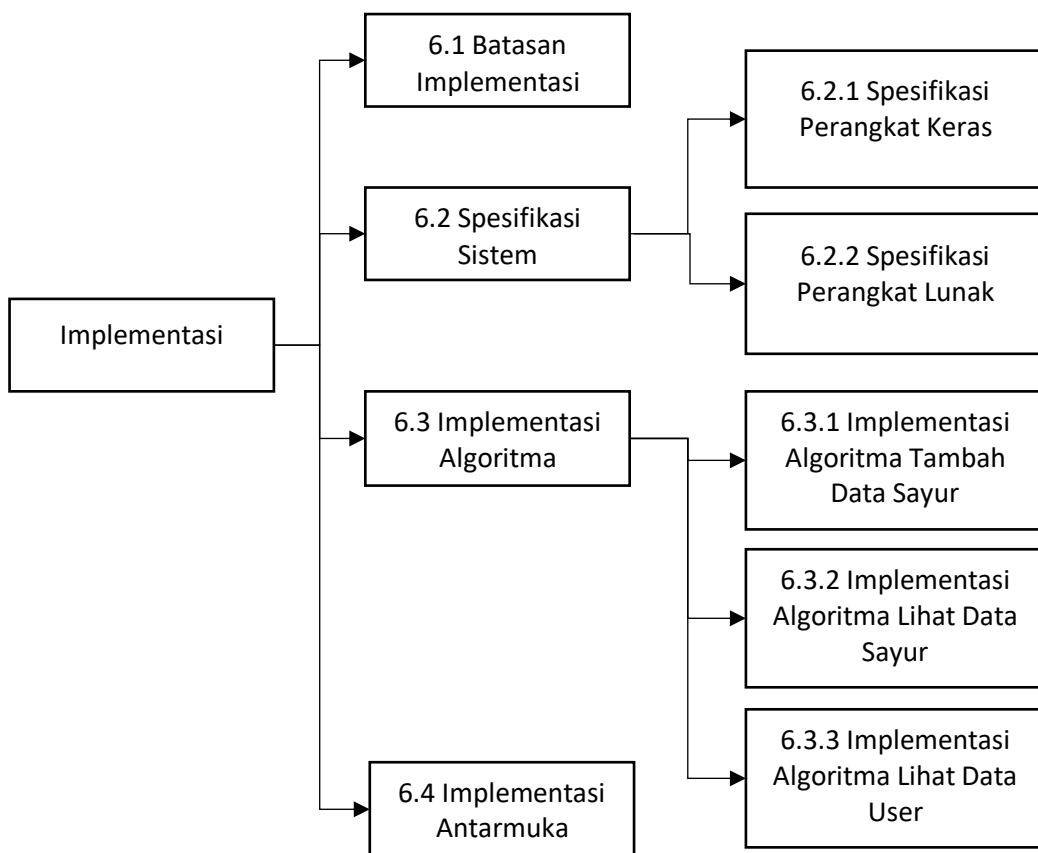
The screenshot displays a mobile application interface for transaction details. The title bar at the top is labeled 'Detail transaksi' with a back arrow. The form is organized into several sections: 'Alamat pengiriman' (Shipping Address) with fields for 'Nama Pemesan' (Buyer Name) and 'Nomor Pemesan' (Order Number); 'Item to Order' with a placeholder for an image and fields for 'Nama Sayur' (Vegetable Name), 'Harga / Kat' (Price / Category), and 'Harga' (Price); 'Alamat pengiriman' (Shipping Address) with a location pin icon and an 'Add Note...' field; 'Waktu Pengiriman' (Shipping Time) with a 'Ubah' (Change) button; 'Tanggal kirim' (Shipping Date) showing '08.00 WIB'; 'Detail Pembayaran' (Payment Details) with a table showing 'Harga (Barang)' (Rp. 12.000,-), 'Biaya pengiriman' (Rp. 3.000,-), and 'Total' (Rp. 15.000,-); and 'Ubah Status' (Change Status) with three buttons: 'Dikonfirmasi', 'Menunggu Pembayaran', and 'Dikirim'.

Detail Pembayaran	
Harga (Barang)	Rp. 12.000,-
Biaya pengiriman	Rp. 3.000,-
Total	Rp. 15.000,-

Gambar 5.35 Perancangan Tampilan Halaman *Manage* Transaksi

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
 4. Data User

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 Menggunakan 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 Menggunakan Webservice menggunakan sebuah komputer yang sesuai dengan spesifikasi perangkat lunak yang dijelaskan pada Tabel 6.2 berikut.

Tabel 6.2 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
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 sayur beserta penjelasannya.

Tabel 6.3 Kode Program Tambah Data Sayur

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

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 Lihat 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	LayoutManager layoutManager =
22	new LinearLayoutManager(getContext(), 2);
23	
24	recyclerView.setLayoutManager(layoutManager);
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 6.5 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();
10	i++){
11	JSONObject jsonObject =
12	response.getJSONObject(i);
13	
14	if(jsonObject.getString("tipe").equals("user")){
15	userdata.add(new
16	Userdata(jsonObject.getInt("id"),Server.URLIMAGE+jsonObject
17	.getString("foto"), jsonObject.getString("name")
18	,jsonObject.getString("email"),
19	jsonObject.getString("nomor_telepon")));
20	RecyclerView recyclerView =
21	(RecyclerView)
22	view.findViewById(R.id.recyclerviewadminhomeuser);
23	LayoutManager
24	gridLayoutManager = new GridLayoutManager(getContext(), 2);
25	
26	recyclerView.setLayoutManager(gridLayoutManager);
27	recyclerView.setAdapter(sayur);
28	}
29	}
30	}catch (JSONException e){
31	e.printStackTrace();
32	}
33	}
34	}, new Response.ErrorListener() {
35	@Override
36	public void onErrorResponse(VolleyError error) {
37	error.printStackTrace();
38	}
39	}){
40	@Override
41	public Map<String, String> getHeaders() throws
42	AuthFailureError {
43	HashMap<String, String> headers = new
44	HashMap<>();
45	headers.put("Accept", "application/json");
46	headers.put("Authorization", token);
47	return headers;
48	}
49	};
50	
51	Singleton.getInstance(this.getContext()).addToRequestQueue(
52	jsonArrayRequest); }

Penjelasan Kode :

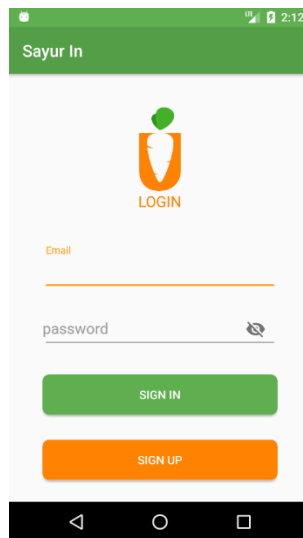
- Baris 1-5: Mengisikan *URL API* dan *method* yang digunakan untuk mengambil data
- Baris 6-7: Merupakan *method* yang digunakan jika proses mengambil data sukses
- Baris 8-29: Melakukan *looping* untuk menyimpan data yang diterima pada *web service* pada array list dan data tersebut akan ditampilkan melalui *recylerview*
- 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 Menggunakan Webservice ini digunakan sebagai perantara antara pengguna dan sistem untuk berinteraksi. Berikut merupakan hasil implementasi antarmuka dari Sistem Jual Beli Sayur Online Berbasis Android Menggunakan Webservice.

a. Implementasi Halaman *Login*

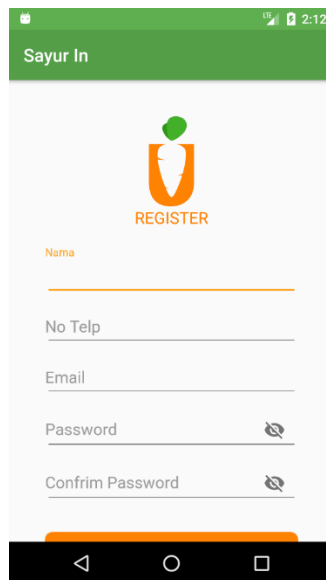
Tampilan halaman login sebagai *user* berisi logo, judul sistem, *field email* dan *password* serta tombol *Sign In* yang digunakan *admin* dan *user* untuk masuk ke dalam sistem setelah menginputkan *email* dan *password* . Pada Gambar 6.2 berikut ini menunjukkan halaman login.



Gambar 6.2 Implementasi Halaman *Login*

b. Implementasi Halaman *Register*

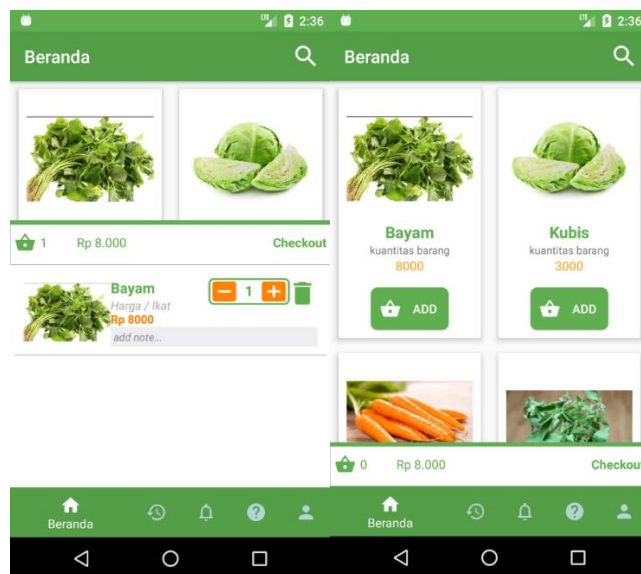
Tampilan halaman register berisi logo, judul sistem, *field* nama, no telp, email, password, konfirmasi password serta tombol *Sign Up* yang digunakan *user* untuk mendaftar ke dalam sistem setelah menginputkan *field* yang tersedia. Pada Gambar 6.3 berikut ini menunjukkan halaman *register*.



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 dijawab 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*.

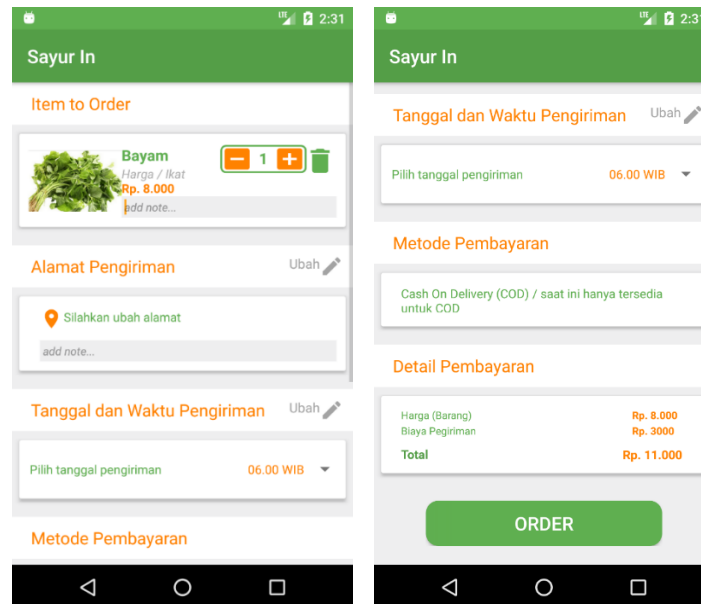


Gambar 6.4 Implementasi Halaman *Home User*

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

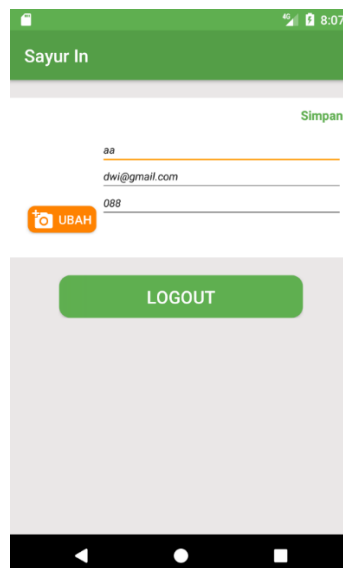
Tampilan halaman Transaksi belanja berisi tampilan yang muncul ketika *user* menekan tombol order. Halaman ini akan menampilkan data sayur, total sayur yang sebelumnya telah di order, jika barang sudah terkirim maka data akan berpindah ke tab *completed*. Pada Gambar 6.6 berikut ini menunjukkan halaman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



Gambar 6.6 Implementasi Halaman Transaksi

f. Implementasi Halaman Profil

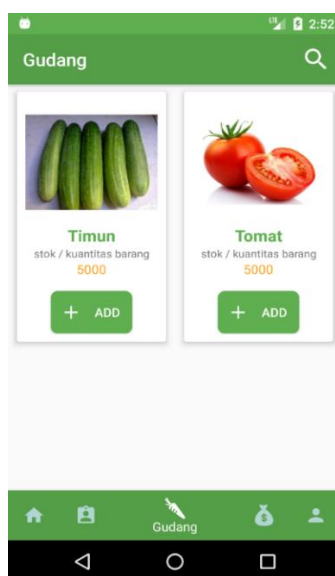
Tampilan halaman Profil muncul ketika *user* menekan tombol yang ada dipojok kanan bawah yaitu tombol *profile*. Halaman ini akan menampilkan data *user* yang yang berhasil login. Di halaman ini *user* dapat merubah nama ,email ,no telepon dan foto. Pada Gambar 6.7 berikut ini menunjukkan halaman Profil.



Gambar 6.7 Implementasi Halaman Profil

g. Implementasi Halaman Gudang Sayur

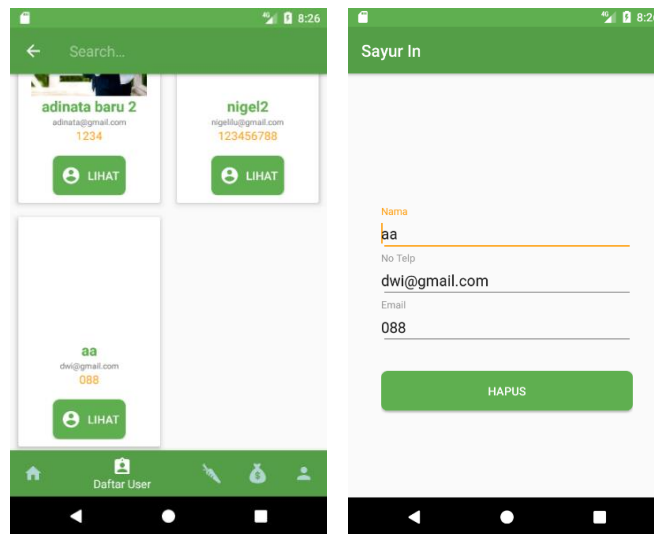
Tampilan halaman gudang sayur adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data sayur yang ada di gudang. Di halaman ini terdapat tombol add yang berfungsi untuk menambahkan sayur yang ingin dijual dari gudang, dan di atas juga terdapat tombol search untuk mencari sayur di 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*

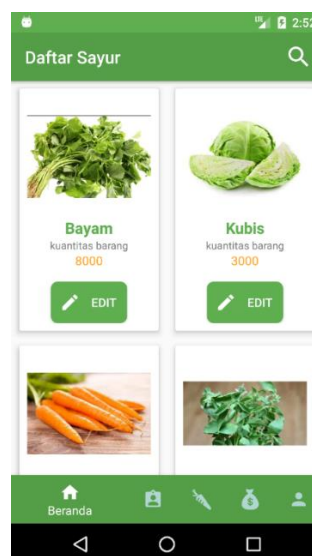
Tampilan halaman *Manage User* adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data *user* yang telah mendaftar. Di halaman ini terdapat tombol lihat yang berfungsi untuk mengelola *user*, dan di atas juga terdapat tombol search untuk mencari *user* dengan spesifik. Pada Gambar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



Gambar 6.9 Implementasi Halaman *Manage User*

i. Implementasi Halaman *Manage Sayur*

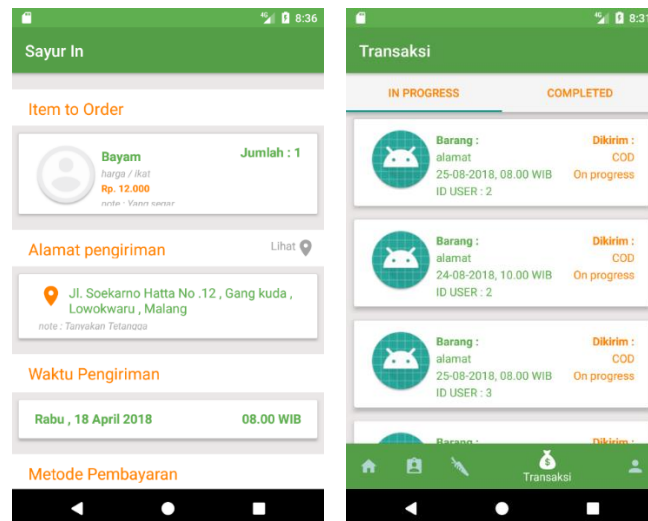
Tampilan halaman *Manage Sayur* adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data sayur yang telah dijual. Di halaman ini terdapat tombol edit yang berfungsi untuk manage data sayur, dan diatas juga terdapat tombol search untuk mencari sayur yang dijual dengan spesifik. Pada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



Gambar 6.10 Implementasi Halaman *Manage Sayur*

j. Implementasi Halaman *Manage* Transaksi

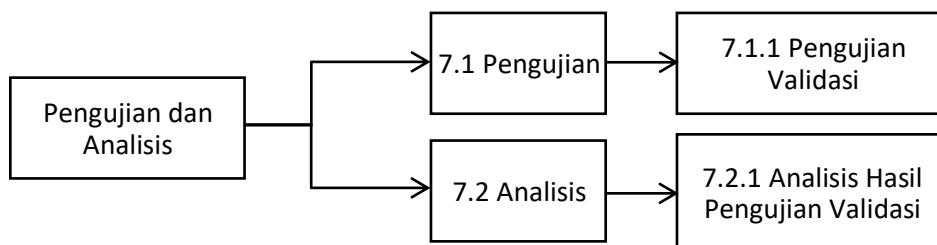
Tampilan halaman *Manage* Transaksi adalah menu ketika admin berhasil *login*. Halaman ini akan menampilkan data sayur yang dibeli oleh *user*. Di halaman ini admin dapat mengedit transaksi yang telah dilakukan oleh *user*. Pada Gambar 6.11 berikut ini menunjukkan halaman *Manage* Transaksi.



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

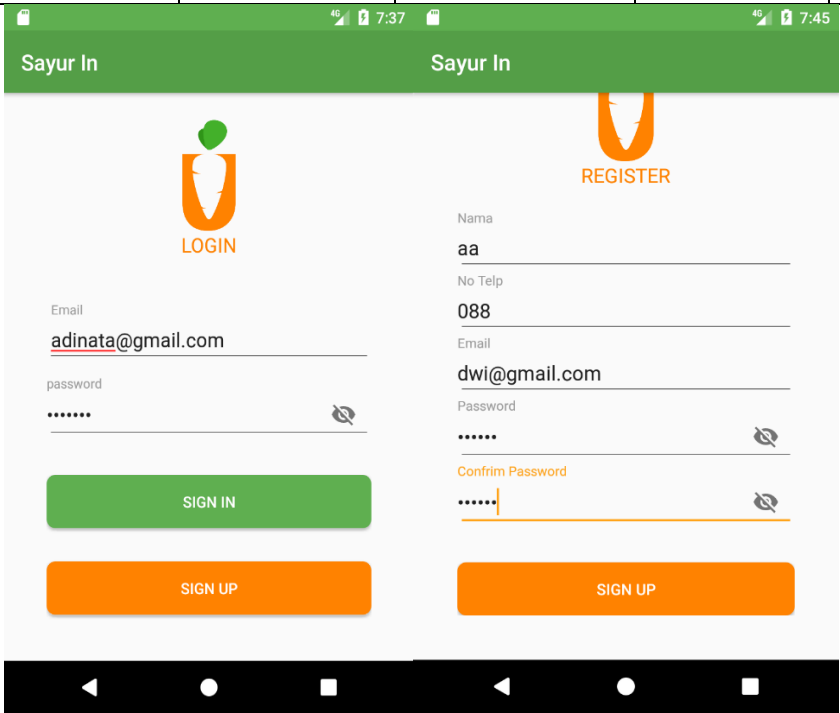
7.1 Pengujian

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

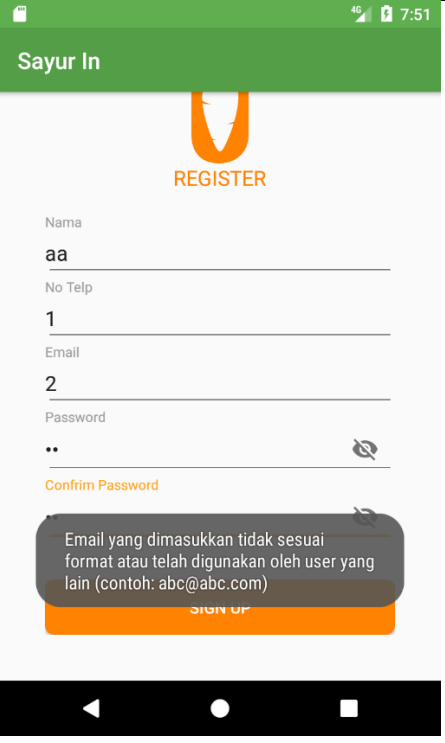
7.1.1 Pengujian Validasi

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*.

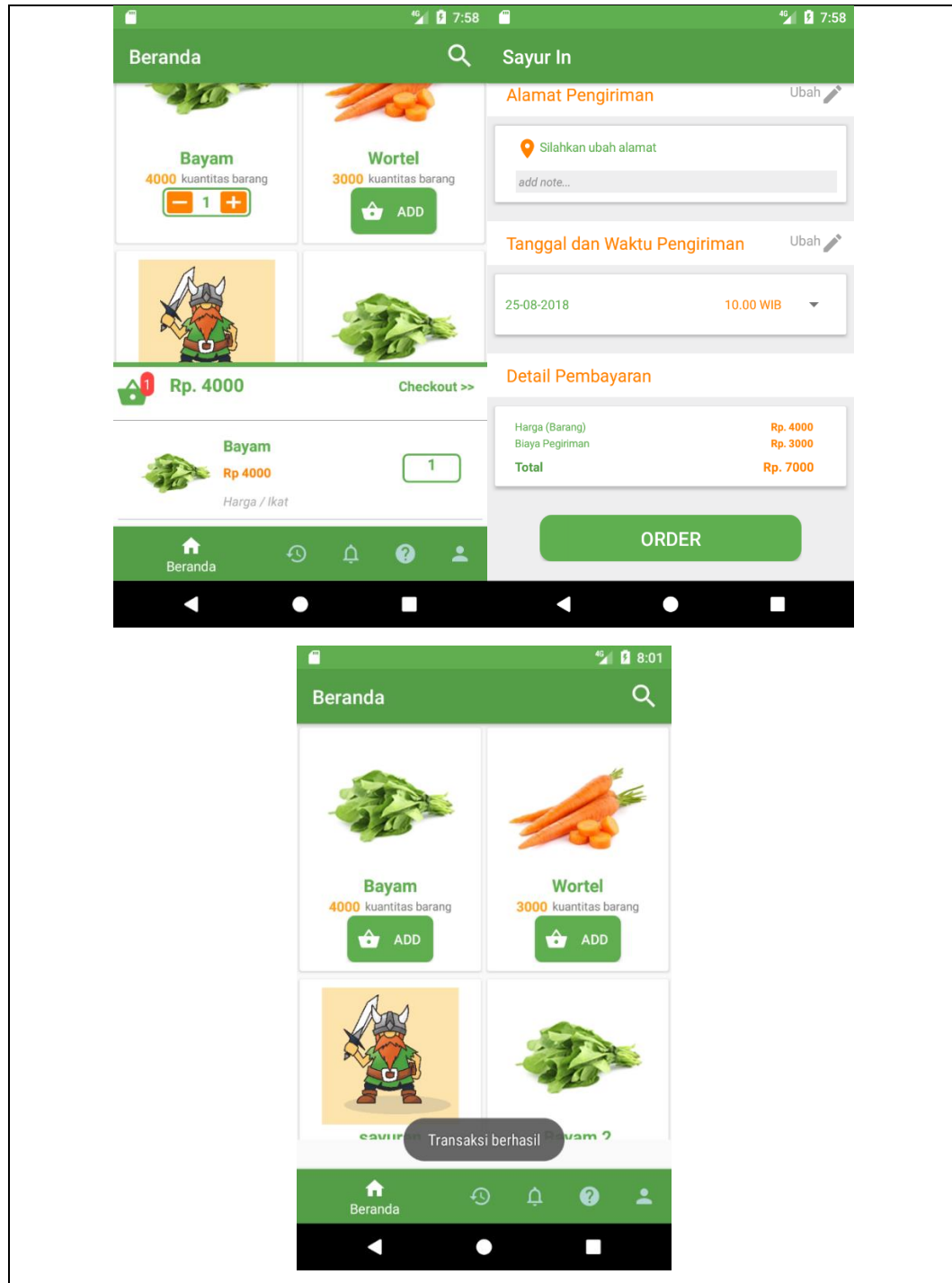
Tabel 7.1 Tabel Pengujian Validasi

Pengujian <i>Sign In</i> , dan <i>Sign Up</i> pada <i>mobile</i>					
1	Pengujian Sign in, dan sign up.	Pengujian dengan mengisi data pada form <i>sign in</i> dan <i>sign up</i> .	Masuk ke halaman home.	Masuk ke halaman <i>home</i> .	Valid
					
2	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 <i>toast</i> “password dan email salah” jika sign in gagal.	Valid

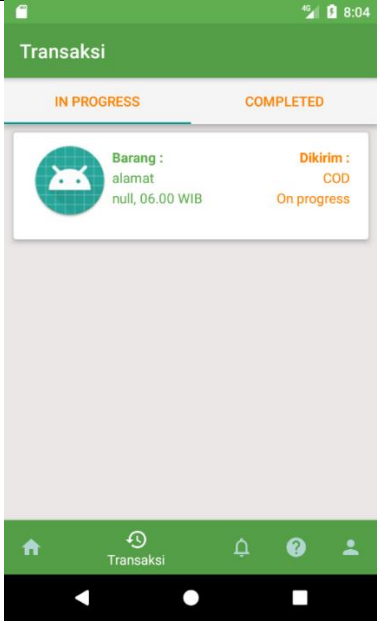
Tabel 7.2 Tabel Pengujian Validasi (lanjutan)

<div>   </div>					
Pengujian Lihat daftar sayur dan melakukan order					
3	Pengujian Lihat daftar sayur dan melakukan order.	Pengujian dengan mengklik menu home dan menambah dengan klik tombol "Add" lalu menekan tombol checkout pada bottomsheets 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 7.3 Tabel Pengujian Validasi (lanjutan)



Tabel 7.4 Tabel Pengujian Validasi (lanjutan)

Pengujian Melihat histori transaksi					
4	Melihat histori transaksi.	Pengujian dengan memilih menu histori .	Menampilkan data transaksi yang telah dilakukan yang sedang “on progress” atau sudah “completed”.	Menampilkan data transaksi yang telah dilakukan yang sedang “on progress” atau sudah “completed”.	Valid
					
Pengujian Edit profil					
5	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

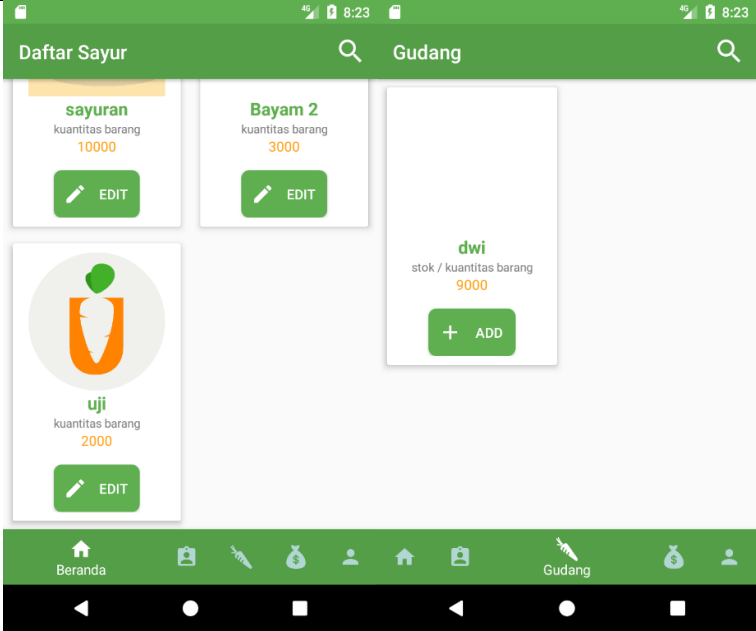
Tabel 7.5 Tabel Pengujian Validasi (lanjutan)

					
Pengujian search sayur					
6	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

Tabel 7.6 Tabel Pengujian Validasi (lanjutan)

					
Pengujian Menambah sayur yang dijual dari gudang					
7	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
					

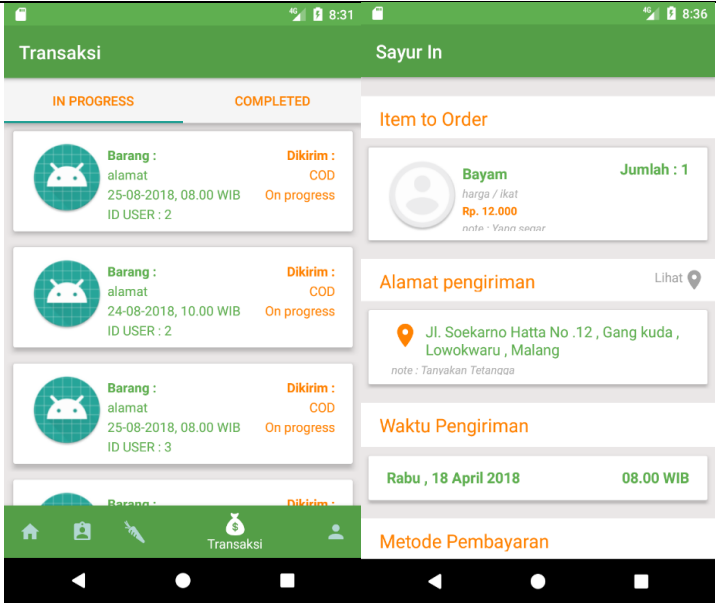
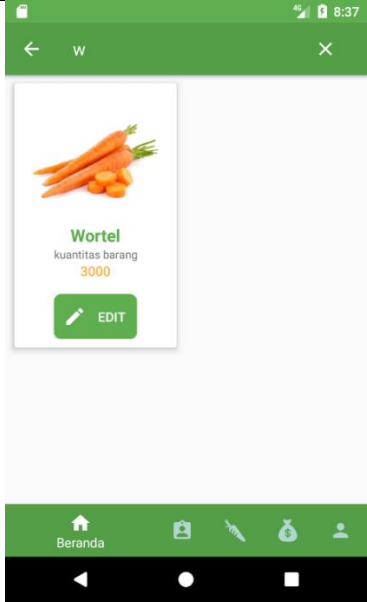
Tabel 7.7 Tabel Pengujian Validasi (lanjutan)

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

Tabel 7.8 Tabel Pengujian Validasi (lanjutan)

 					
Melihat data transaksi dari user					
9	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 7.9 Tabel Pengujian Validasi (lanjutan)

					
<p align="center">Pengujian search sayur (admin)</p>					
10	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 7.10 Tabel Pengujian Validasi (lanjutan)

Pengujian Logout					
11	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*, dan *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 antara *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 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

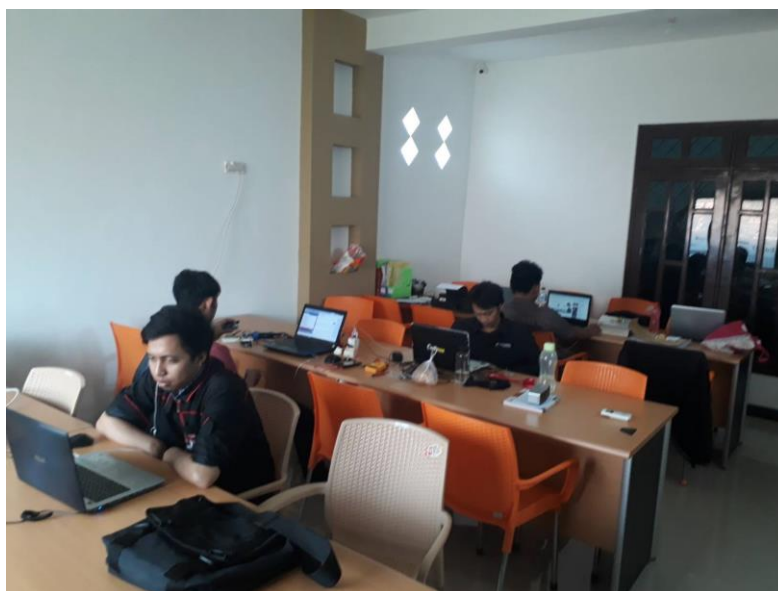
- Android, 2018. *About the Android Open Source Project*. [online] Tersedia di : < <https://source.android.com/> > [Diakses 11 September 2018].
- Android Developer, 2018. *Meet Android Studio*. [online] Tersedia di : < https://developer.android.com/studio/intro/#top_of_page > [Diakses 10 September 2018].
- 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 > [Diakses pada 18 September 2018]
- Dennis, A., Wixom, B.H. & Roth, R.M., 2012. *System Analysis & Design*. 5th ed. USA : Jhon Wiley & Sons, Inc.
- 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*. 6th ed [e-book]. Tersedia di : [ebookfrenzy<https://www.ebookfrenzy.com/pdf_previews/AndroidStudio6EssentialsPreview.pdf>](https://www.ebookfrenzy.com/pdf_previews/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

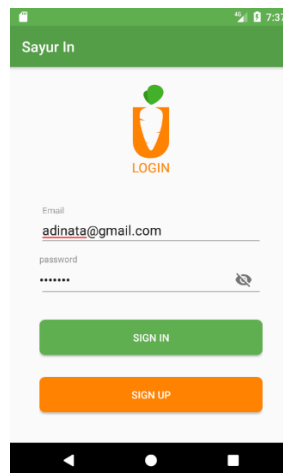
1.1 Admin

1.1.1 Bagian I – Login

Login pada android :

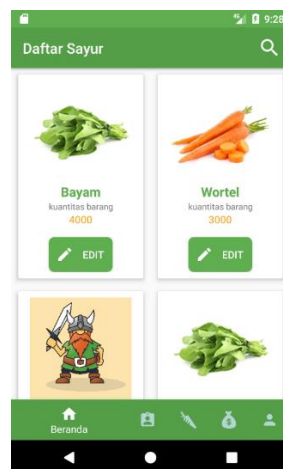
Pada bagian ini pengguna melakukan proses *login* pada aplikasi *android*.

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*.

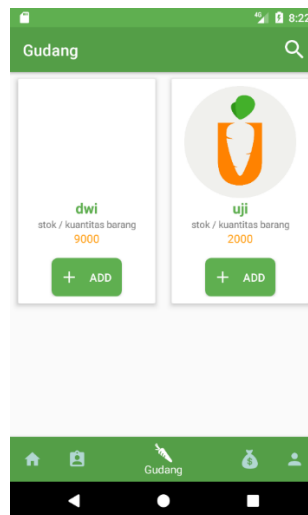


Tampilan home admin android

1.1.2 Bagian II – Menambahkan 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.

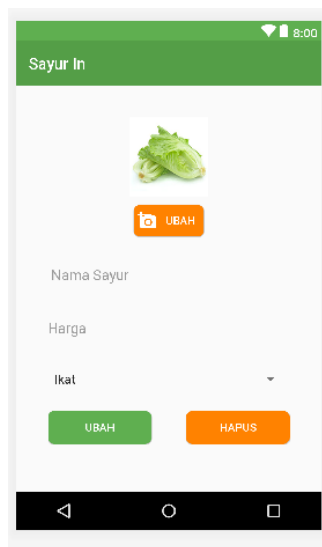


Tampilan gudang sayur

1.1.3 Bagian III – Edit sayur.

Edit sayur pada 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 android :

Untuk menghapus sayur pada android admin hanya perlu menekan tombol




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




dan sayur akan dihapus.

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

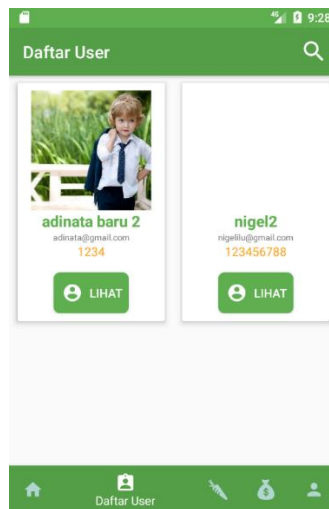
Melihat dan menghapus daftar *user* pada android :

Pada android untuk melihat daftar *user* admin memilih menu , lalu akan ditampilkan tampilan daftar *user* yang terdaftar. Untuk menghapus *user*, admin perlu memilih *user* yang ingin dihapus lalu tekan tombol

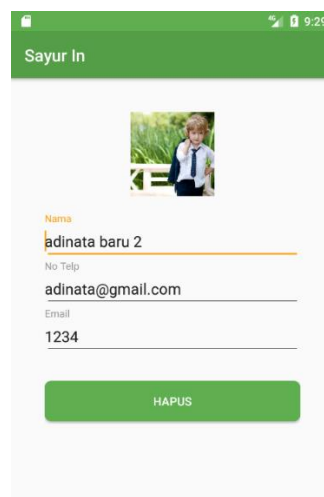


. Setelah itu akan ditampilkan detail dari *user* dan admin menekan tombol  untuk menghapus *user* yang dipilih.

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 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

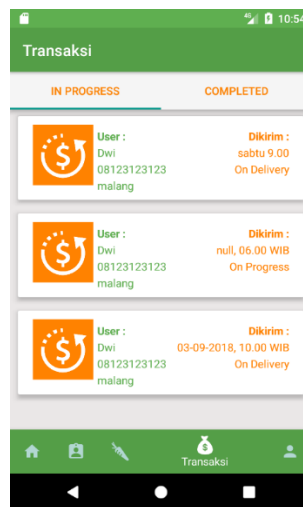
menekan salah satu dari status yang diinginkan. Jika memilih

ON DELIVERY

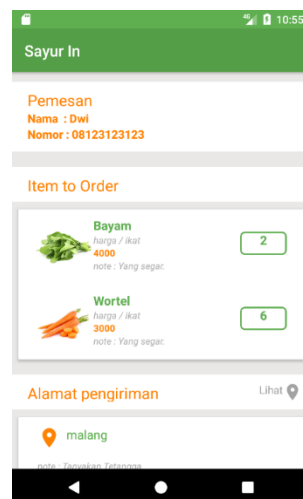
maka artinya transaksi sedang proses pengiriman, jika

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

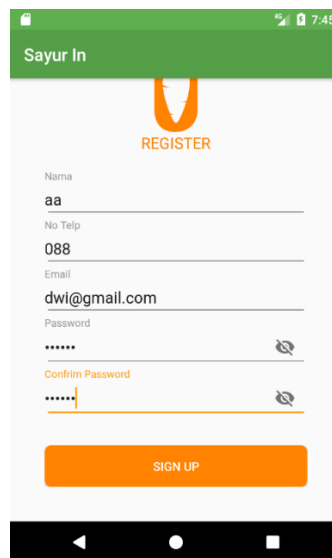


Tampilan detail transaksi

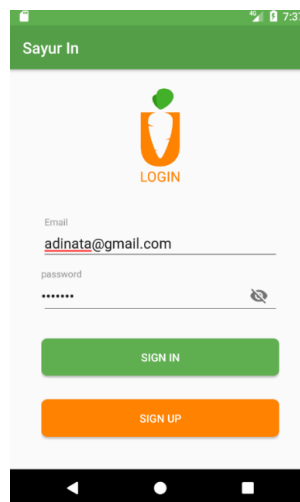
2.2. User

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:

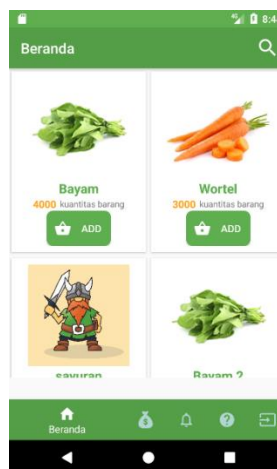


Tampilan halaman *register*



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*

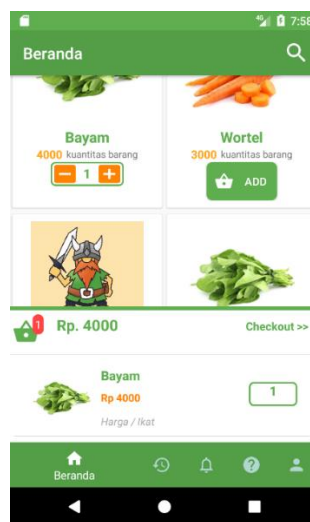
1.2.2 Bagian II – Memesan sayur

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

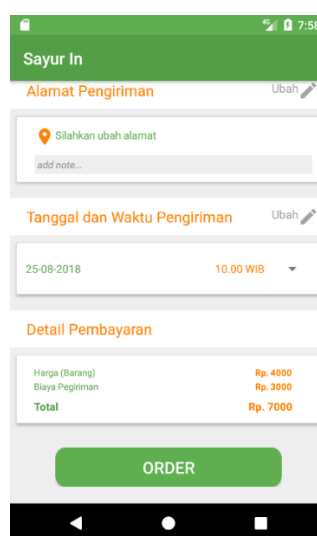


pada daftar sayur yang ingin dibeli, lalu akan muncul keranjang pada bagian bawah. *User* dapat mengatur jumlah sayur yang ingin dibeli dengan

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  untuk menyelesaikan proses pemesanan sayur. Berikut tampilan keranjang sayur dan halaman checkout :




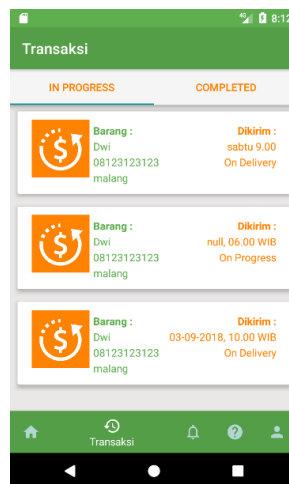
Tampilan keranjang sayur



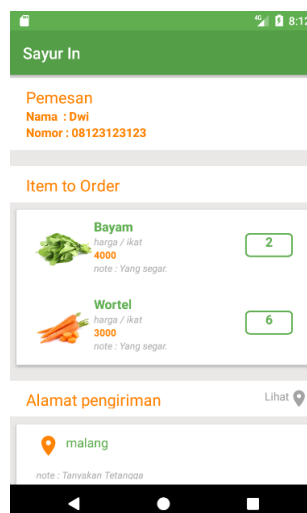
Tampilan halaman *checkout*

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 :





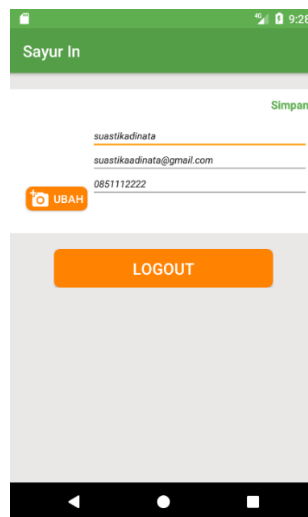
Tampilan menu transaksi



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  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  pada menu *profile*. Setelah proses *logout* selesai tampilan akan berubah ke halaman *home*.

LOG BOOK