**LAPORAN PRAKTIK KERJA LAPANG (PKL)**

**CV SARANA UTAMA SOLUSINDO**

**SISTEM JUAL BELI SAYUR *ONLINE* BERBASIS *ANDROID***

**MENGGUNAKAN *WEB SERVICE***



**Disusun oleh:**

Kadek Dwi Aryasa 155150201111124

Gusti Agung Suastika adinata 155150201111273

Nigel Erico Pangestu

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2018

# PENGESAHAN

**LAPORAN PRAKTIK KERJA LAPANGAN (PKL)**

Diajukan Untuk Memenuhi Sebagai Persyaratan Kurikulum

Program Studi Teknik Informatika

Jurusan Teknik Informatika

Universitas Brawijaya

SISTEM JUAL BELI SAYUR *ONLINE* BERBASIS *ANDROID*

MENGGUNAKAN *WEB SERVICE*

Dilaksanakan di :

CV SARANA UTAMA SOLUSINDO

Lantai 2 Jalan Anggrek garuda No. 49, kota malang

Tanggal :

1 Juli 2018 – 30 Agustus 2018

Disusun Oleh :

|  |  |
| --- | --- |
| Kadek Dwi Aryasa | 155150201111124 |
| Gusti Agung Suastika Adinata | 155150201111273 |

Telah diperiksa dan disetujui oleh :

Mengetahui, Menyetujui,

Kepala Jurusan Teknik Informatika Dosen Pembimbing PKL

Tri Astoto Kurniawan, S.T, M.T, Ph.D. Komang Candra Brata, S.Kom., M.T., M.Sc.

NIP: 19710518 200312 1 001 NIP: 201607 890711 1 000

# LEMBAR PENGESAHAN

**LAPORAN PRAKTIK KERJA LAPANGAN (PKL)**

Diajukan Untuk Memenuhi Sebagai Persyaratan Kurikulum

Program Studi Teknik Informatika

Jurusan Teknik Informatika

Universitas Brawijaya

SISTEM JUAL BELI SAYUR *ONLINE* BERBASIS *ANDROID*

MENGGUNAKAN *WEB SERVICE*

Dilaksanakan di :

CV SARANA UTAMA SOLUSINDO

Lantai 2 Jalan Anggrek garuda No. 49, kota malang

Tanggal :

1 Juli 2018 – 30 Agustus 2018

Disusun Oleh :

|  |  |
| --- | --- |
| Kadek Dwi Aryasa | 155150201111124 |
| Gusti Agung Suastika Adinata  Nigel Erico Pangestu | 155150201111273 |

Telah diperiksa dan disetujui oleh :

Pihak CV. SARANA UTAMA SOLUSINDO

Menyetujui, Menyetujui,

General Manager Pembimbing Lapangan

Sony Winarto Selvina Yustriasanti

# KATA PENGANTAR

*Assalamu’alaikum Wr. Wb.*

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

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

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



1. Teman-teman satu angkatan Program Studi Informatika 2015 tercinta yang selalu memberikan informasi, semangat, dorongan dan bantuan pikiran.
2. Segenap civitas akademika Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya.

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

*Wassalamu’alaikum Wr. Wb.*

Malang, 24 Oktober 2018

Penulis

# ABSTRAK

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

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

# DAFTAR ISI

[PENGESAHAN ii](#_Toc528129547)

[LEMBAR PENGESAHAN iii](#_Toc528129548)

[KATA PENGANTAR iv](#_Toc528129549)

[ABSTRAK vi](#_Toc528129550)

[DAFTAR ISI vii](#_Toc528129551)

[DAFTAR TABEL x](#_Toc528129552)

[DAFTAR GAMBAR xii](#_Toc528129553)

[DAFTAR LAMPIRAN xvi](#_Toc528129554)

[BAB 1 PENDAHULUAN 1](#_Toc528129555)

[1.1 Latar belakang 1](#_Toc528129556)

[1.2 Rumusan masalah 1](#_Toc528129557)

[1.3 Tujuan 1](#_Toc528129558)

[1.4 Manfaat 1](#_Toc528129559)

[1.5 Batasan masalah 1](#_Toc528129560)

[1.6 Sistematika pembahasan 2](#_Toc528129561)

[1.7 Pelaksanaan Praktek Kerja Lapang 3](#_Toc528129562)

[BAB 2 PROFIL PERUSAHAAN 4](#_Toc528129563)

[2.1 Sejarah dan Perkembangan CV Sarana Utama Solusindo (SASINDO) 4](#_Toc528129564)

[2.2 Visi, Misi dan Komitmen Perusahaan. 4](#_Toc528129565)

[2.2.1 Visi 4](#_Toc528129566)

[2.2.2 Misi 4](#_Toc528129567)

[2.2.3 Komitmen 5](#_Toc528129568)

[2.3 Struktur Organisasi CV SARANA UTAMA SOLUSINDO 5](#_Toc528129569)

[2.4 Ruang Lingkup CV SARANA UTAMA SOLUSINDO 5](#_Toc528129570)

[2.4.1 Solusi Bisnis TI (IT Business Solution) dan Konsultan IT (IT Consultant) 6](#_Toc528129571)

[2.4.2 Pendidikan dan Pelatihan (IT Training) 7](#_Toc528129572)

[BAB 3 LANDASAN KEPUSTAKAAN 8](#_Toc528129573)

[3.1 Sistem Penjualan. 8](#_Toc528129574)

[3.1.2 Sistem Jual Beli *Online*. 8](#_Toc528129575)

[3.2 *Android* 9](#_Toc528129576)

[3.2.1 *Android Studio* 11](#_Toc528129577)

[3.3 *Web Service* 12](#_Toc528129578)

[3.4 *Laravel* 13](#_Toc528129579)

[3.4.1 *Model-View-Controller* (MVC) 13](#_Toc528129580)

[3.5 *Unified Modeling Language (UML)* 14](#_Toc528129581)

[3.5.1 *Structured Diagram.* 15](#_Toc528129582)

[3.5.2 *Behavior Diagram*. 17](#_Toc528129583)

[*3.6* *Entity Relationship Diagram* 22](#_Toc528129584)

[BAB 4 METODOLOGI 24](#_Toc528129585)

[4.1 Penentuan Objek Penelitian 24](#_Toc528129586)

[4.2 Studi Literatur 25](#_Toc528129587)

[4.3 Analisa Kasus 25](#_Toc528129588)

[4.4 Perancangan Konsep dan Model 25](#_Toc528129589)

[4.5 Implementasi 26](#_Toc528129590)

[4.6 Pengujian dan Analisis 26](#_Toc528129591)

[4.7 Kesimpulan 26](#_Toc528129592)

[BAB 5 ANALISIS DAN PERANCANGAN 27](#_Toc528129593)

[5.1 Analisa Kebutuhan Sistem 27](#_Toc528129594)

[5.1.1 Deskripsi Umum Sistem 28](#_Toc528129595)

[5.1.2 Deskripsi Umum Perangkat Lunak 28](#_Toc528129596)

[5.1.3 Fungsi Perangkat Lunak 29](#_Toc528129597)

[5.1.4 Karakteristik Pengguna 29](#_Toc528129598)

[5.2 Analisa Kebutuhan Perangkat Lunak 30](#_Toc528129599)

[5.2.1 Kebutuhan Fungsional dan Non-Fungsional 31](#_Toc528129600)

[5.2.2 Pemodelan Kebutuhan 33](#_Toc528129601)

[5.3 Perancangan Perangkat Lunak 76](#_Toc528129602)

[5.3.1 Perancangan Algoritma 76](#_Toc528129603)

[5.3.2 Perancangan Antarmuka 80](#_Toc528129604)

[5.3.3.1 Perancangan Tampilan Halaman *Login* 81](#_Toc528129605)

[5.3.3.2 Perancangan Tampilan Halaman *Register* 82](#_Toc528129606)

[5.3.3.3 Perancangan Tampilan Halaman *Home User* 83](#_Toc528129607)

[5.3.3.4 Perancangan Tampilan Halaman Keranjang Belanja 84](#_Toc528129608)

[5.3.3.5 Perancangan Tampilan Halaman Transaksi 85](#_Toc528129609)

[5.3.3.6 Perancangan Tampilan Halaman Profil 86](#_Toc528129610)

[5.3.3.7 Perancangan Tampilan Gudang Sayur 87](#_Toc528129611)

[5.3.3.8 Perancangan Tampilan *Manage User* 88](#_Toc528129612)

[5.3.3.9 Perancangan Tampilan *Manage* Sayur 89](#_Toc528129613)

[5.3.3.10 Perancangan Tampilan *Manage* Transaksi 90](#_Toc528129614)

[5.3.3.11 Perancangan Tampilan *Login* Web 91](#_Toc528129615)

[5.3.3.12 Perancangan Tampilan *Manage* SayurWeb 92](#_Toc528129616)

[5.3.3.13 Perancangan Tampilan *Manage* TransaksiWeb 93](#_Toc528129617)

[5.3.3.14 Perancangan Tampilan *Manage User* Web 94](#_Toc528129618)

[5.3.3.14 Perancangan Tampilan *Manage User* Web 95](#_Toc528129619)

[5.3.3.15 Perancangan Tampilan Detail SayurWeb 96](#_Toc528129620)

[5.3.3.16 Perancangan Tampilan Detail TransaksiWeb 97](#_Toc528129621)

[BAB 6 IMPLEMENTASI 98](#_Toc528129622)

[6.1 Batasan Implementasi 99](#_Toc528129623)

[6.2 Spesifikasi Sistem 99](#_Toc528129624)

[6.2.1 Spesifikasi Perangkat Keras 99](#_Toc528129625)

[6.2.2 Spesifikasi Perangkat Lunak 100](#_Toc528129626)

[6.3 Implementasi Algoritma 100](#_Toc528129627)

[6.3.1 Implementasi Algoritma *Login* 100](#_Toc528129628)

[6.3.2 Implementasi Algoritma Tambah Data Sayur 101](#_Toc528129629)

[6.3.3 Implementasi Algoritma *Edit* Data Sayur 102](#_Toc528129630)

[6.4 Antarmuka 103](#_Toc528129631)

[BAB 7 PENGUJIAN DAN ANALISIS 112](#_Toc528129632)

[7.1 Pengujian 112](#_Toc528129633)

[7.1.1 Pengujian Validasi 112](#_Toc528129634)

[7.2 Analisis 130](#_Toc528129635)

[7.2.1 Analisis Hasil Pengujian Validasi 130](#_Toc528129636)

[BAB 8 PENUTUP 132](#_Toc528129637)

[8.1 Kesimpulan 132](#_Toc528129638)

[8.2 Saran 133](#_Toc528129639)

[DAFTAR PUSTAKA 134](#_Toc528129640)

[LAMPIRAN 136](#_Toc528129641)

# DAFTAR TABEL

[Tabel 1.1 Kegiatan 3](#_Toc528134013)

[Tabel 3.1 Notasi pada Structured diagram. 17](#_Toc528134014)

[Tabel 3.2 Notasi pada *Use Case Diagram* 18](#_Toc528134015)

[Tabel 3.3 Notasi pada *Sequence Diagram* 20](#_Toc528134016)

[Tabel 3.4 Notasi pada *Entity Relationship Diagram* 22](#_Toc528134017)

[Tabel 5.1 Karakter Pengguna 30](#_Toc528134018)

[Tabel 5.2 Kebutuhan Fungsional 31](#_Toc528134019)

[Tabel 5.4 Skenario *Use Case* *Sign in* 37](#_Toc528134020)

[Tabel 5.5 Skenario *Use Case* *Sign up* 37](#_Toc528134021)

[Tabel 5.6 Skenario *Use Case* Lihat daftar sayur pada *home* 38](#_Toc528134022)

[Tabel 5.7 Skenario *Use Case* Tambah keranjang belanja 38](#_Toc528134023)

[Tabel 5.8 Skenario *Use Case* hapus keranjang belanja 38](#_Toc528134024)

[Tabel 5.9 Skenario *Use Case* *Search* sayur 39](#_Toc528134025)

[Tabel 5.10 Skenario *Use Case* Lihat Keranjang belanja 39](#_Toc528134026)

[Tabel 5.11 Skenario *Use Case* *Edit* jumlah sayur pada keranjang. 39](#_Toc528134027)

[Tabel 5.12 Skenario *Use Case* Hapus daftar pembelian 40](#_Toc528134028)

[Tabel 5.13 Skenario *Use Case* Melengkapi Data pengiriman 40](#_Toc528134029)

[Tabel 5.14 Skenario *Use Case* Lihat transaksi. 40](#_Toc528134030)

[Tabel 5.15 Skenario *Use Case* Melihat detail transaksi. 41](#_Toc528134031)

[Tabel 5.16 Skenario *Use Case* lihat bantuan 41](#_Toc528134032)

[Tabel 5.17 Skenario *Use Case* lihat profil 41](#_Toc528134033)

[Tabel 5.18 Skenario *Use Case* *Edit* Profil 42](#_Toc528134034)

[Tabel 5.19 Skenario *Use Case* *Logout*. 42](#_Toc528134035)

[Tabel 5.20 Skenario *Use Case* lihat daftar sayur pada gudang. 43](#_Toc528134036)

[Tabel 5.21 Skenario *Use Case* Menambah sayur. 43](#_Toc528134037)

[Tabel 5.22 Skenario *Use Case* Hapus Data sayur 43](#_Toc528134038)

[Tabel 5.23 Skenario *Use Case* *Edit* data sayur 44](#_Toc528134039)

[Tabel 5.24 Skenario *Use Case* Melihat daftar *user* 44](#_Toc528134040)

[Tabel 5.25 Skenario *Use Case* Melihat data *user* 44](#_Toc528134041)

[Tabel 5.26 Skenario *Use Case* Hapus data *user* 45](#_Toc528134042)

[Tabel 5.27 Skenario *Use Case* Mengubah status transaksi. 45](#_Toc528134043)

[Tabel 5.28 Skenario *Use Case* Lihat transaksi *On progress*. 46](#_Toc528134044)

[Tabel 5.29 Skenario *Use Case* Lihat transaksi *Completed*. 46](#_Toc528134045)

[Tabel 5.30 Skenario *Use Case* *Login* pada web.. 46](#_Toc528134046)

[Tabel 5.31 Skenario *Use Case* Manajemen sayur. 47](#_Toc528134047)

[Tabel 5.32 Skenario *Use Case* tambah sayur. 47](#_Toc528134048)

[Tabel 5.33 Skenario *Use Case* lihat data sayur. 47](#_Toc528134049)

[Tabel 5.34 Skenario *Use Case* edit data sayur. 48](#_Toc528134050)

[Tabel 5.35 Skenario *Use Case* hapus data sayur. 48](#_Toc528134051)

[Tabel 5.36 Skenario *Use Case search* sayur. 48](#_Toc528134052)

[Tabel 5.37 Skenario *Use Case* manajemen transaksi. 49](#_Toc528134053)

[Tabel 5.38 Skenario *Use Case* lihat detail transaksi. 49](#_Toc528134054)

[Tabel 5.39 Skenario *Use Case* manajemen pengguna. 49](#_Toc528134055)

[Tabel 5.40 Skenario *Use Case* hapus pengguna. 50](#_Toc528134056)

[Tabel 5.41 Skenario *Use Case* tambah pengguna. 50](#_Toc528134057)

[Tabel 5.42 Skenario *Use Case logout*. 51](#_Toc528134058)

[Tabel 6.1 Spesifikasi Perangkat Keras 99](#_Toc528134059)

[Tabel 6.2 Spesifikasi Perangkat Lunak 100](#_Toc528134060)

[Tabel 7.1 Tabel Pengujian Validasi 113](#_Toc528134061)

# DAFTAR GAMBAR

[Gambar 2.1 Struktur Organisasi 5](#_Toc528134450)

[Gambar 3.1. Illustrasi *Android* *Stack* 10](#_Toc528134451)

[Gambar 3.2. *Class Diagram* 16](#_Toc528134452)

[Gambar 3.3. *Package diagram*. 16](#_Toc528134453)

[Gambar 3.4. *Use Case Diagram*. 18](#_Toc528134454)

[Gambar 3.6. *Sequence Diagram*. 20](#_Toc528134455)

[Gambar 3.7. *Use Case Scenario.* 21](#_Toc528134456)

[Gambar 3.7. *Use Case Scenario*. 22](#_Toc528134457)

[Gambar 4.1 Diagram Alir Metodologi 24](#_Toc528134458)

[Gambar 5.1 Diagram Alir Analisis dan Perancangan 27](#_Toc528134459)

[Gambar 5.2 Deskripsi Sistem 28](#_Toc528134460)

[Gambar 5.3 *Entity Relationship Diagram* Untuk Aplikasi Sayur *Online* 34](#_Toc528134461)

[Gambar 5.4 *Use Case Diagram* Untuk Platform *Android* 35](#_Toc528134462)

[Gambar 5.3 *Use Case Diagram* Untuk Platform Web 36](#_Toc528134463)

[Gambar 5.4 *Class Diagram* Untuk Platform *Android* Pada Bagian *User* 52](#_Toc528134464)

[Gambar 5.5 *Class Diagram* Untuk Platform *Android* Pada Bagian Admin 53](#_Toc528134465)

[Gambar 5.6 *Class Diagram* Untuk Platform *Android* Pada *Package* Admin Bagian 1 54](#_Toc528134466)

[Gambar 5.7 *Class Diagram* Untuk Platform *Android* Pada *Package* Admin Bagian 2 54](#_Toc528134467)

[Gambar 5.8 Detail *Class Diagram* Untuk Platform *Android* Pada *Package* *Adapter* 55](#_Toc528134468)

[Gambar 5.9 Detail *Class Diagram* Untuk Platform *Android* Pada *Package* *Fragment* 56](#_Toc528134469)

[Gambar 5.10 Detail *Class Diagram* Untuk Platform *Android* Pada *Package Server* 56](#_Toc528134470)

[Gambar 5.11 Detail *Class Diagram* Untuk Platform *Android* Pada *Package Singleton* 56](#_Toc528134471)

[Gambar 5.12 Detail *Class Diagram* Untuk Platform *Android* Pada *Package drawersayur* 57](#_Toc528134472)

[Gambar 5.13 *Class Diagram* Untuk Platform Web Untuk Proses Pada Website 58](#_Toc528134473)

[Gambar 5.14 *Class Diagram* Untuk Platform Web Untuk Proses *Web Service* 58](#_Toc528134474)

[Gambar 5.15 Detail *Class Diagram* Untuk Platform Web Pada *Folder Controller* 59](#_Toc528134475)

[Gambar 5.16 Detail *Class Diagram* Untuk Platform Web Pada *Folder API* 59](#_Toc528134476)

[Gambar 5.17 Detail *Class Diagram* Untuk Platform Web Pada *Folder App* 60](#_Toc528134477)

[Gambar 5.18 *Activity Diagram* “hapus data *user*” 60](#_Toc528134478)

[Gambar 5.19 *Activity Diagram* “*search* data *user*” 61](#_Toc528134479)

[Gambar 5.20 *Activity Diagram* “perbarui data sayur” 61](#_Toc528134480)

[Gambar 5.21 *Activity Diagram* “hapus data sayur” 62](#_Toc528134481)

[Gambar 5.22 Activity Diagram “*search* sayur” 62](#_Toc528134482)

[Gambar 5.23 Activity Diagram “menambah sayur” 63](#_Toc528134483)

[Gambar 5.24 Activity Diagram “pesan sayur” 63](#_Toc528134484)

[Gambar 5.25 *Activity Diagram* “melihat daftar transaksi” 64](#_Toc528134485)

[Gambar 5.26 *Activity Diagram* “melihat daftar transaksi” 64](#_Toc528134486)

[Gambar 5.27 *Activity Diagram* “*edit* profil” 65](#_Toc528134487)

[Gambar 5.28 *Activity Diagram* “tambah sayur pada web” 65](#_Toc528134488)

[Gambar 5.29 *Activity Diagram* “*search* sayur pada web” 66](#_Toc528134489)

[Gambar 5.30 *Activity Diagram* “ubah sayur pada *web*” 66](#_Toc528134490)

[Gambar 5.31 *Activity Diagram* “tambah *user* pada web” 67](#_Toc528134491)

[Gambar 5.32 *Activity Diagram* “hapus *user* pada web” 67](#_Toc528134492)

[Gambar 5.33 *Activity Diagram* “lihat data transaksi pada web” 68](#_Toc528134493)

[Gambar 5.34 Activity Diagram “Ubah data transaksi pada web” 68](#_Toc528134494)

[Gambar 5.35 *Sequence Diagram* “menambah sayur” 69](#_Toc528134495)

[Gambar 5.36 *Sequence Diagram* “perbarui data sayur” 69](#_Toc528134496)

[Gambar 5.37 *Sequence Diagram* “hapus sayur” 70](#_Toc528134497)

[Gambar 5.38 *Sequence Diagram* “hapus *user*” 70](#_Toc528134498)

[Gambar 5.39 *Sequence Diagram* “*search* sayur” 71](#_Toc528134499)

[Gambar 5.40 *Sequence Diagram* “*search user*” 71](#_Toc528134500)

[Gambar 5.41 *Sequence Diagram* “keranjang belanja” 72](#_Toc528134501)

[Gambar 5.42 *Sequence Diagram* “pesan sayur” 72](#_Toc528134502)

[Gambar 5.43 *Sequence Diagram* “Tambah sayur web” 73](#_Toc528134503)

[Gambar 5.44 *Sequence Diagram* “*Search* sayur web” 73](#_Toc528134504)

[Gambar 5.45 *Sequence Diagram* “ubah sayur web” 74](#_Toc528134505)

[Gambar 5.46 *Sequence Diagram* “hapus *user* web” 74](#_Toc528134506)

[Gambar 5.47 *Sequence Diagram* “tambah *user* web” 75](#_Toc528134507)

[Gambar 5.48 *Sequence Diagram* “lihat transaksi web” 75](#_Toc528134508)

[Gambar 5.49 Diagram Alir Proses Login Sebagai Admin 76](#_Toc528134509)

[Gambar 5.50 Diagram Alir Proses Tambah Data Sayur 77](#_Toc528134510)

[Gambar 5.51 Diagram Alir Proses Lihat Data Sayur 78](#_Toc528134511)

[Gambar 5.52 Diagram Alir Proses Perbarui Data Sayur 79](#_Toc528134512)

[Gambar 5.53 Diagram Alir Proses Hapus Data Sayur 80](#_Toc528134513)

[Gambar 5.54 Perancangan Tampilan Halaman *Login* 81](#_Toc528134514)

[Gambar 5.55 Perancangan Tampilan Halaman *Register* 82](#_Toc528134515)

[Gambar 5.56 Perancangan Tampilan Halaman *Home User* 83](#_Toc528134516)

[Gambar 5.57 Perancangan Tampilan Halaman Keranjang Belanja 84](#_Toc528134517)

[Gambar 5.58 Perancangan Tampilan Halaman Transaksi 85](#_Toc528134518)

[Gambar 5.59 Perancangan Tampilan Halaman Profil 86](#_Toc528134519)

[Gambar 5.60 Perancangan Tampilan Halaman Gudang Sayur 87](#_Toc528134520)

[Gambar 5.61 Perancangan Tampilan Halaman *Manage* *User* 88](#_Toc528134521)

[Gambar 5.62 Perancangan Tampilan Halaman Manage Sayur 89](#_Toc528134522)

[Gambar 5.63 Perancangan Tampilan Halaman *Manage* Transaksi 90](#_Toc528134523)

[Gambar 5.64 Perancangan Tampilan Halaman *Login* Web 91](#_Toc528134524)

[Gambar 5.65 Perancangan Tampilan Halaman *Manage* Sayur Web 92](#_Toc528134525)

[Gambar 5.66 Perancangan Tampilan Halaman *Manage* Transakasi Web 93](#_Toc528134526)

[Gambar 5.67 Perancangan Tampilan Halaman *Manage* *User* Web 94](#_Toc528134527)

[Gambar 5.68 Perancangan Tampilan Halaman *Manage* *User* Web 95](#_Toc528134528)

[Gambar 5.69 Perancangan Tampilan Halaman Detail Sayur Web 96](#_Toc528134529)

[Gambar 5.70 Perancangan Tampilan Halaman Detail Transaski Web 97](#_Toc528134530)

[Gambar 6.1 Diagram Alir Implementasi 98](#_Toc528134531)

[Gambar 6.2 Kode Program *Login* 101](#_Toc528134532)

[Gambar 6.3 Kode Program Tambah Sayur 102](#_Toc528134533)

[Gambar 6.4 Kode Program *Edit* Data Sayur 103](#_Toc528134534)

[Gambar 6.5 Implementasi Halaman *Login* 104](#_Toc528134535)

[Gambar 6.6 Implementasi Halaman *Register* 104](#_Toc528134536)

[Gambar 6.7 Implementasi Halaman *Home User* 105](#_Toc528134537)

[Gambar 6.8 Implementasi Halaman Keranjang Belanja 106](#_Toc528134538)

[Gambar 6.9 Implementasi Halaman Transaksi 107](#_Toc528134539)

[Gambar 6.10 Implementasi Halaman Profil 108](#_Toc528134540)

[Gambar 6.11 Implementasi Halaman Gudang Sayur 109](#_Toc528134541)

[Gambar 6.12 Implementasi Halaman *Manage User* 109](#_Toc528134542)

[Gambar 6.13 Implementasi Halaman *Manage* Sayur 110](#_Toc528134543)

[Gambar 6.13 Implementasi Halaman *Manage* Transaksi 111](#_Toc528134544)

[Gambar 7.1 Diagram Alir Pengujian dan Analisis 112](#_Toc528134545)

# DAFTAR LAMPIRAN

[1. FOTO DOKUMENTASI KEGIATAN 136](#_Toc528134547)

[2. *USER* MANUAL 136](#_Toc528134548)

[1.1 Admin 136](#_Toc528134549)

[1.1.1 Bagian I – *Login* 136](#_Toc528134550)

[1.1.2 Bagian II – Menambahkan sayur. 139](#_Toc528134551)

[1.1.3 Bagian III – *Edit* sayur. 141](#_Toc528134552)

[1.1.4 Bagian IV – Hapus sayur 142](#_Toc528134553)

[1.1.5 Bagian V – Melihat dan menghapus daftar *user*. 143](#_Toc528134554)

[1.1.6 Bagian VI – Melihat dan mengubah status transaksi. 145](#_Toc528134555)

[*2.2.* *User* 147](#_Toc528134556)

[1.2.1 Bagian I – Login dan *Register*. 147](#_Toc528134557)

[1.2.2 Bagian II – Memesan sayur 149](#_Toc528134558)

[1.2.3 Bagian III – Melihat transaksi 150](#_Toc528134559)

[1.2.4 Bagian IV – *Edit* profil 152](#_Toc528134560)

[1.2.5 Bagian V – *Logout* 152](#_Toc528134561)

PENDAHULUAN

## Latar belakang

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

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

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

## 1.2 Rumusan masalah

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

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

## 1.3 Tujuan

Tujuan dari pembuatan aplikasi ini adalah sebagai berikut:

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

## 1.4 Manfaat

Penulisan dokumentasi ini memilki beberapa manfaat diantaranya :

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

## 1.5 Batasan masalah

Batasan masalah pada pengembangan ini sebagai berikut :

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

## 1.6 Sistematika pembahasan

Sistematika penulisan pada pengembangan aplikasi ini direncanakan sebagai berikut :

**BAB 1 PENDAHULUAN**

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

**BAB 2 PROFIL PERUSAHAAN**

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

**BAB 3 LANDASAN KEPUSTAKAAN**

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

**BAB 4 METODOLOGI**

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

**BAB 5 ANALISIS DAN PERANCANGAN**

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

**BAB 6 IMPLEMENTASI**

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

**BAB 7 PENGUJIAN DAN ANALISIS**

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

**BAB 8 PENUTUP**

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

## 1.7 Pelaksanaan Praktek Kerja Lapang

* **Waktu**

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

Tabel 1.1 Kegiatan

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NO | Kegiatan | Juli 2018 | | | | Agustus 2018 | | | |
| I | II | III | IV | II | III | IV | I |
| 1 | Pengenalan CV SASINDO secara garis besar dan ruang lingkup kerja | X |  |  |  |  |  |  |  |
| 2 | Penentuan objek yang akan dibuat | X |  |  |  |  |  |  |  |
| 3 | Pembagian tugas pengerjaan baik isi laporan dan kode sistem | X |  |  |  |  |  |  |  |
| 4 | Analisis kebutuhan *user* | 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 perusahan 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 membatu 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 :

Director

Administration & Marketing

IT Engineer

Customer Service

**Power House:**

2nd Floor, 49th Anggrek Garuda Street Malang, 65141

http://sasindo.id

##### 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*). Beberapas 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.

1. Pengembangan Aplikasi Dekstop & Perangkat Bergerak (Dekstop &

*Mobile Application*)

* + Aplikasi Antrean Puskesmas – *on going*

1. 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.

### 3.1.2 Sistem Jual Beli *Online*.

Sistem Jual Beli *Online* merupakan sistem jual beli atau transaksi yang dilakukan dengan bantuan teknologi informasi antara satu atau dua lebih organiasi 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;

1. 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 kompabilitas sebuah perangkat untuk menjaga ekosistem dari pengembangan *Android* memiliki lingkungan yang sehat dan stabil untuk para penggunanya (Android, 2018).



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

Sumber : Android (2018)

*Android* diimplementasikan dalam bentuk tumpukan arsitektur perangkat lunak atau sebuah *Software Stack* yang terdiri dari kernel Linux, perangkat *Runtime* dan *libraries* yang sesuai, *Framework* aplikasi dan sekumpulan aplikasi. Kernel Linux bekerja sebagai jembatan antara perangkat keras dan lapisan atas, ART melakukan penerjemahan *bytecode* menjadi perintah yang dikenali oleh *processor* perangkat, *Android Libraries* menyediakan bantuan untuk menangani perintah-perintah umum (*String*, jaringan, dan manipulasi file),*Native libraries* adalah sekumpulan layanan yang merupakan dari bahasa C/C++, *Android Framework* adalah serangkaian layanan yang secara kolektif membentuk lingkungan untuk menjalankan dan mengelola android, lalu yang terakhir aplikasi yang merupakan sekumpulan aplikasi android *native* dengan implementasi tertentu.Aplikasi yang terdapat dan dibuat pada android sebagian besar ditulis dalam Bahasa *Java* dan dikompilasi ke dalam format *bytecode* pada lingkungan pengembangan aplikasi dengan Android Studio dan ketika aplikasi dijalankan pada perangkat maka bytecode tadi akan dikompilasi oleh *ART* agar dapat dibaca oleh *CPU*. Tujuan utama dari pembuatan arsitektur *Android* yang seperti itu adalah untuk meningkatkan kinerja dan efisiensi, baik dalam mengeksekusi aplikasi yang telah dibuat dan dalam penerapan penggunaan kembali dan pengembangan dalam desain aplikasi android yang dibuat. (Smyth, 2015)

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

### 3.2.1 *Android Studio*

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

1. Sistem bangun berbasis *Gradle* yang fleksibel
2. Emulator yang kaya dan kaya fitur
3. Lingkungan terpadu tempat Anda dapat mengembangkan untuk semua perangkat *Android*
4. Instant Run untuk mendorong perubahan pada aplikasi yang sedang berjalan tanpa membangun *APK* baru
5. Template kode dan integrasi *GitHub* untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
6. Alat dan kerangka pengujian ekstensif
7. Alat serat untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya
8. Dukungan *C ++* dan *NDK*
9. Dukungan bawaan untuk *Google Cloud Platform*, sehingga mudah untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine.*

Dengan segala fitur yang telah diberikan dalam *IDE* android studio, hal tersebut memudahkan para pengembang perangkat lunak androiddalam melakukan penggembangan 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 diidentifikasikan 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 menggandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi yang kemudian akan diolah lagi oleh perangkat maupun aplikasi yang terhubung.

## 3.4 *Laravel*

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

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

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

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

1. *Model*

*Model* merupakan sebuah basis dimana pengembang dapat melakukan manipulasi data. *Model* sendiri berada di antara data dan aplikasi. Data dapat disimpan dalam berbagai jenis sistem *database* seperti *MySQL*, *file* *XML*, atau bahkan *Excel*.

1. *View*

*View* merupakan kumpulan kode yang bertugas untuk menampilkan sebuah tampilan visual data yang diterima *Controller* dari *Model*. Sedangkan dalam *Laravel* menggunakan template Blade untuk tampilan dengan menggunakan kode PHP seperti biasa.

1. *Controller*.

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

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

## 3.5 *Unified Modeling Language (UML)*

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

Tujuan utama pemanfaatan *UML* menurut Sugrue J. 2009 adalah sebagai berikut:

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

### 3.5.1 *Structured Diagram.*

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

#### *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)

#### *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 | *Class* | Notasi yang digunakan untuk menggambarkan sebuah objek, atribut, maupun operasi. |  |
| 2 | *Association* | Notasi yang digunakan untuk menunjukkan hubungan antar kelas, banyak kelas atau dengan kelas itu sendiri. |  |
| 3 | *Package* | Notasi yang digunakan untuk menunjukkan package atau kompulan dari kelas yang dibungkus dalam package. |  |
| 4 | *Generalization* | Notasi yang digunakan untuk menunjukkan hubungan antar kelas yang menggunakan pewarisan dari kelas utama ke kelas anak. |  |
| 5 | *Dependency* | Notasi yang menunjukkan hubungan kelas yang bergantung pada kelas lain. |  |
| 6 | *Agregation* | Notasi yang menunjukkan hubungan kelas yang terdiri dari beberapa komponen yang lebih kecil. |  |

Sumber : Dennis (2012)

### **3.5.2 *Behavior Diagram***.

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

#### *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 (*user*) atau sistem lain yang menggunakan fitur atau fungsi yang berada dalam sistem. Perlu dicatat bahwa *actor* berinteraksi dengan *use case* tetapi tidak memiliki kontrol atas *use case*. |  |
| 2 | *Use case* | Notasi yang digunakan untuk menggambarkan fungsionalitas atau fitur yang disediakan oleh sistem yang saling bertukar pesan antar unit dengan *actor* , yang dinyatakan dalam bentuk kata kerja. |  |
| 3 | *Asociation* | Notasi yang digunakan untuk menunjukkan hubungan antar siapa dan apa yang meminta interaksi secara langsung antar *use case* maupun *actor*. |  |
| 4 | *extend* | *Extend*, merupakan perluasan dari *use case* lain jika dalam kondisi tertentu atau memiliki syarat. |  |
| 5 | *include* | *Include*, merupakan di dalam *use case* lain atau pemanggilan *use case* oleh *use case* lain. |  |

Sumber : Dennis (2012)

#### *Sequence Diagram*

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



##### Gambar 3.6. *Sequence Diagram*

Sumber : UML Diagrams(2018)

Tabel 3.4 Notasi pada *Sequence Diagram*

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

Tabel 3.5 Notasi pada *Sequence Diagram* (lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | *Lifeline* | Notasi yang digunakan untuk menunjukkan lama hidup suatu *object* dalam *sequence*. |  |
| 4 | *Focus Control* | Notasi yang digunakan untuk menunjukkan bahwa sebuah *object* sedang aktif dan menjalankan tugasnya. |  |
| 5 | *Message* | Notasi yang digunakan untuk menunjukkan bahwa sebuahpesan sedang dikirim dari suatu *object* ke *object* yang lain. |  |

Sumber : Dennis (2012)

#### *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:

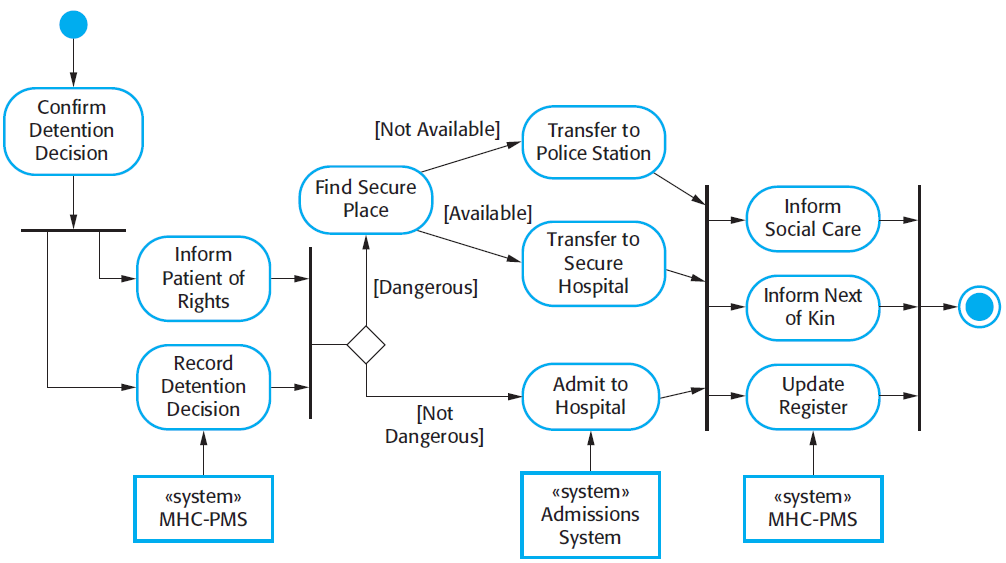


##### Gambar 3.7. *Use Case Scenario.*

Sumber : Bittner (2006)

#### *Activity Diagram*

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



##### Gambar 3.7. *Activity diagram*.

Sumber : Sommerville (2011)

## *Entity Relationship Diagram*

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

Tabel 3.6 Notasi pada *Entity Relationship Diagram*

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

Sumber : Dennis (2012)

BAB 4 METODOLOGI

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

Adapun diagram alir metodologi tersebut dijelaskan pada Gambar 4.1 berikut :

4.1 Penentuan Objek Penelitian

4.2 Studi Literatur

4.3 Analisa Kasus

4.4 Perancangan Konsep dan Model

4.5 Implementasi

4.6 Pengujian dan Analisis

4.7 Kesimpulan

##### Gambar 4.1 Diagram Alir Metodologi

## 4.1 Penentuan Objek Penelitian

Merupakan langkah awal yang bertujuan menentukan objek penentuan yang bisa didapat dari kegiatan PKL yang dilakukan di CV. Sarana Utama Solusindo yang kemudian bisa dijadikan suatu yang akan dikerjakan. Melakukan diskusi awal dengan *software developer* CV. Sarana Utama Solusindo yang bersangkutan, kemudian diberikan daftar objek apa saja yang dapat dijadikan sebagai objek untuk dikerjakan pada PKL ini.

## 4.2 Studi Literatur

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

## 4.3 Analisa Kasus

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

## 4.4 Perancangan Konsep dan Model

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

1. Menganalisa kebutuhan sistem dan menerjemahkan ke dalam fungsi fungsi pada aplikasi.
2. Menentukan objek aplikasi berdasarkan kebutuhan yang ada.
3. 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 pemograman *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 sepeti pada Gambar 5.1.

Gambar 5.1 Diagram Alir Analisis dan Perancangan

Analisis dan Perancangan

5.1 Analisa Kebutuhan Sistem

5.2 Analisa Kebutuhan Perangkat Lunak

5.3 Perancangan Perangkat Lunak

5.1.1 Deskripsi Umum Sistem

5.1.2 Deskripsi Umum Perangkat Lunak

5.1.3 Fungsi Perangkat Lunak

5.1.4 Karakteristik Perangkat Lunak

5.2.2 Kebutuhan Fungsional dan Non Funfsional

5.2.2 Pemodelan Kebutuhan

5.3.1 Perancangan Algoritma

5.3.2 Perancangan Antarmuka User

## 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 installasi 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 menggandung 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 ditunjukan pada gambar 5.2

Admin



User

Sayur Online

Web Service

Database

##### 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* adminatau *user* dan keluar sistem melalui fitur *logout*.

### 5.1.3 Fungsi Perangkat Lunak

Fungsi aplikasi sayur online ini diantaranya :

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

### 5.1.4 Karakteristik Pengguna

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

Tabel 5.1 Karakter Pengguna

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

## 5.2 Analisa Kebutuhan Perangkat Lunak

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

### 5.2.1 Kebutuhan Fungsional dan Non-Fungsional

**a) Kebutuhan Fungsional**

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

Tabel 5.2 Kebutuhan Fungsional

|  |  |  |  |
| --- | --- | --- | --- |
| No | Aktor | Nama Fungsi | Deskripsi |
| 1 | *User*/Admin | Login | Untuk menyeleksi pengguna yang memiliki autorisasi agar dapat mengakses ke dalam sistem dengan memasukkan *email*  dan *password* |
| 2 | *User*/Admin | Lihat daftar sayur | *User*/admin dapat melihat tampilan data sayur yang dijual dalam bentuk *grid layout*. |
| 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 transakasi  Pembelian jika barang sudah dikirim ataupun diterima |
| 6 | *User*/Admin | Lihat transaksi | *User*/Admin dapat melihar daftar transaksi di dalam menu transaksi, |
| 7 | *User*/Admin | Search sayur | *User*/Admin dapat mencari sayur yang diinginkan di halaman *home*. |

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 *User* | Admin dapat melakukan hapus terhadap data *user* yang  tersimpan dalam sistem |
| 12 | *User* /Admin | *Edit* profil | *User*/admin dapat melakukan *edit* *profile* untuk kelengkapan data dari pengguna user ataupun admin |
| 13 | *User*/Admin | Lihat profil | *User*/Admin dapat melihat profil |
| 14 | *User* | *Register* | *User* dapat melakukan pendaftaran pada aplikasi sayur online untuk membeli sayur yang tersedia. |
| 15 | *User* | Order sayur | *User* dapat memilih dan melakukan penyimpanan data sayur yang akan dibeli ke dalam fitur *cart*. |
| 16 | *User* | Melengkapi data transaksi | *User* dapat melakukan pembelian sayur dengan melengkapi data transaksi pada keranjang. |
| 17 | *User* | *Edit* jumlah sayur pada keranjang | *User* dapat mengubah jumlah sayur yang terdapat di dalam keranjang. |

Tabel 5.4 Kebutuhan Fungsional (lanjutan)

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

1. **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 ditunjukan pada tabel 5.5.

Tabel 5.5 Kebutuhan Non-Fungsional Sistem

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

### 5.2.2 Pemodelan Kebutuhan

#### 5.2.2.1 *Entity Relationship Diagram*

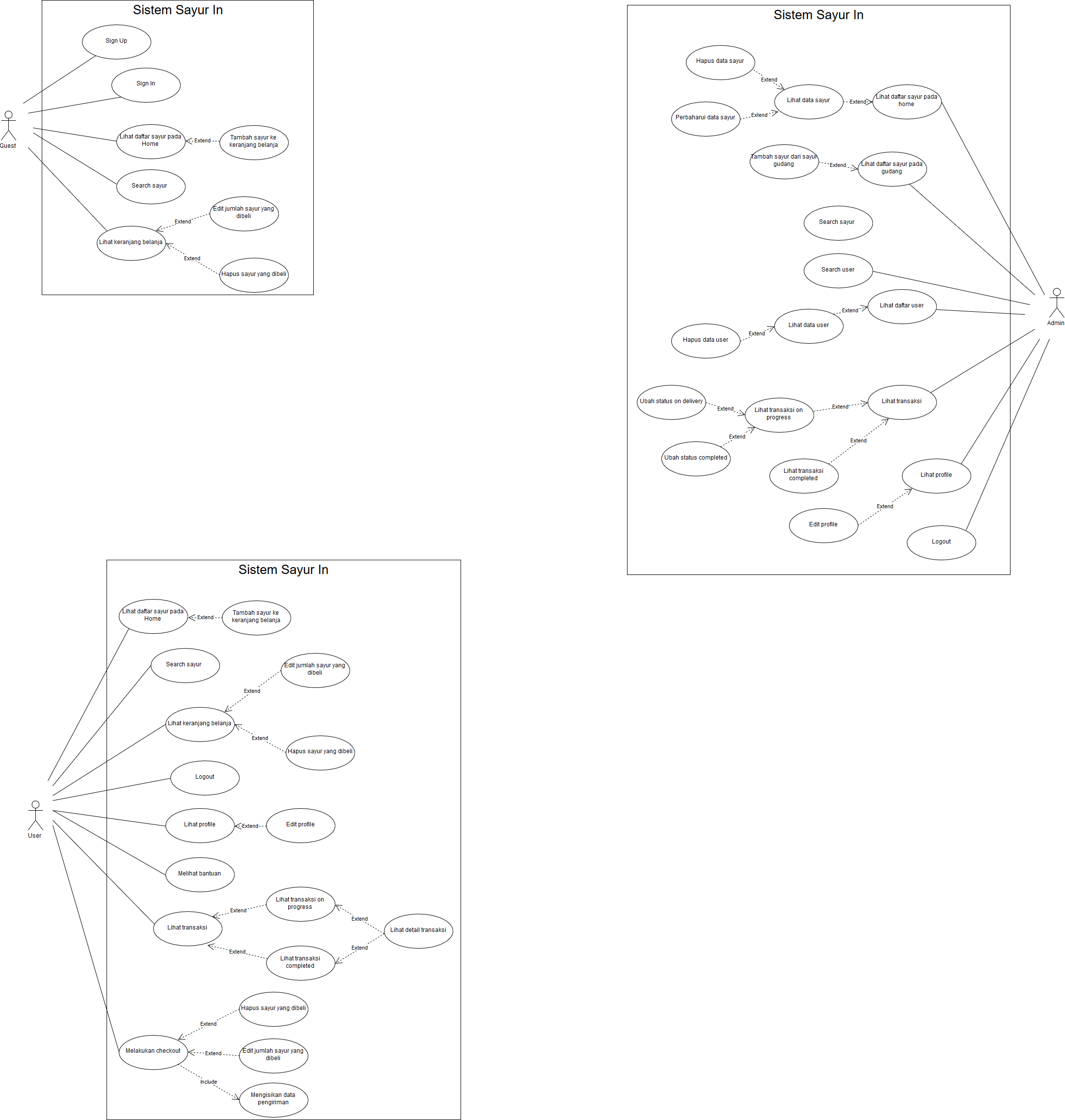
*Entity relationship diagram* digunakan untuk menggambarkan perencanaan data yang akan digunakan pada sistem. Berikut adalah *entity relationship diagram* dari aplikasi sayur online yang ditunjukkan pada gambar 5.3.



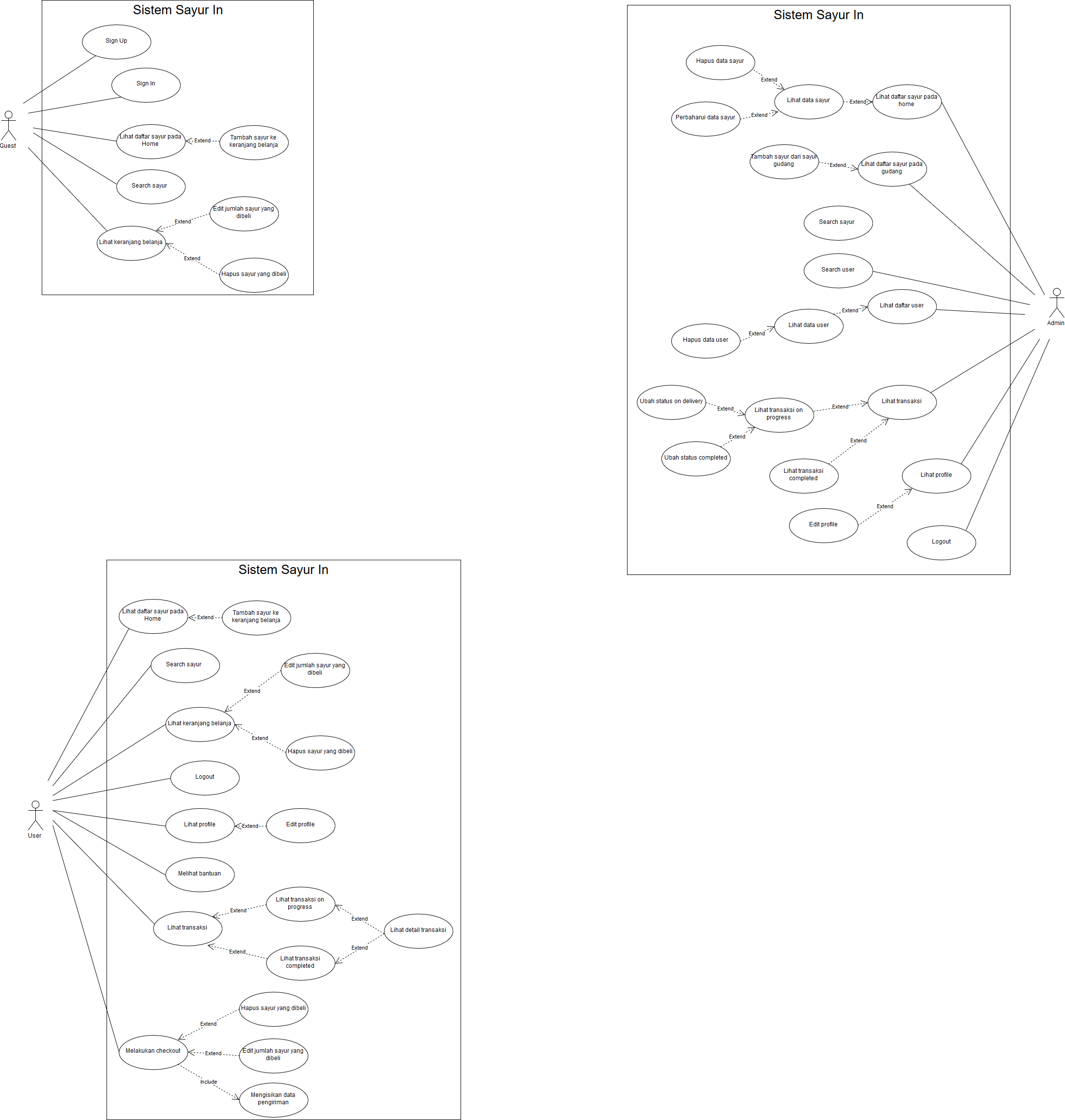
##### Gambar 5.3 *Entity Relationship Diagram* Untuk Aplikasi Sayur *Online*

#### 5.2.2.2 *Use Case Diagram*

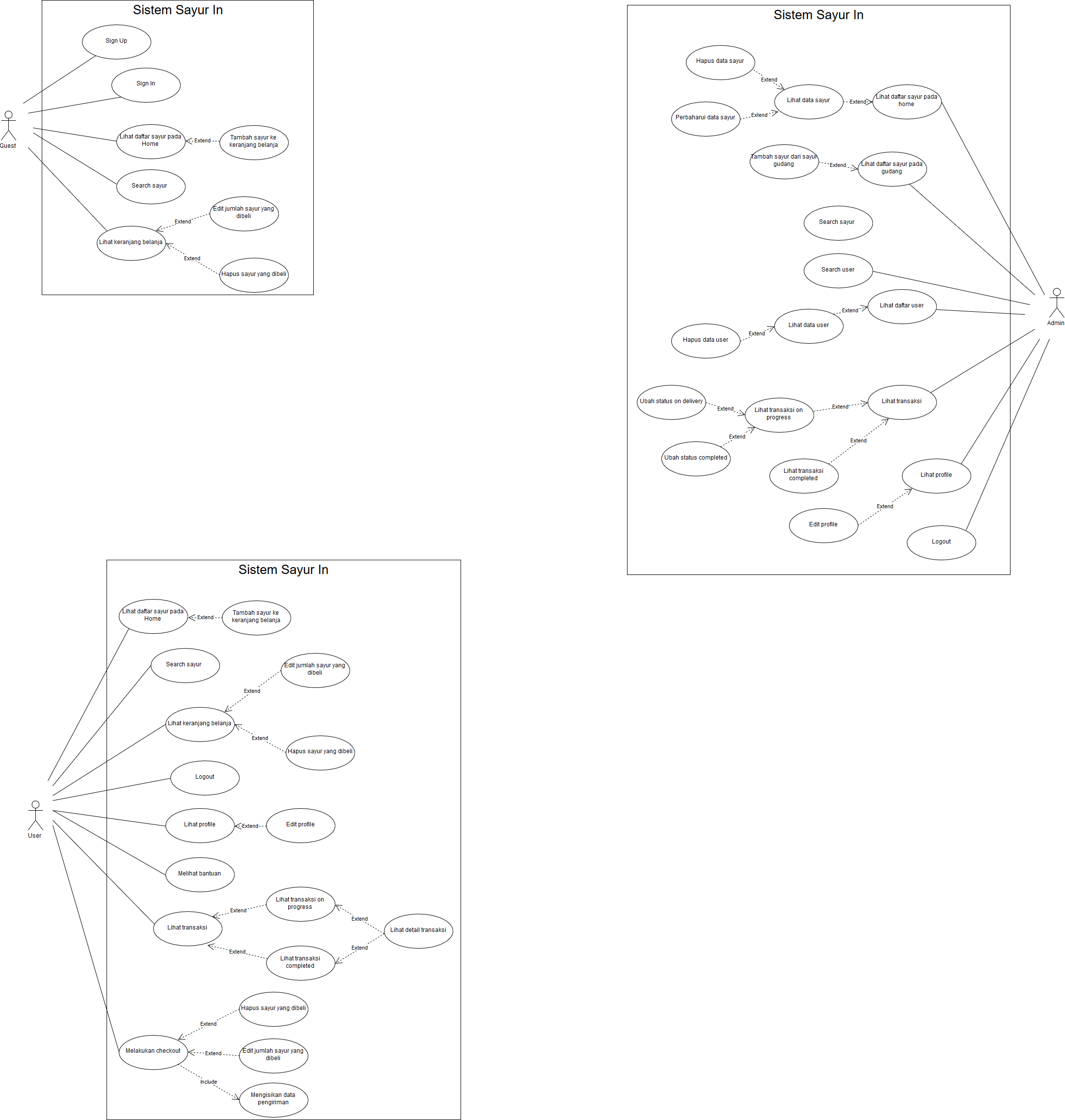
Diagram *use case* digunakan untuk menggambarkan kebutuhan-kebutuhan dan fungsionalitas dari sistem serta digunakan untuk menunjukan aksi-aksi yang dilakukan oleh aktor terhadap sistem. Berikut adalah use case diagram dari aplikasi sayur online yang ditunjukan pada gambar 5.4, 5.5, 5.6, dan 5.7.



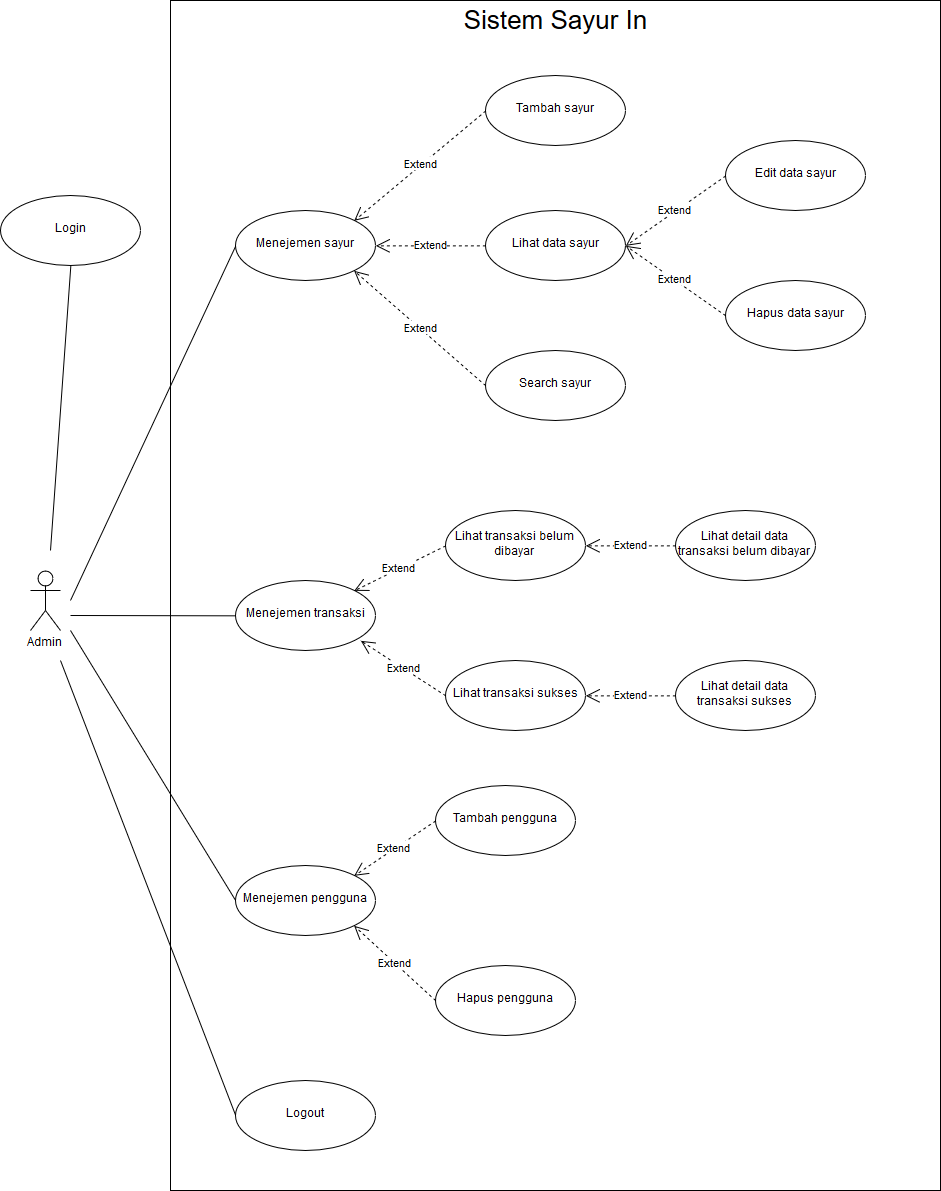
##### Gambar 5.4 *Use Case Diagram* Untuk Platform Mobile Dengan Aktor Guest



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



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



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

#### 5.2.2.3 *Use Case Skenario*

Untuk menjelaskan diagram *use case* secara terperinci dalam skenario *use case*. Penggunaan skenario *use case* ini bertujuan untuk mendapatkan deskripsi secara global mengenai *use case*, kondisi awal dan akhir yang harus dipenuhi oleh *use case* setelah fungsionalitas selesai dijalankan. Dalam skenario ini akan diulas bagaimana tanggapan sistem terhadap aksi yang dilakukan oleh aktor. Berikut adalah skenario *use case* dari aplikasi sayur online pada platform mobile yang ditunjukan pada tabel 5.6 – 5.31 dan skenario *use case* dari sayur online pada platform web yang ditujukan pada tabel 5.32-5.44.

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

Tabel 5.6 Skenario *Use Case* *Sign in*

|  |  |
| --- | --- |
| *Use case* *Sign in.* | |
| Tujuan | Masuk ke dalam sistem. |
| Aktor | Guest. |
| Kondisi awal | Halaman *Sign in.* |
| Skenario Utama | 1. Guest mengisi data pada kolom yang kosong, 2. Data yang dimasukkan tervalidasi 3. Menuju halaman home. |
| Skenario Alternatif | 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*

|  |  |
| --- | --- |
| *Use case* *Sign up.* | |
| Tujuan | Mendaftar ke dalam sistem. |
| Aktor | Guest. |
| Kondisi awal | Halaman *Sign in.* |
| Skenario Utama | 1. Menekan tombol *sign up.* 2. Guest mengisi kolom dengan lengkap dan format yang benar. 3. Guest menekan tombol sign up. 4. Guest terdaftar dan dapat *login* pada halaman *sign in*. |
| Skenario Alternatif | 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*

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

Tabel 5.9 Skenario *Use Case* Tambah keranjang belanja

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

Tabel 5.10 Skenario *Use Case* hapus keranjang belanja

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

Tabel 5.11 Skenario *Use Case* *Search* sayur

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

Tabel 5.12 Skenario *Use Case* Lihat Keranjang belanja

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

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

|  |  |
| --- | --- |
| *Use case* *edit* jumlah sayur pada keranjang*.* | |
| Tujuan | Mengubah jumlah sayur pada keranjang belanja. |
| Aktor | *User*. |
| Kondisi awal | Halaman *Checkout.* |
| 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

|  |  |
| --- | --- |
| *Use case* hapus daftar pembelian*.* | |
| Tujuan | Menghapus daftar pembelian*.* |
| Aktor | *User*. |
| Kondisi awal | Halaman *Checkout.* |
| 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

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

Tabel 5.16 Skenario *Use Case* Lihat transaksi.

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

Tabel 5.17 Skenario *Use Case* Melihat detail transaksi.

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

Tabel 5.18 Skenario *Use Case* lihat bantuan

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

Tabel 5.19 Skenario *Use Case* lihat profil

|  |  |
| --- | --- |
| *Use case* Lihat profil. | |
| Tujuan | Melihat profil *user*. |
| Aktor | *User*/Admin. |
| Kondisi awal | Halaman *Home.* |
| Skenario Utama | 1. *User*/Admin Menekan menu profil. |
| Skenario Alternatif | - |
| Kondisi akhir | Menampilkan halaman profil. |

Tabel 5.20 Skenario *Use Case* *Edit* Profil

|  |  |
| --- | --- |
| *Use case* *edit* profil. | |
| Tujuan | Melakukan edit profil. |
| Aktor | *User*/Admin. |
| Kondisi awal | Halaman Profil*.* |
| Skenario Utama | 1. *User* menekan icon camera untuk mengganti foto *user* 2. *User* dapat langsung *mengedit* nama, *email*, dan no telp pada *field* nama, *email*, dan no telp 3. *User* menekan tombol simpan |
| Skenario Alternatif | - |
| Kondisi akhir | Profil berubah. |

Tabel 5.21 Skenario *Use Case* *Logout*.

|  |  |
| --- | --- |
| *Use case* *logout*. | |
| Tujuan | Keluar sistem. |
| Aktor | *User*/Admin. |
| Kondisi awal | Halaman Profil*.* |
| Skenario Utama | 1. *User* tombol logout. |
| Skenario Alternatif | - |
| Kondisi akhir | Keluar dari sistem. |

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

|  |  |
| --- | --- |
| *Use case* lihat daftar sayur pada gudang | |
| Tujuan | Melihat daftar sayur pada gudang |
| Aktor | Admin. |
| Kondisi awal | Halaman Home*.* |
| Skenario Utama | 1. Admin menekan tombol menu gudang. |
| Skenario Alternatif | - |
| Kondisi akhir | Menampilkan daftar sayur pada gudang |

Tabel 5.23 Skenario *Use Case* Menambah sayur.

|  |  |
| --- | --- |
| *Use case* Menambah sayur. | |
| Tujuan | Keluar sistem. |
| Aktor | Admin. |
| Kondisi awal | Halaman Profil*.* |
| Skenario Utama | 1. Admin menekan tombol menu gudang. 2. Admin memilih sayur yang akan ditambahkan dengan menekan tombol “Add” 3. Muncul pesan “Apakah anda yakin ingin menambahkan sayur?” 4. JIka admin memilih “Iya” maka sayur ditambahkan ke halaman home. |
| Skenario Alternatif | - |
| Kondisi akhir | Sayur ditambahkan ke halaman *home*. |

Tabel 5.24 Skenario *Use Case* Hapus Data sayur

|  |  |
| --- | --- |
| *Use case* hapus data sayur. | |
| Tujuan | Menghapus data sayur. |
| Aktor | Admin. |
| Kondisi awal | Halaman Home*.* |
| Skenario Utama | 1. Admin menekan tombol lihat pada sayur. 2. Admin menekan tombol hapus. |
| Skenario Alternatif | - |
| Kondisi akhir | Sayur berhasil dihapus. |

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

|  |  |
| --- | --- |
| *Use case* *Edit* sayur. | |
| Tujuan | Melakukan edit data sayur. |
| Aktor | Admin. |
| Kondisi awal | Halaman Home*.* |
| Skenario Utama | 1. Admin menekan tombol lihat pada sayur. 2. Admin menekan tombol edit. 3. Mengisi data sayur. |
| Skenario Alternatif | - |
| Kondisi akhir | Sayur berhasil di-*edit* |

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

|  |  |
| --- | --- |
| *Use case* 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*

|  |  |
| --- | --- |
| *Use case* 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*

|  |  |
| --- | --- |
| *Use case* 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.

|  |  |
| --- | --- |
| *Use case* mengubah status transaksi. | |
| Tujuan | Mengubah status transaksi. |
| Aktor | Admin. |
| Kondisi awal | Halaman *history.* |
| Skenario Utama | 1. Admin memilih daftar transaksi yang ingin diubah pada tab *On Progress*. 2. Memilih status transaksi yang tersedia pada bgaian bawah halaman. |
| Skenario Alternatif | - |
| Kondisi akhir | Status transaksi berhasil berubah. |

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

|  |  |
| --- | --- |
| *Use case* Lihat transaksi *On progress*. | |
| Tujuan | Melihat transaksi yang berstatus *on progress* |
| Aktor | *User*/Admin. |
| Kondisi awal | Halaman *history.* |
| Skenario Utama | 1. Memilih tab *on progress* pada menu transaksi. |
| Skenario Alternatif | - |
| Kondisi akhir | Status transaksi berhasil berubah. |

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

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

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

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

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

Tabel 5.33 Skenario *Use Case* Manajemen sayur.

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

Tabel 5.34 Skenario *Use Case* tambah sayur.

|  |  |
| --- | --- |
| *Use case* tambah sayur. | |
| Tujuan | Menambahkan sayur pada gudang. |
| Aktor | Admin. |
| Kondisi awal | Halaman manage sayur*.* |
| Skenario Utama | 1. Admin menekan tombol tambah pada manage sayur. 2. Akan ditampilkan form untuk tambah sayur, admin mengisi data foto, nama, harga, stok dan kuantitas sayur dengan lengkap. 3. Menekan tombol tambah. |
| Skenario Alternatif | - |
| Kondisi akhir | Sayur ditambahkan dan menampilkan halaman *manage* sayur. |

Tabel 5.35 Skenario *Use Case* lihat data sayur.

|  |  |
| --- | --- |
| *Use case* lihat sayur. | |
| Tujuan | Melihat detail data sayur pada gudang. |
| Aktor | Admin. |
| Kondisi awal | Halaman manage sayur*.* |
| Skenario Utama | 1. Admin memilih sayur yang ingin dilihat pada daftar. |
| Skenario Alternatif | - |
| Kondisi akhir | Menampilkan halaman detail sayur. |

Tabel 5.36 Skenario *Use Case* edit data sayur.

|  |  |
| --- | --- |
| *Use case* edit data sayur. | |
| Tujuan | Melakukan edit data sayur pada gudang. |
| Aktor | Admin. |
| Kondisi awal | Halaman detail sayur*.* |
| Skenario Utama | 1. Admin menekan tombol ubah pada halaman. 2. Akan ditampilkan form untuk melakukan *edit* sayur, admin mengisi data yang ingin diubah dengan lengkap. 3. Menekan tombol ubah. |
| Skenario Alternatif | - |
| Kondisi akhir | Sayur diubah dan menampilkan halaman *manage* sayur. |

Tabel 5.37 Skenario *Use Case* hapus data sayur.

|  |  |
| --- | --- |
| *Use case* hapus data sayur. | |
| Tujuan | Melakukan hapus data sayur pada gudang. |
| Aktor | Admin. |
| Kondisi awal | Halaman detail sayur*.* |
| Skenario Utama | 1. Admin menekan tombol hapus pada halaman. 2. Muncul pesan “Apakah anda yakin ingin menghapus sayur?” 3. Admin menekan tombol “yakin”. |
| Skenario Alternatif | - |
| Kondisi akhir | Sayur dihapus dan menampilkan halaman *manage* sayur. |

Tabel 5.38 Skenario *Use Case search* sayur.

|  |  |
| --- | --- |
| *Use case* *search* sayur. | |
| Tujuan | Mencari sayur pada gudang. |
| Aktor | Admin. |
| Kondisi awal | Halaman manage sayur*.* |
| Skenario Utama | 1. Admin mengisikan keyword pada kolom *search* lalu menekan enter. |
| Skenario Alternatif | - |
| Kondisi akhir | Menampilkan sayur yang dicari. |

Tabel 5.39 Skenario *Use Case* manajemen transaksi.

|  |  |
| --- | --- |
| *Use case* manajemen transaksi. | |
| Tujuan | Melakukan manajemen transaksi pada web. |
| Aktor | Admin. |
| Kondisi awal | Halaman home*.* |
| Skenario Utama | 1. Admin menekan menu *manage* transaksi pada menu web. |
| Skenario Alternatif | - |
| Kondisi akhir | Menampilkan daftar transaksi. |

Tabel 5.40 Skenario *Use Case* lihat detail transaksi.

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

Tabel 5.41 Skenario *Use Case* manajemen pengguna.

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

Tabel 5.42 Skenario *Use Case* hapus pengguna.

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

Tabel 5.43 Skenario *Use Case* tambah pengguna.

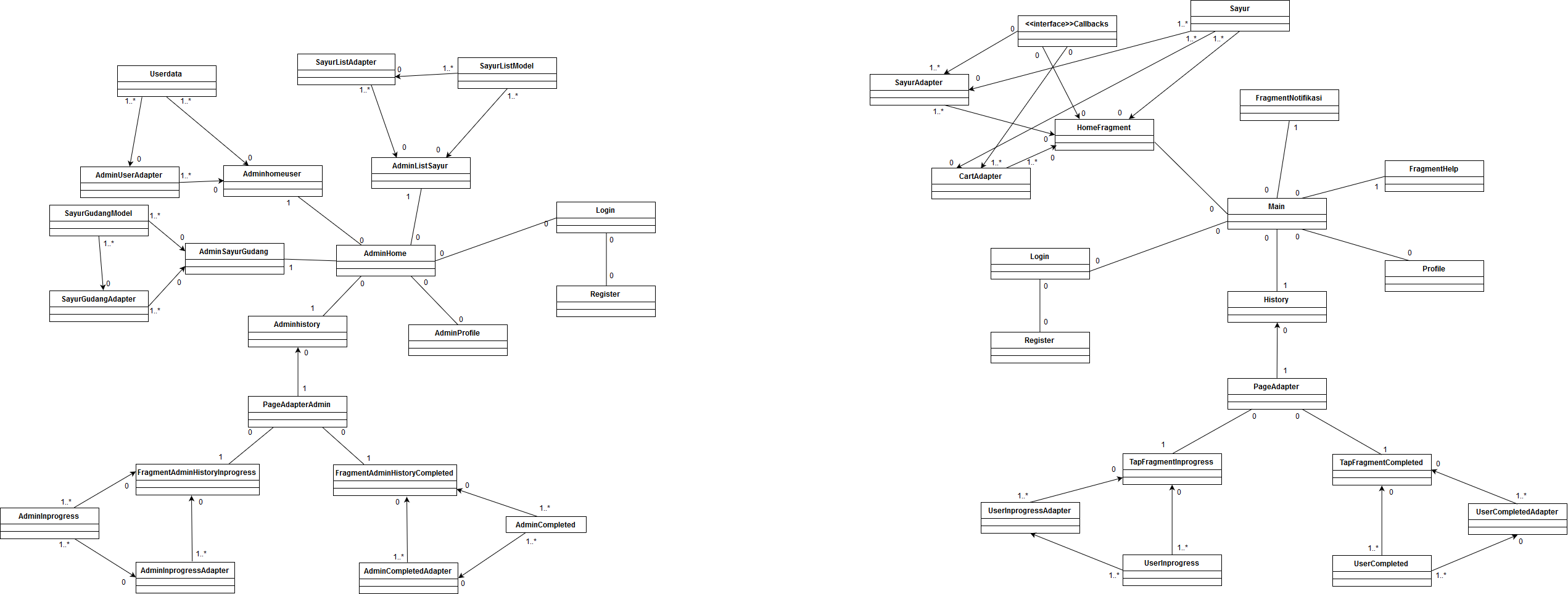
|  |  |
| --- | --- |
| *Use case* tambah pengguna. | |
| Tujuan | Menambahkan pengguna dari web. |
| Aktor | Admin. |
| Kondisi awal | Halaman manage *user.* |
| Skenario Utama | 1. Admin menekan tombol tambah pada halaman. 2. Akan ditampilkan form untuk mengisi data pengguna, admin mengisikan data dengan lengkap. 3. Admin menekan tombol tambah. |
| Skenario Alternatif | - |
| Kondisi akhir | Pengguna berhasil ditambahkan |

Tabel 5.44 Skenario *Use Case logout*.

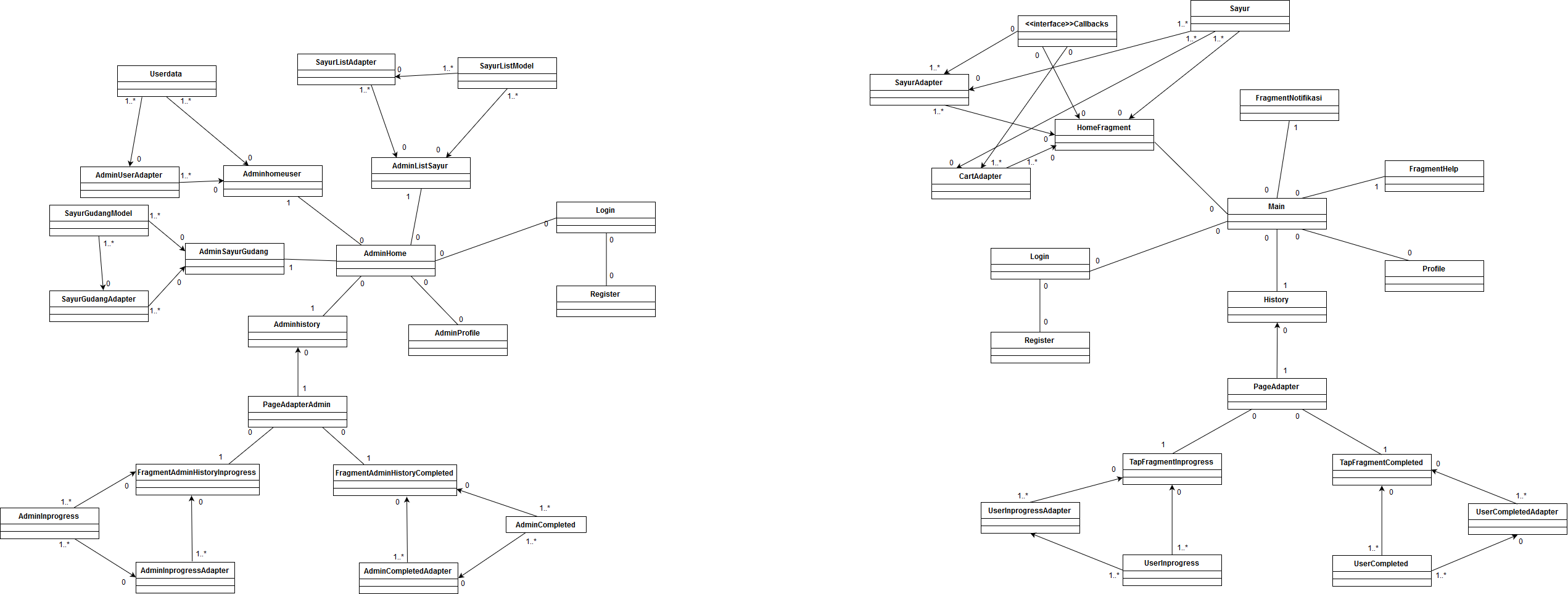
|  |  |
| --- | --- |
| *Use case* logout. | |
| Tujuan | Keluar dari sistem web. |
| Aktor | Admin. |
| Kondisi awal | Halaman home*.* |
| Skenario Utama | 1. Admin menekan tombol *hello* pada *navigation bar*, lalu menekan pilihan *logout*. |
| Skenario Alternatif | - |
| Kondisi akhir | Berhasil keluar sistem. |

#### 5.2.2.4 *Class Diagram*

Class *diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti komposisi, asosiasi, pewarisan dan lain-lain yang menyusun sebuah perangkat lunak. *Class diagram* dirancang dari hasil analisis terhadap *use case* yang dimodelkan sebelumya. Berikut Gambar 5.4 merupakan class diagram aplikasi Sayur Online :



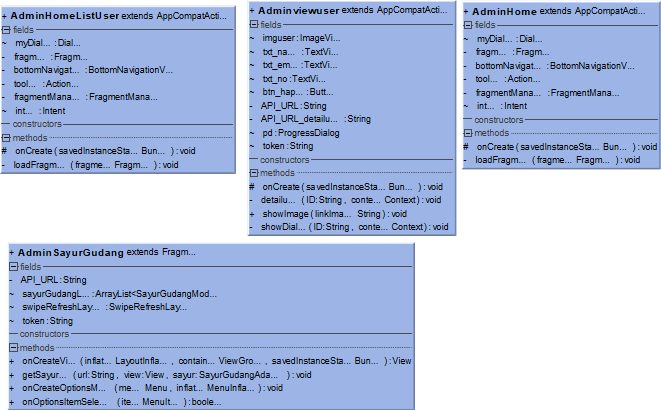
##### Gambar 5.4 *Class Diagram* Untuk Platform *Android* Pada Bagian *User*



##### Gambar 5.5 *Class Diagram* Untuk Platform *Android* Pada Bagian Admin



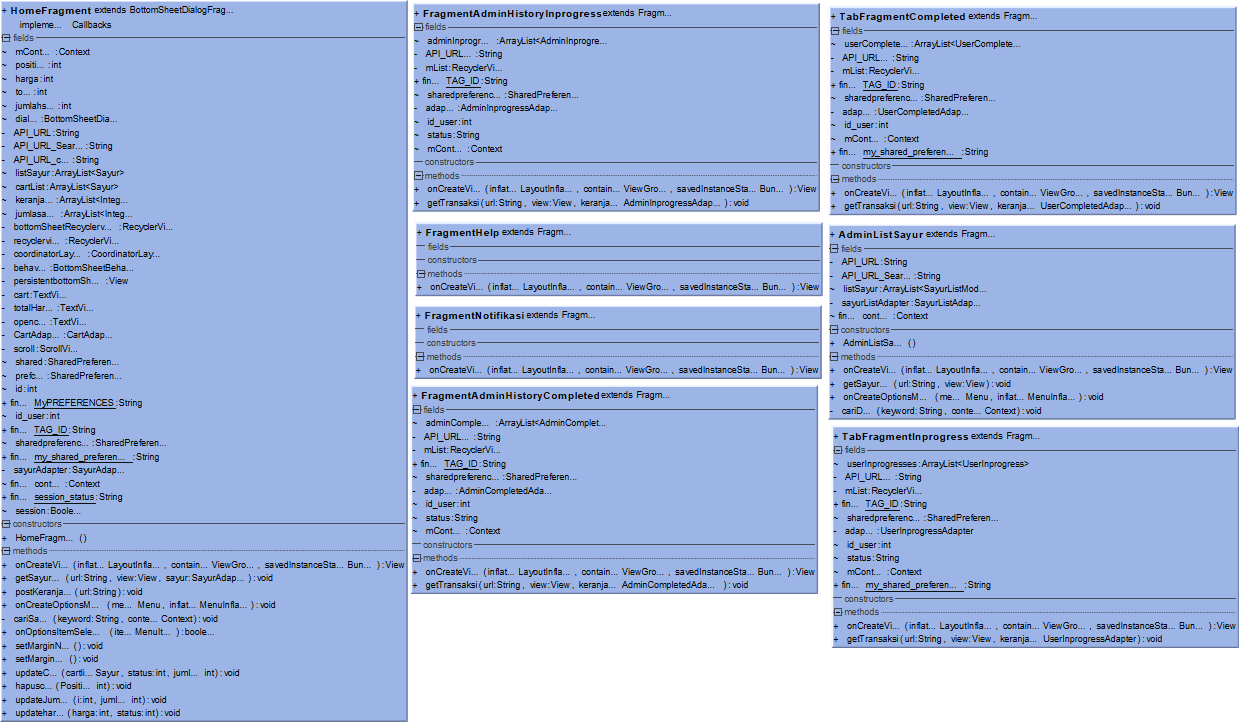
##### Gambar 5.6 *Class Diagram* Untuk Platform *Android* Pada *Package* Admin Bagian 1



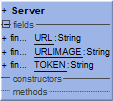
##### Gambar 5.7 *Class Diagram* Untuk Platform *Android* Pada *Package* Admin Bagian 2



##### Gambar 5.8 Detail *Class Diagram* Untuk Platform *Android* Pada *Package* *Adapter*



##### Gambar 5.9 Detail *Class Diagram* Untuk Platform *Android* Pada *Package* *Fragment*



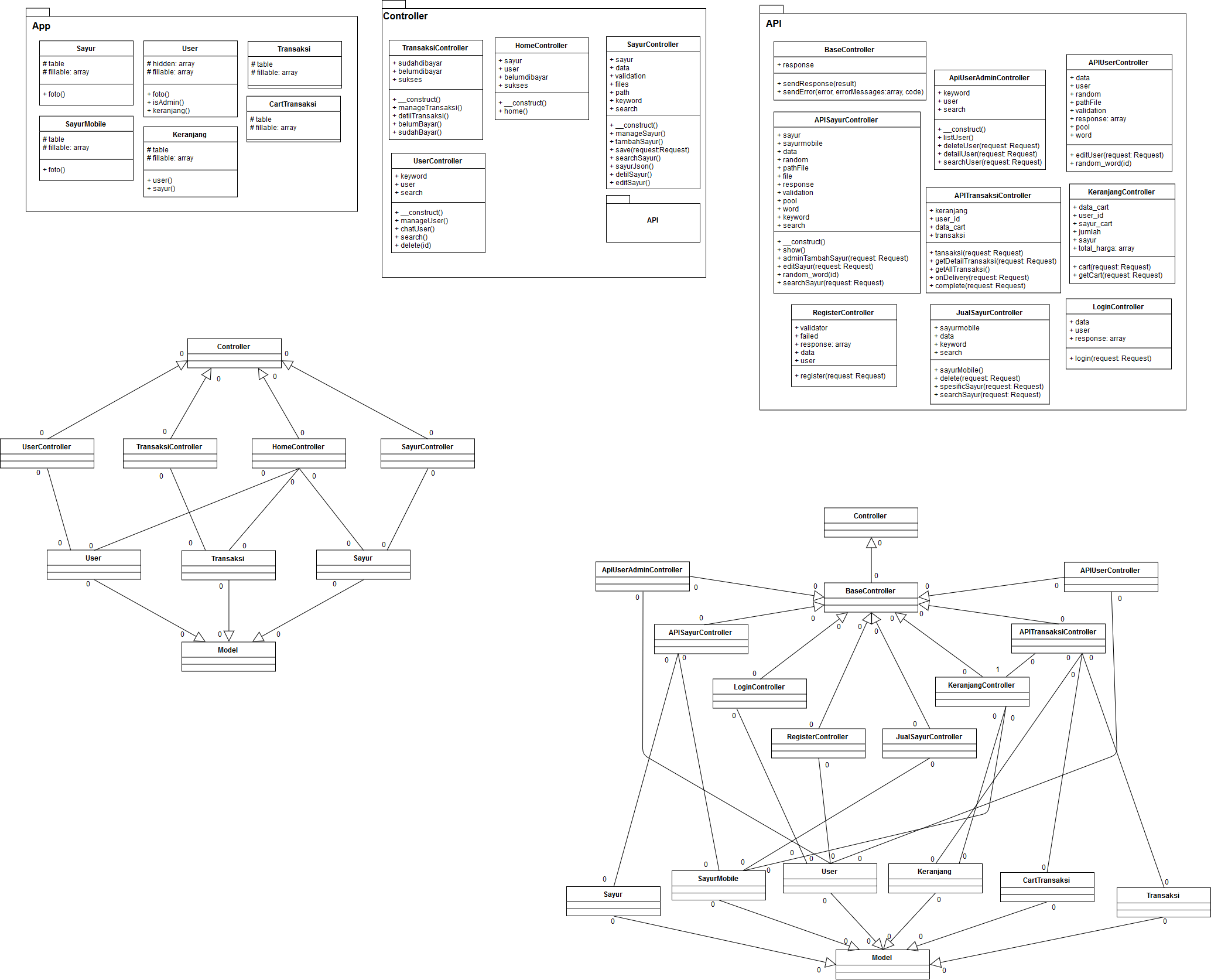
##### Gambar 5.10 Detail *Class Diagram* Untuk Platform *Android* Pada *Package Server*



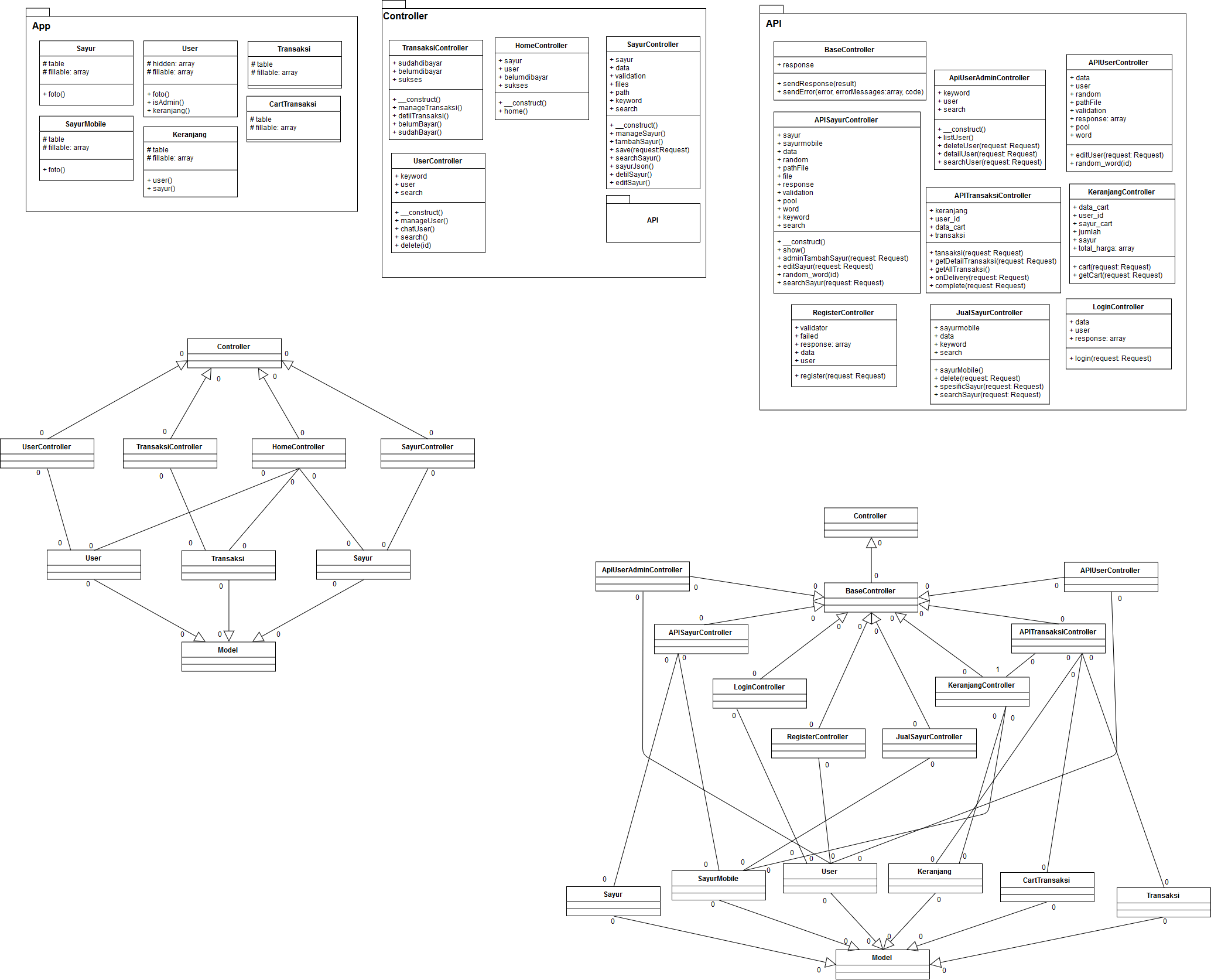
##### Gambar 5.11 Detail *Class Diagram* Untuk Platform *Android* Pada *Package Singleton*



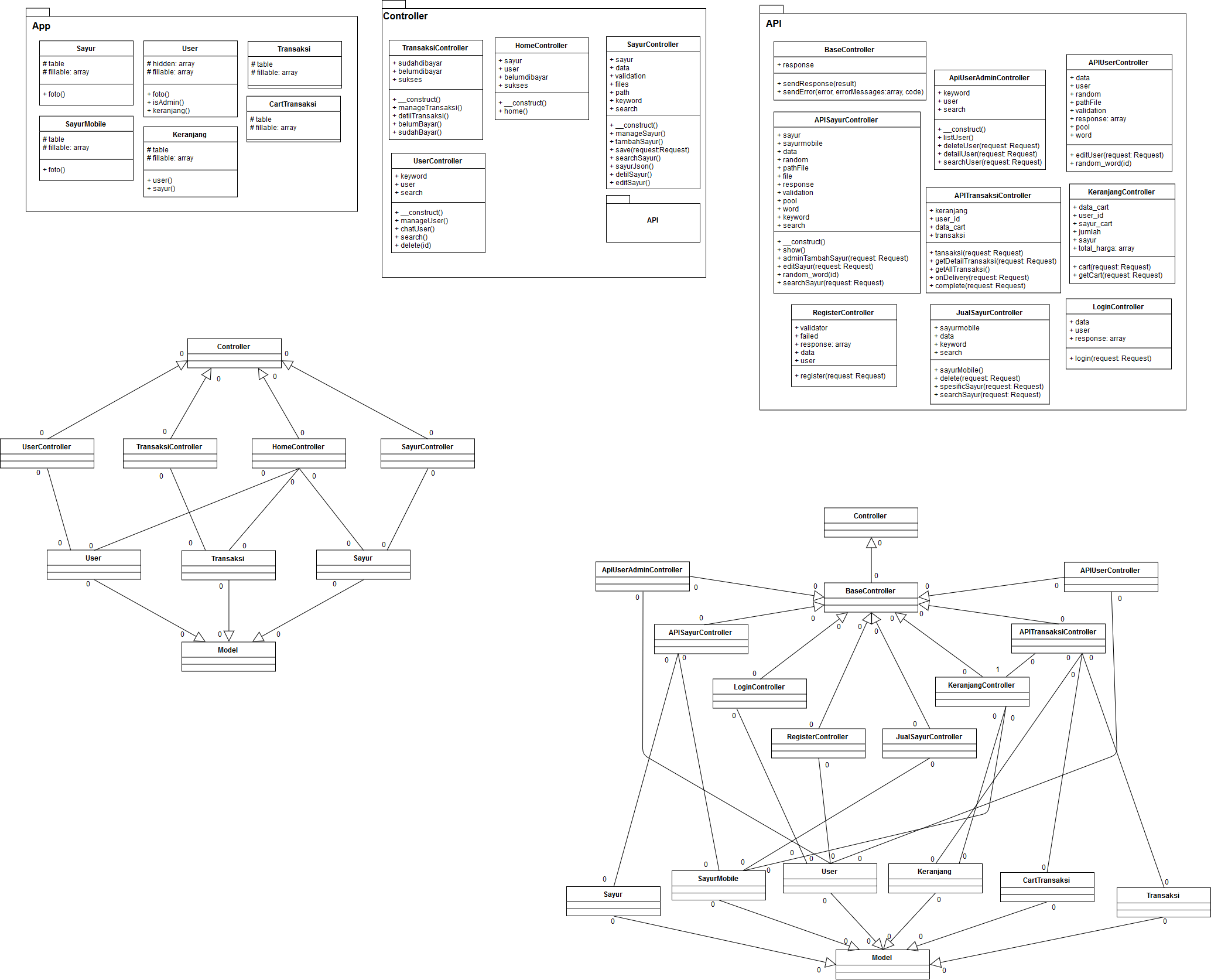
##### Gambar 5.12 Detail *Class Diagram* Untuk Platform *Android* Pada *Package drawersayur*



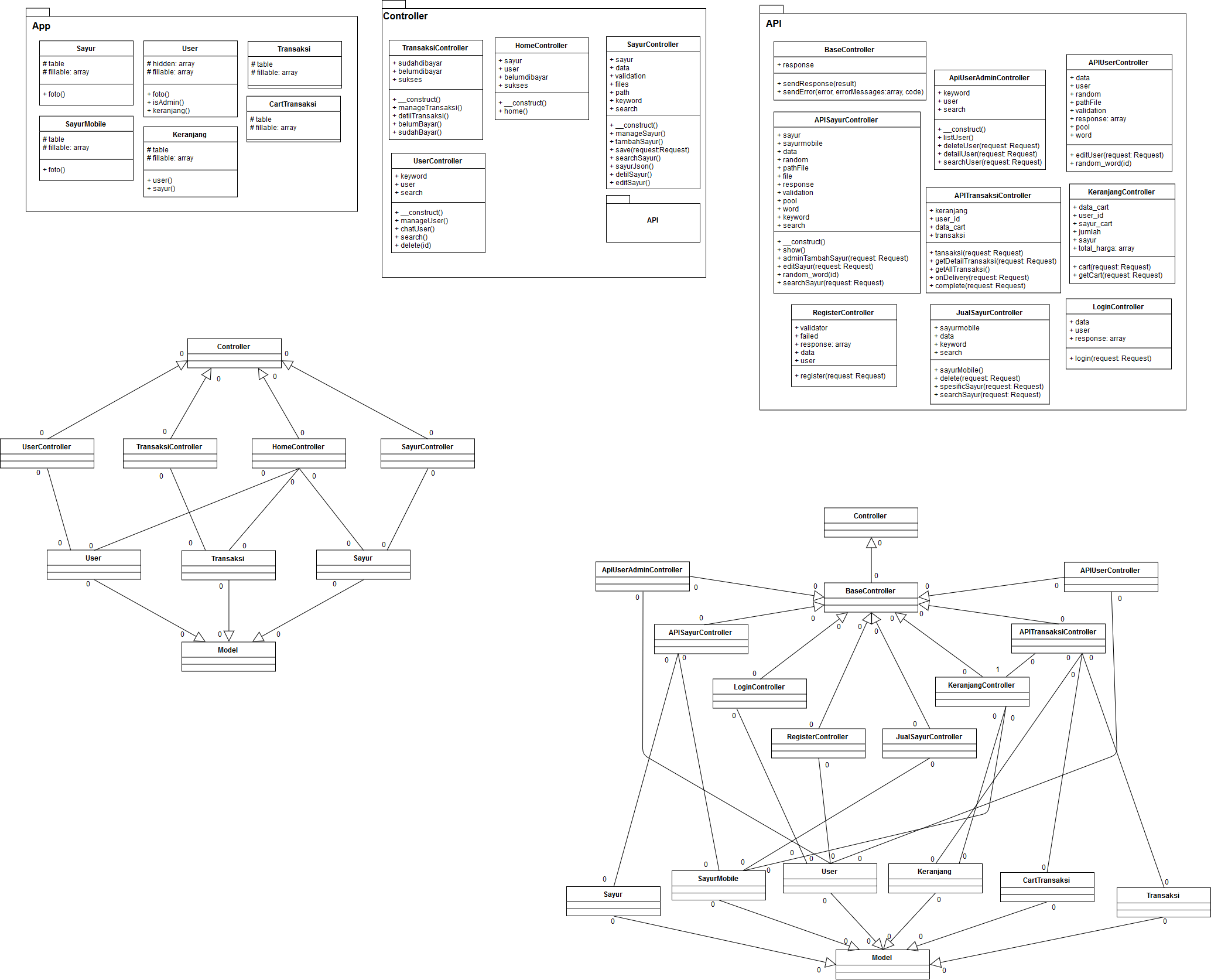
##### Gambar 5.13 *Class Diagram* Untuk Platform Web Untuk Proses Pada Website



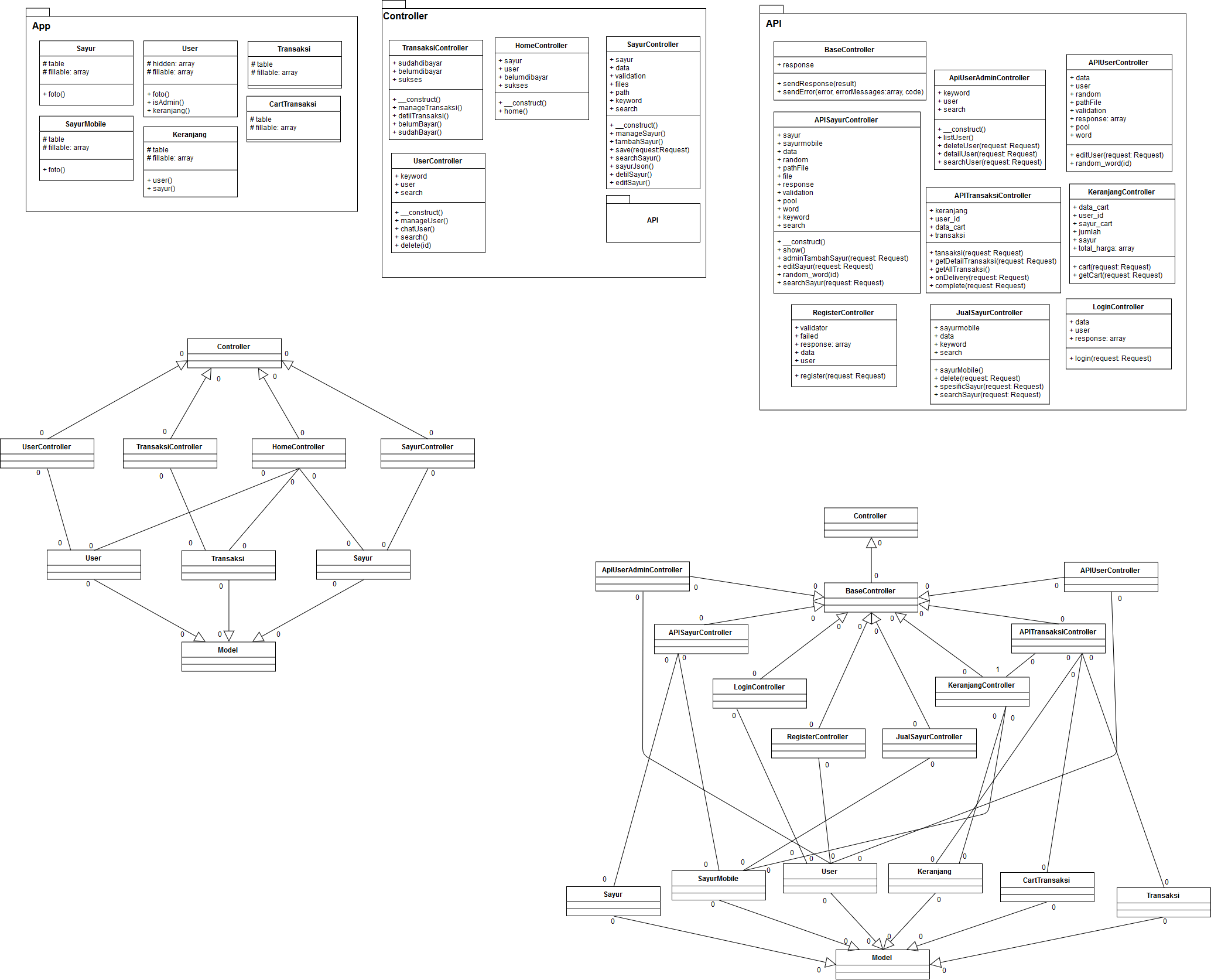
##### Gambar 5.14 *Class Diagram* Untuk Platform Web Untuk Proses *Web Service*



##### Gambar 5.15 Detail *Class Diagram* Untuk Platform Web Pada *Folder Controller*



##### Gambar 5.16 Detail *Class Diagram* Untuk Platform Web Pada *Folder API*



##### Gambar 5.17 Detail *Class Diagram* Untuk Platform Web Pada *Folder App*

#### 5.2.2.5 *Activity Diagram*

*Activity diagram* merupakan alur kerja atau kegiatan dari sebuah sistem atau menu yang ada pada pernagkat lunak. *Activity diagram* juga digunakan untuk mendefinisikan urutan atau pengelompokan tampilan dari sistem/**user** interface dimana setiap aktivitas dianggap memiliki sebuah rancang antar muka tampilan serta rancang menu yang ditampilkan pada perangkat lunak. Gambar 5.18 merupakan activity diagram dari hapus *user* :



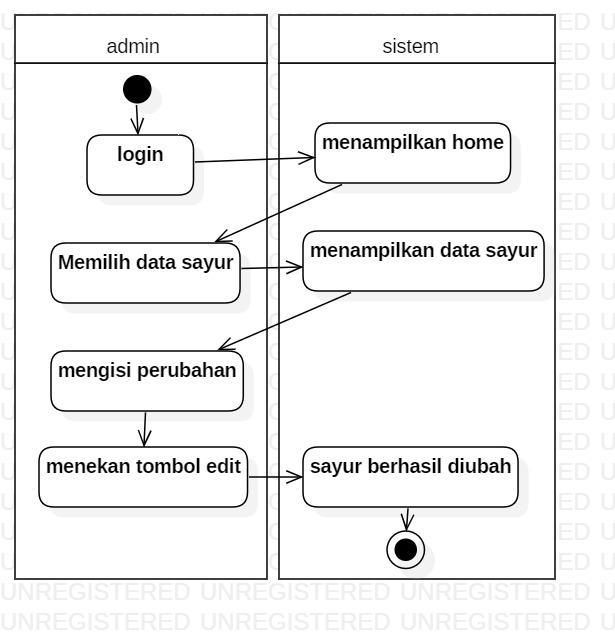
##### Gambar 5.18 *Activity Diagram* “hapus data *user*”

*Activity Diagram search user* merupakan alur kerja program untuk mencari dengan spesifik data *user* yang terdaftar. Berikut Gambar 5.19 activity diagram dari search *user* :



##### Gambar 5.19 *Activity Diagram* “*search* data *user*”

*Activity Diagram perbarui data sayur* merupakan alur kerja program untuk merubah/meng-*edit* data sayur yang dijual. Berikut Gambar 5.20 *activity diagram* dari perbarui data sayur :



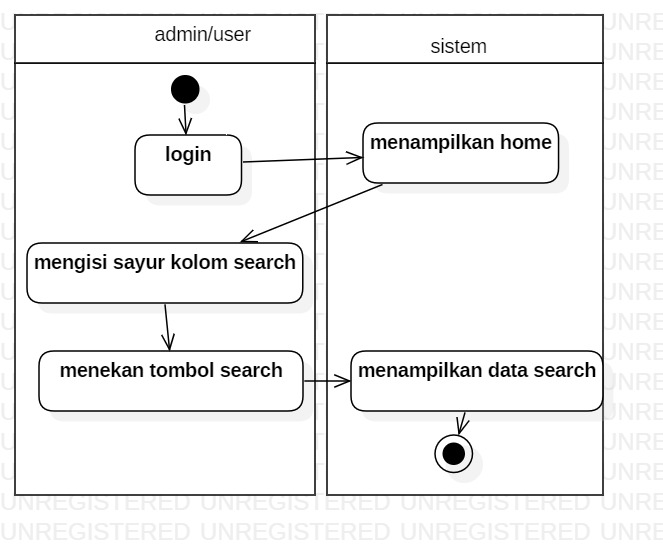
##### Gambar 5.20 *Activity Diagram* “perbarui data sayur”

*Activity Diagram hapus sayur* merupakan alur kerja program untuk menghapus data sayur yang dijual. Berikut Gambar 5.21 activity diagram dari hapus sayur :



##### Gambar 5.21 *Activity Diagram* “hapus data sayur”

*Activity Diagram search* sayurmerupakan alur kerja program untuk mencari dengan spesifik data sayur yang dijual. Berikut Gambar 5.22 *activity diagram* dari search sayur :



##### Gambar 5.22 Activity Diagram “*search* sayur”

*Activity Diagram* menambahsayurmerupakan alur kerja program untuk menambah data sayur dari gudang *webservice*. Berikut Gambar 5.23 activity diagram dari menambah sayur :

****

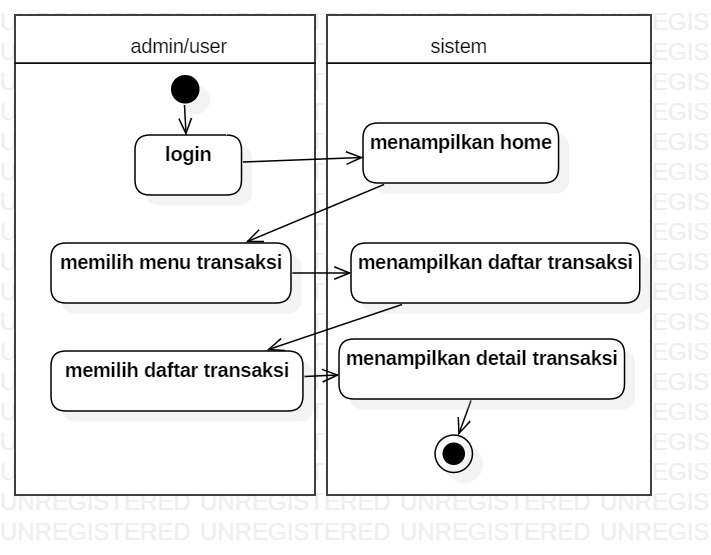
##### Gambar 5.23 Activity Diagram “menambah sayur”

*Activity Diagram* pesansayurmerupakan alur kerja program untuk *user*/pengguna untuk memesan sayur yang telah dijual di aplikasi sayur online. Berikut Gambar 5.24 *activity diagram* dari pesan sayur :



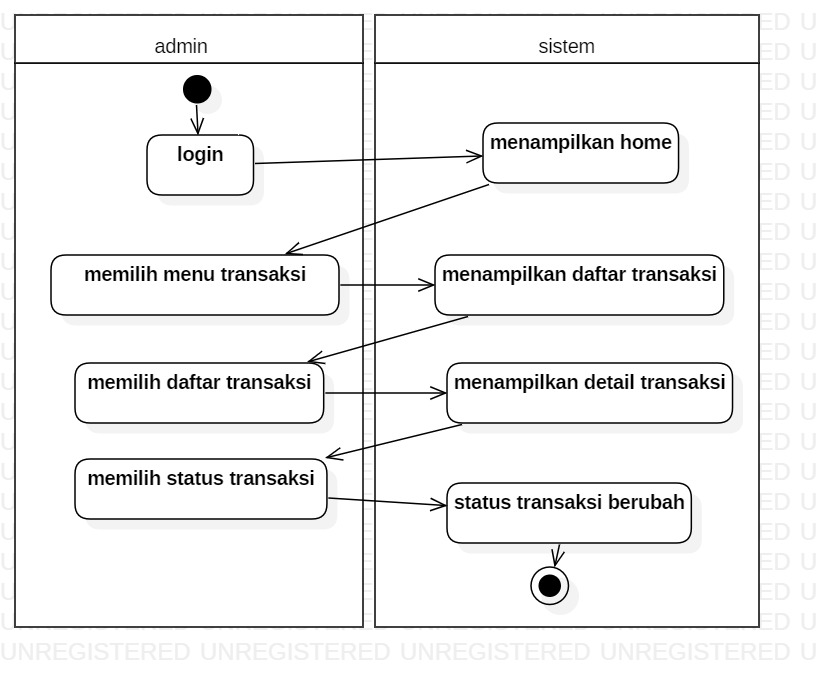
##### Gambar 5.24 Activity Diagram “pesan sayur”

*Activity Diagram* lihat transaksi merupakan alur kerja program untuk melihat *history* transaksi yang dilakukan sebelumnya oleh *user*. Berikut Gambar 5.25 *activity diagram* lihat daftar transaksi :



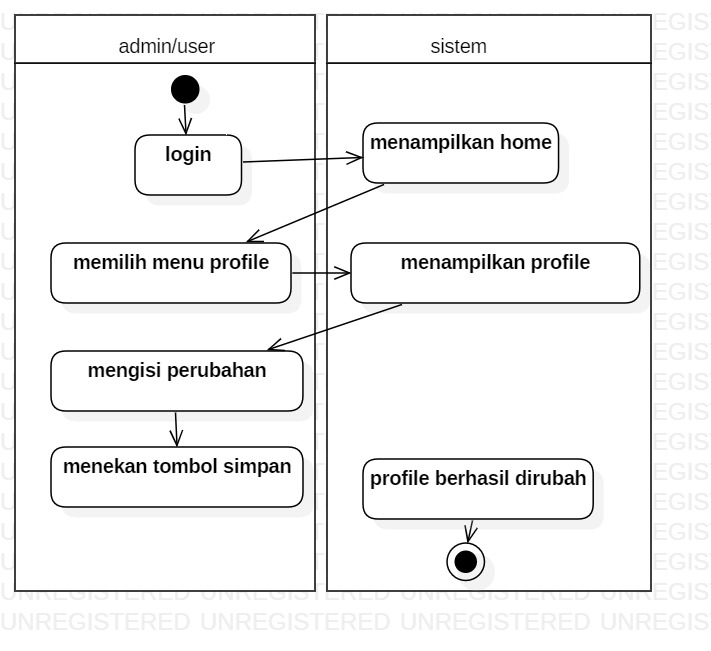
##### Gambar 5.25 *Activity Diagram* “melihat daftar transaksi”

*Activity Diagram* ubah status transaksi merupakan alur kerja program untuk mengubah status transaksi yang dilakukan sebelumnya oleh *user*. Berikut Gambar 5.26 *activity diagram* lihat daftar transaksi :



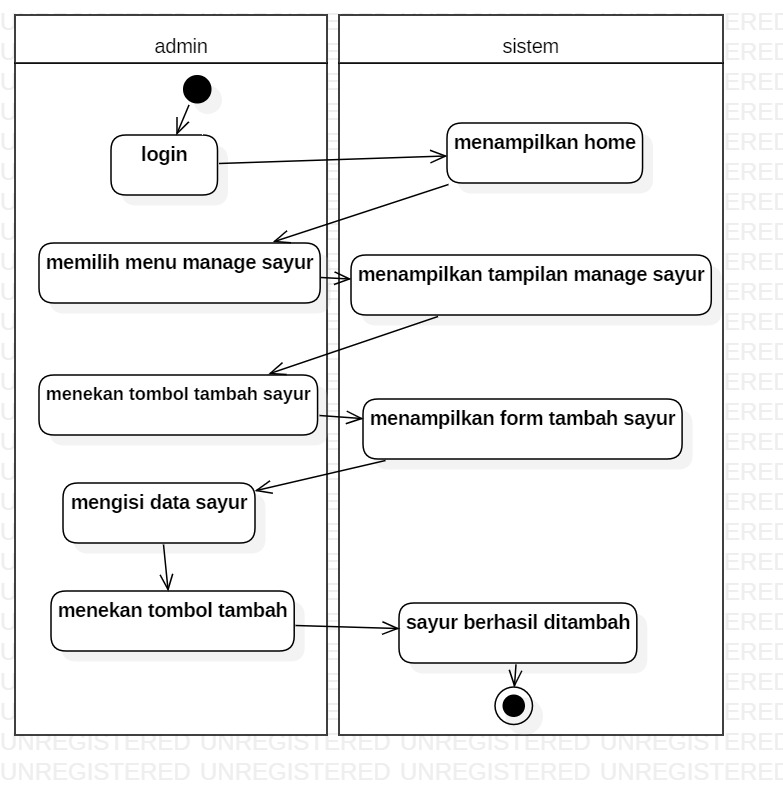
##### Gambar 5.26 *Activity Diagram* “melihat daftar transaksi”

*Activity Diagram* hapus daftar pembelianmerupakan alur kerja program untuk melakukan edit pada bagian profile. Berikut Gambar 5.27 *activity diagram* hapus daftar pembelian :



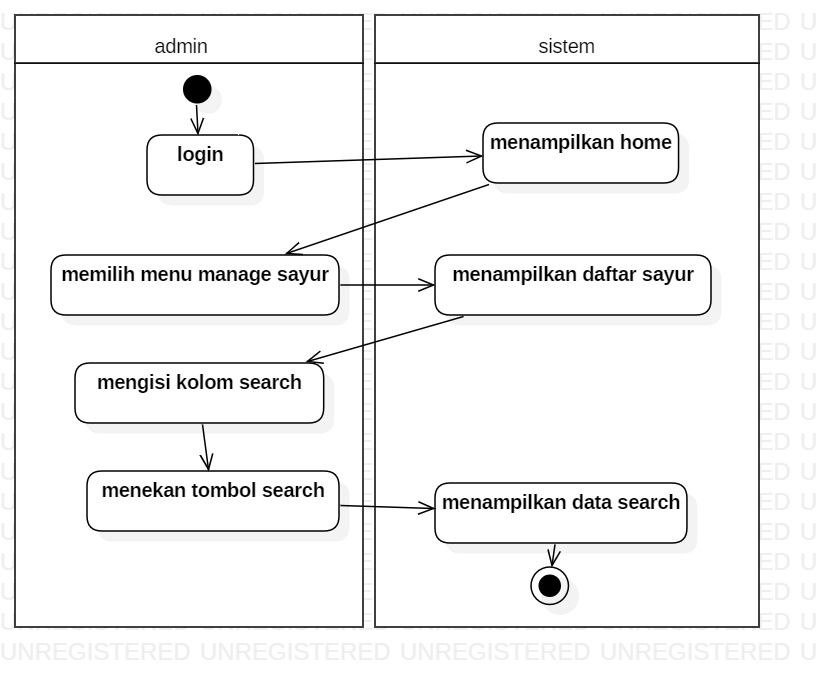
##### Gambar 5.27 *Activity Diagram* “*edit* profil”

*Activity diagram* menambah sayur pada web merupakan alur dari proses menambahkan sayur pada gudang dengan web. Berikut Gambar 5.28 *activity diagram* tambah sayur dengan web:



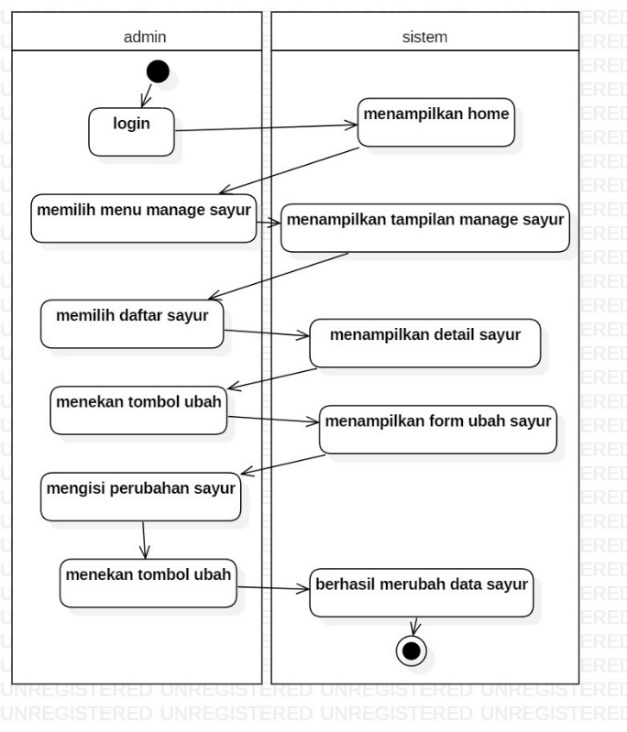
##### Gambar 5.28 *Activity Diagram* “tambah sayur pada web”

*Activity diagram* search sayur pada web merupakan alur dari proses mencari daftar sayur yang telah ditambahkan pada web. Berikut Gambar 5.29 *activity diagram* search sayur pada web:



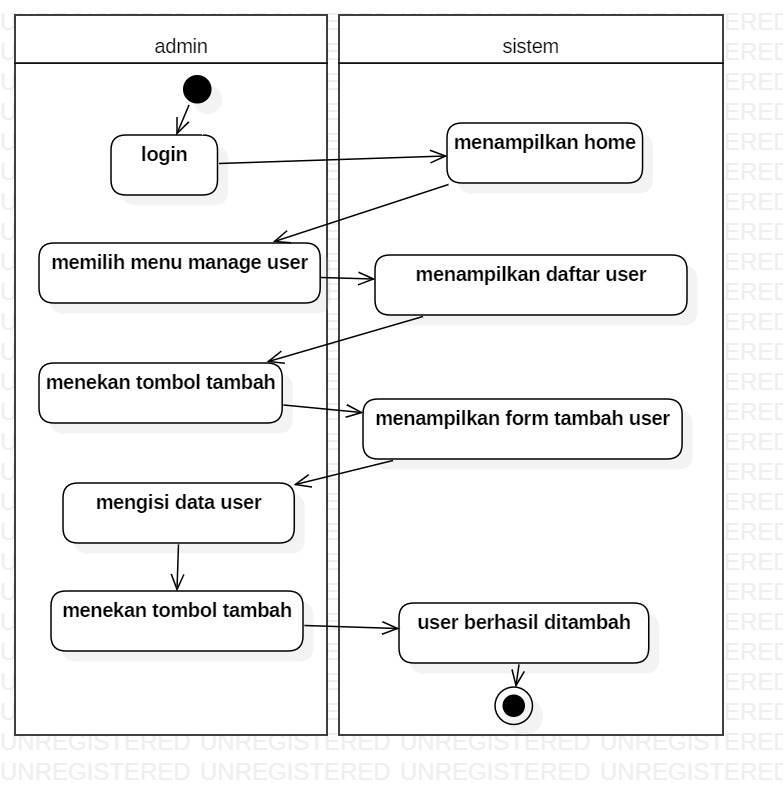
##### Gambar 5.29 *Activity Diagram* “*search* sayur pada web”

*Activity diagram* ubah sayur pada web merupakan alur dari proses melakukan perubahan pada daftar sayur yang telah ditambahkan pada web. Berikut Gambar 5.30 *activity diagram* ubah sayur pada web:



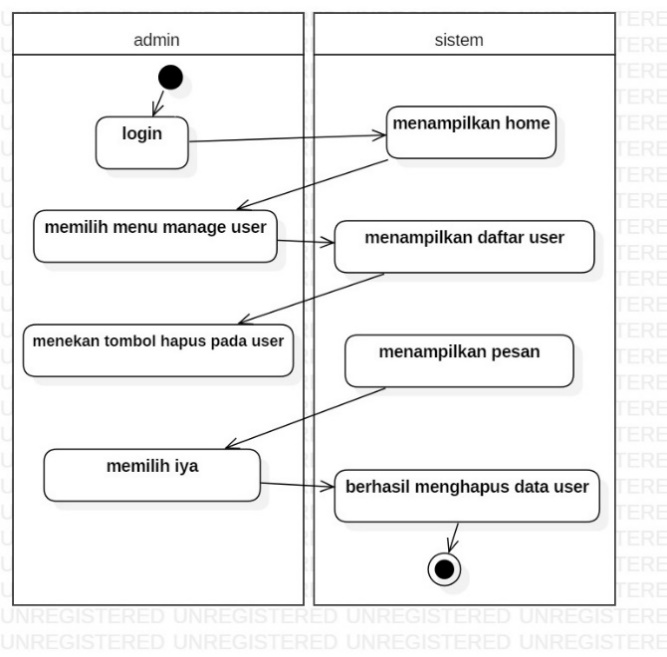
##### Gambar 5.30 *Activity Diagram* “ubah sayur pada *web*”

*Activity diagram* tambah *user* pada web merupakan alur dari proses melakukan penambahan *user* pada web. Berikut Gambar 5.31 *activity diagram* tambah *user* pada web:



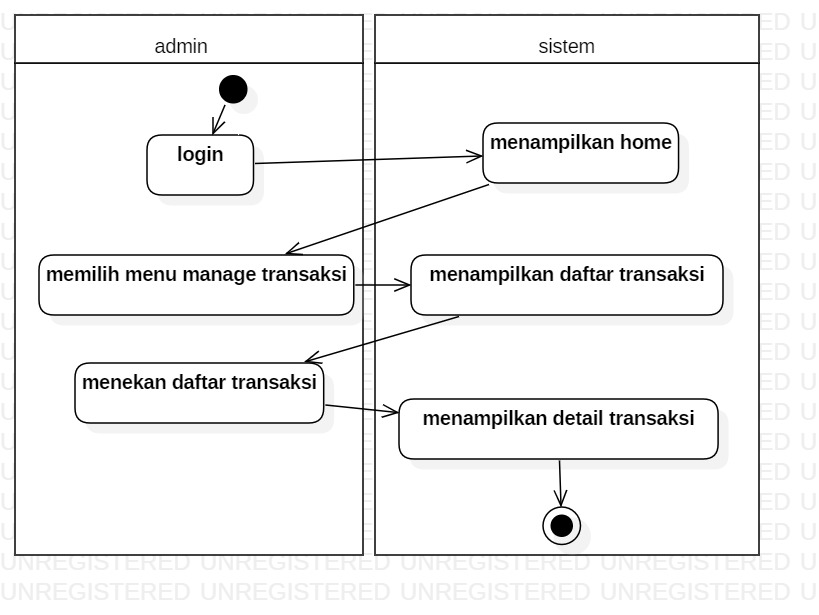
##### Gambar 5.31 *Activity Diagram* “tambah *user* pada web”

*Activity diagram* hapus *user* pada web merupakan alur dari proses melakukan penghapusan *user* yang telah terdaftar pada web. Berikut Gambar 5.32 *activity diagram* hapus *user* pada web:



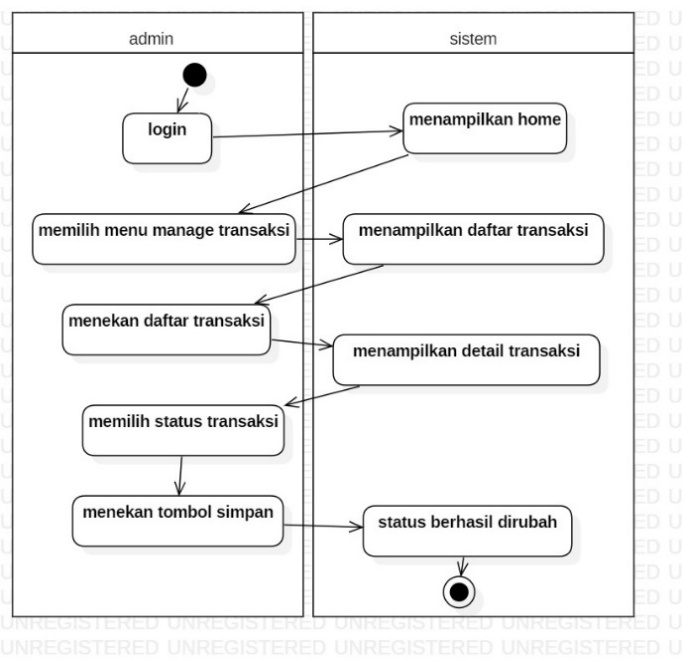
##### Gambar 5.32 *Activity Diagram* “hapus *user* pada web”

*Activity diagram* lihat data transaksi pada web merupakan alur dari proses melihat transaksi yang dilakukan *user* dan telah terdaftar pada *web*. Berikut Gambar 5.33 *activity diagram* lihat data transaksi pada web:



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

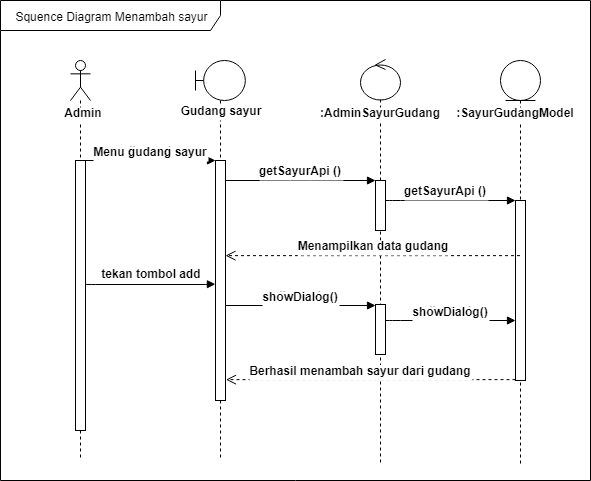
*Activity diagram* ubah status transaksi pada web merupakan alur dari proses merubah status transaksi yang dilakukan *user* dan telah terdaftar pada web. Berikut Gambar 5.34 *activity diagram* lihat data transaksi pada web:



##### Gambar 5.34 Activity Diagram “Ubah data transaksi pada web”

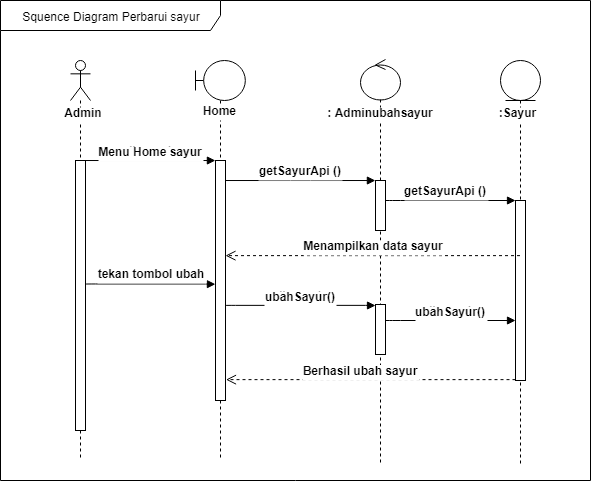
#### 5.2.2.6 *Sequence Diagram*

*Sequence diagram* digunakan untuk menggambarkan perilaku sebuah skenario. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Gambar 5.35 merupakan *sequence diagram* menambah sayur :



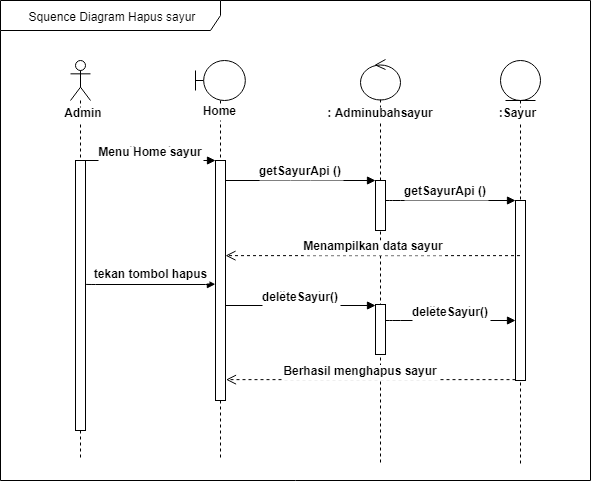
##### Gambar 5.35 *Sequence Diagram* “menambah sayur”

*Sequence Diagram* perbarui data sayurmerupakan interaksi antara objek untuk memperbarui/meng-*edit* data sayur. Berikut Gambar 5.36 merupakan *sequence diagram* perbarui data sayur :

**

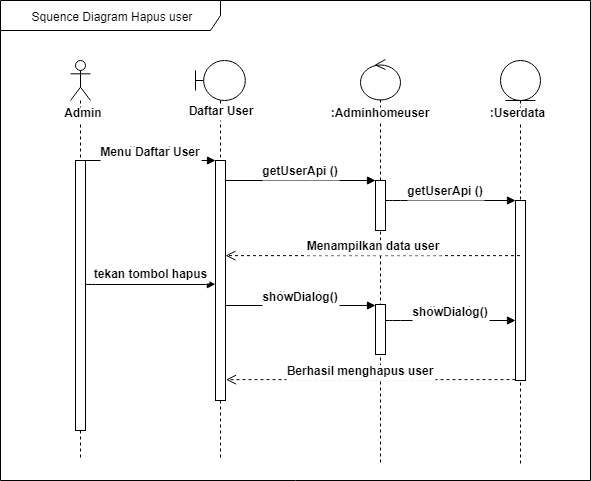
##### Gambar 5.36 *Sequence Diagram* “perbarui data sayur”

*Sequence Diagram* hapus sayurmerupakan interaksi antara objek untuk menghapusdata sayur. Berikut Gambar 5.37 merupakan *Squence diagram* hapus sayur :

****

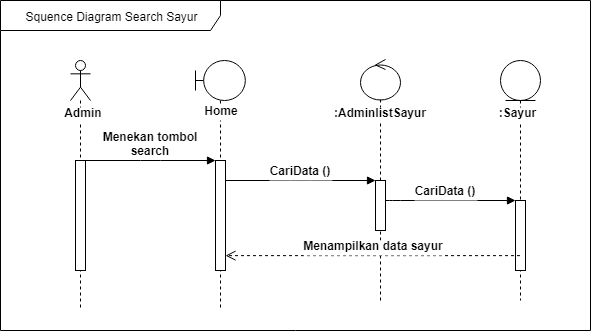
##### Gambar 5.37 *Sequence Diagram* “hapus sayur”

*Sequence Diagram* hapus *user* merupakan interaksi antara objek untuk menghapus *user* yang terdaftar. Berikut Gambar 5.38 merupakan *sequence diagram* hapus *user* :

****

##### Gambar 5.38 *Sequence Diagram* “hapus *user*”

*Sequence Diagram* search sayurmerupakan interaksi anatara object untuk mencari data sayur berdasarkan namasayur. Berikut Gambar 5.39 merupakan *sequence diagram* *search* sayur :



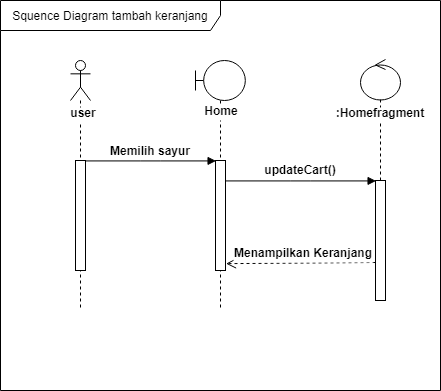
##### Gambar 5.39 *Sequence Diagram* “*search* sayur”

*Sequence Diagram* search *user* merupakan interaksi antara objek untuk mencarai data *user* berdasarkan nama *user*. Berikut Gambar 5.40 merupakan *sequence diagram* *search user* :



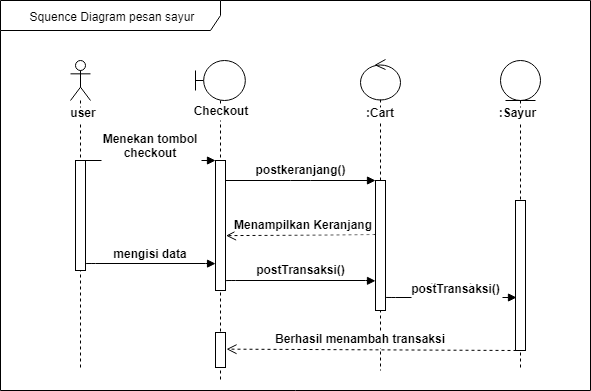
##### Gambar 5.40 *Sequence Diagram* “*search user*”

*Sequence Diagram* keranjang belanja merupakan interaksi antara objek untuk menyimpan data sayur ke dalam kerajang belanja/*cart*. Berikut Gambar 5.41 merupakan *sequence diagram* keranjang belanja :



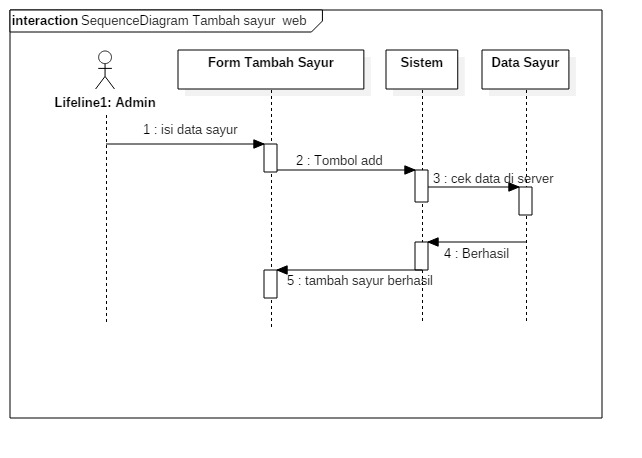
##### Gambar 5.41 *Sequence Diagram* “keranjang belanja”

*Sequence Diagram* pesan sayurmerupakan interaksi antara objek untuk memesan sayur yang telah dijual di sayur online. Berikut Gambar 5.42 merupakan *sequence diagram* pesan sayur :

****

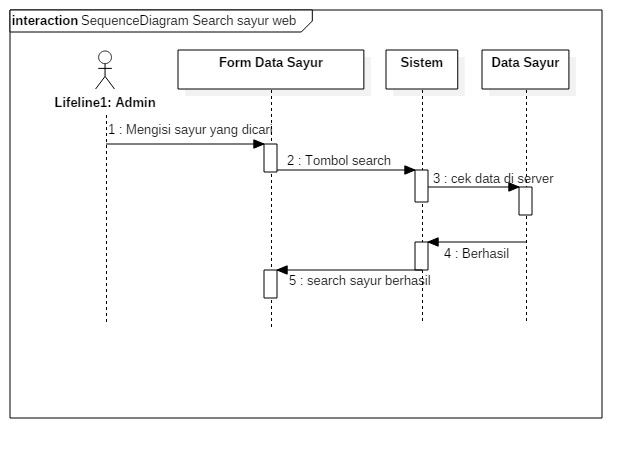
##### Gambar 5.42 *Sequence Diagram* “pesan sayur”

*Sequence Diagram* Tambah sayur web merupakan interaksi antara objek untuk menambah sayur digudang. Berikut Gambar 5.43 merupakan *sequence diagram* tambah sayur *web* :



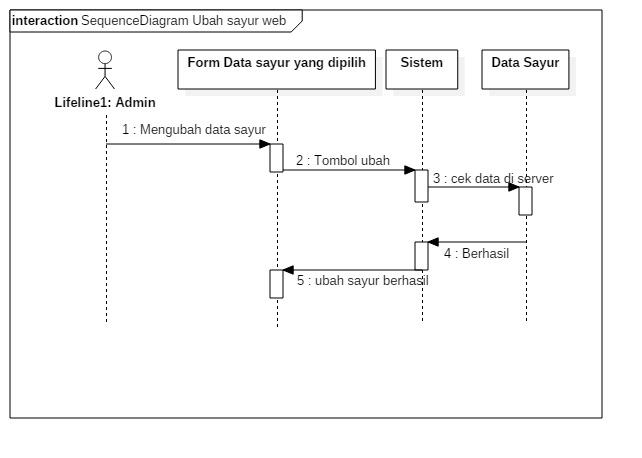
##### Gambar 5.43 *Sequence Diagram* “Tambah sayur web”

*Sequence Diagram Search* sayur web merupakan interaksi antara objek untuk mencari sayur digudang. Berikut Gambar 5.44 merupakan *sequence diagram* *search* sayur *web* :



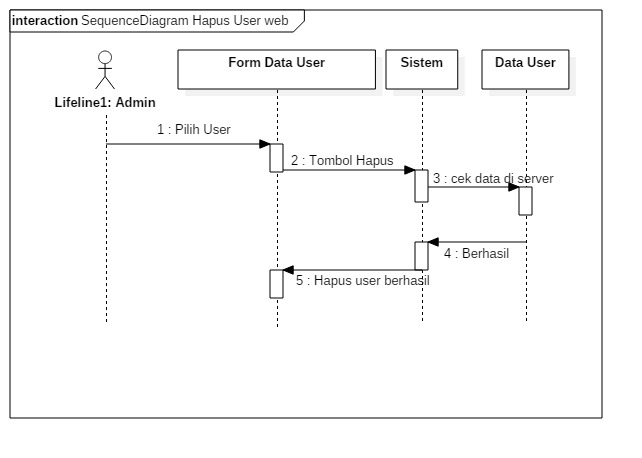
##### Gambar 5.44 *Sequence Diagram* “*Search* sayur web”

*Sequence Diagram* Ubah sayur web merupakan interaksi antara objek untuk mengubah sayur digudang. Berikut Gambar 5.45 merupakan *sequence diagram* ubah sayur web :



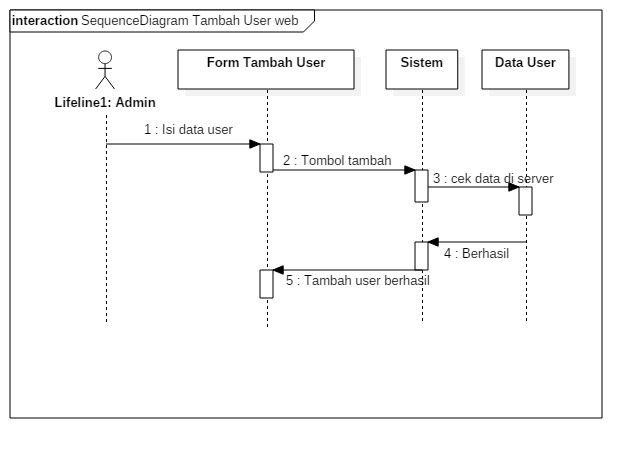
##### Gambar 5.45 *Sequence Diagram* “ubah sayur web”

*Sequence Diagram* Hapus *user* web merupakan interaksi antara objek untuk menghapus *user*. Berikut Gambar 5.46 merupakan *sequence diagram* hapus *user* web :



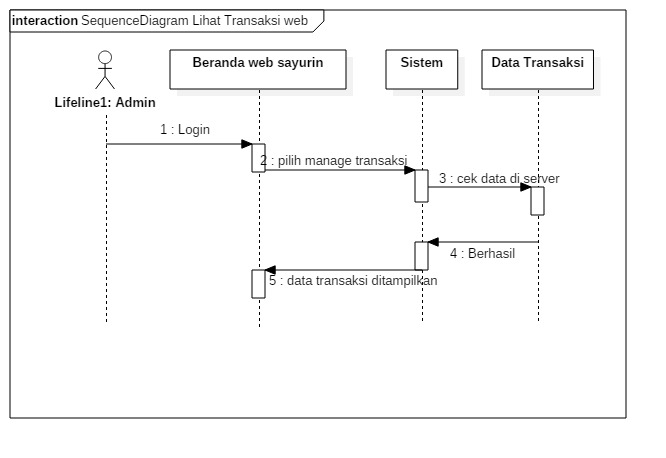
##### Gambar 5.46 *Sequence Diagram* “hapus *user* web”

*Sequence Diagram* Tambah *user* web merupakan interaksi antara objek untuk menambah *user*. Berikut Gambar 5.47 merupakan *sequence diagram* Tambah *user* web :



##### Gambar 5.47 *Sequence Diagram* “tambah *user* web”

*Sequence Diagram* Lihat Transaksi web merupakan interaksi antara objek untuk melihat transaksi yang dilakukan oleh *user*. Berikut Gambar 5.48 merupakan *sequence diagram* lihat transaksi web :



##### Gambar 5.48 *Sequence Diagram* “lihat transaksi web”

## 5.3 **Perancangan Perangkat Lunak**

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

### 5.3.1 Perancangan Algoritma

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

#### 5.3.1.1 Diagram Alir Proses Login

Diagram alir proses login sebagai *user* ini merupakan gambaran secara visual dari algoritma login, Admin melakukan *input username* dan *password*. Diagram alir proses tersebut ditunjukan seperti pada Gambar 5.22 berikut ini.



##### Gambar 5.49 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 proses tersebut ditunjukan seperti pada Gambar 5.23 berikut ini.



##### Gambar 5.50 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 proses tersebut ditunjukan seperti pada Gambar 5.24 berikut ini.



##### Gambar 5.51 Diagram Alir Proses Lihat Data Sayur

#### 5.3.1.4 Diagram Alir Proses Perbarui Data Sayur

Diagram alir proses perbaru data sayur ini merupakan gambaran secara visual dari algoritma proses perbarui data sayur, admin melakukan perbarui data sayur pada form menejemen sayur. Diagram alir proses tersebut ditunjukan seperti pada Gambar 5.25 berikut ini.



##### Gambar 5.52 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 form menejemen sayur. Diagram alir proses tersebut ditunjukan seperti pada Gambar 5.26 berikut ini.



##### Gambar 5.53 Diagram Alir Proses Hapus Data Sayur

### 5.3.2 Perancangan Antarmuka

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

### 5.3.3.1 Perancangan Tampilan Halaman *Login*

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



##### Gambar 5.54 Perancangan Tampilan Halaman *Login*

### 5.3.3.2 Perancangan Tampilan Halaman *Register*

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

**

##### Gambar 5.55 Perancangan Tampilan Halaman *Register*

### 5.3.3.3 Perancangan Tampilan Halaman *Home User*

Tampilan halaman *Home User*  berisi tampilan yang muncul ketika *user* berhasil login. *User* dapat memilih dan melihat sayur sesuai yang dijaul oleh admin, di halaman home *user* memiliki fitur *bottom sheet layout* berfungsi untuk menyimpan sayur yang akan dibeli dan menentukan jumlah sayur yang akan dibeli. Terdapat Button add digunakan untuk menyimpan sayur yang akan dibeli ke bottom sheet layout, dan button checkout berfungsi sebagai menyimpan sayur yang akan dibeli ke database dan berpindah halaman ke keranjang belanja untuk melengkapi data pengiriman. Yang terakhir terdapat tombol *search* yang berfungsi untuk mencari data sayur secara spesifik. Pada Gambar 5.29 berikut ini menunjukkan halaman *Home User*



##### Gambar 5.56 Perancangan Tampilan Halaman *Home User*

### 5.3.3.4 Perancangan Tampilan Halaman Keranjang Belanja

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

**

##### Gambar 5.57 Perancangan Tampilan Halaman Keranjang Belanja

### 5.3.3.5 Perancangan Tampilan Halaman Transaksi

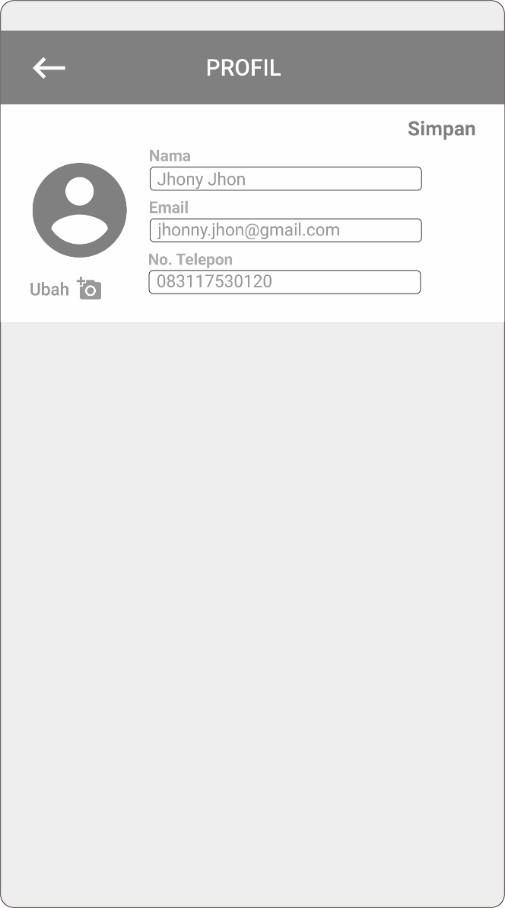
Tampilan halaman Transaksi belanja berisi tampilan yang muncul ketika *user* menekan tombol order. Halaman ini akan menampilkan data sayur, total sayur yang sebelumnya telah di order,jika barang sudah terkirim makan data akan berpindah ke tab *completed*. Pada Gambar 5.31 berikut ini menunjukkan halaman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

**

##### Gambar 5.58 Perancangan Tampilan Halaman Transaksi

### 5.3.3.6 Perancangan Tampilan Halaman Profil

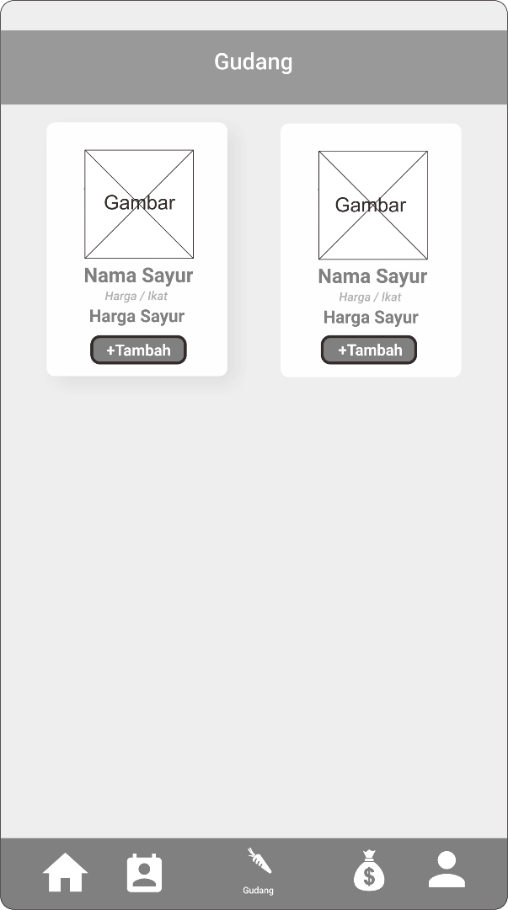
Tampilan halamanProfil muncul ketika *user* menekan tombol yang ada dipojok kanan bawah yaitu tombol *profile*. Halaman ini akan menampilkan data *user* yang yang berhasil login. Dihalaman ini *user* dapat merubah nama , *email* , no telepon dan foto. Pada Gambar 5.32 berikut ini menunjukkan halamanProfil.

**

##### Gambar 5.59 Perancangan Tampilan Halaman Profil

### 5.3.3.7 Perancangan Tampilan Gudang Sayur

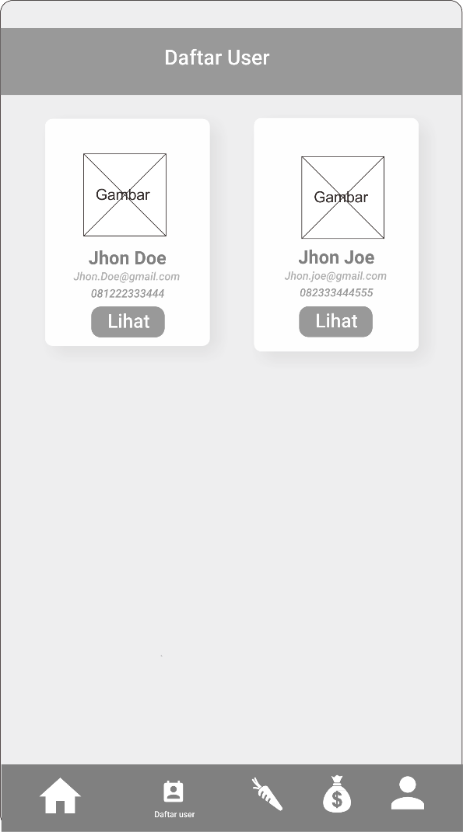
Tampilan halamangudang sayur adalah menu ketika admin berhasi login. Halaman ini akan menampilkan data sayur yang ada digudang. Dihalaman ini terdapat tombol *add* yang berfungsi untuk menambahkan sayur yang ingin dijual dari gudang ,dan diatas juga terdapat tombol search untuk mencari sayur di gudang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

******

##### Gambar 5.60 Perancangan Tampilan Halaman Gudang Sayur

### 5.3.3.8 Perancangan Tampilan *Manage User*

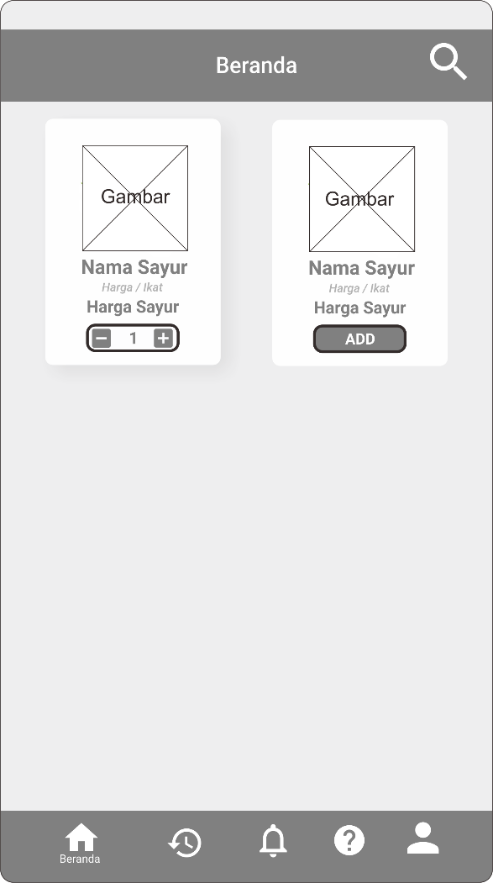
Tampilan halaman *Manage* *User* adalah menu ketika admin berhasi *login*. Halaman ini akan menampilkan data *user* yang telah mendaftar. Dihalaman ini terdapat tombol lihat yang berfungsi untuk *memanage* *user*, dan diatas juga terdapat tombol search untuk mencari *user* dengan spesifik. Pada Gambar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



##### Gambar 5.61 Perancangan Tampilan Halaman *Manage* *User*

### 5.3.3.9 Perancangan Tampilan *Manage* Sayur

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



##### Gambar 5.62 Perancangan Tampilan Halaman Manage Sayur

### 5.3.3.10 Perancangan Tampilan *Manage* Transaksi

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



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

### 5.3.3.11 Perancangan Tampilan *Login* Web

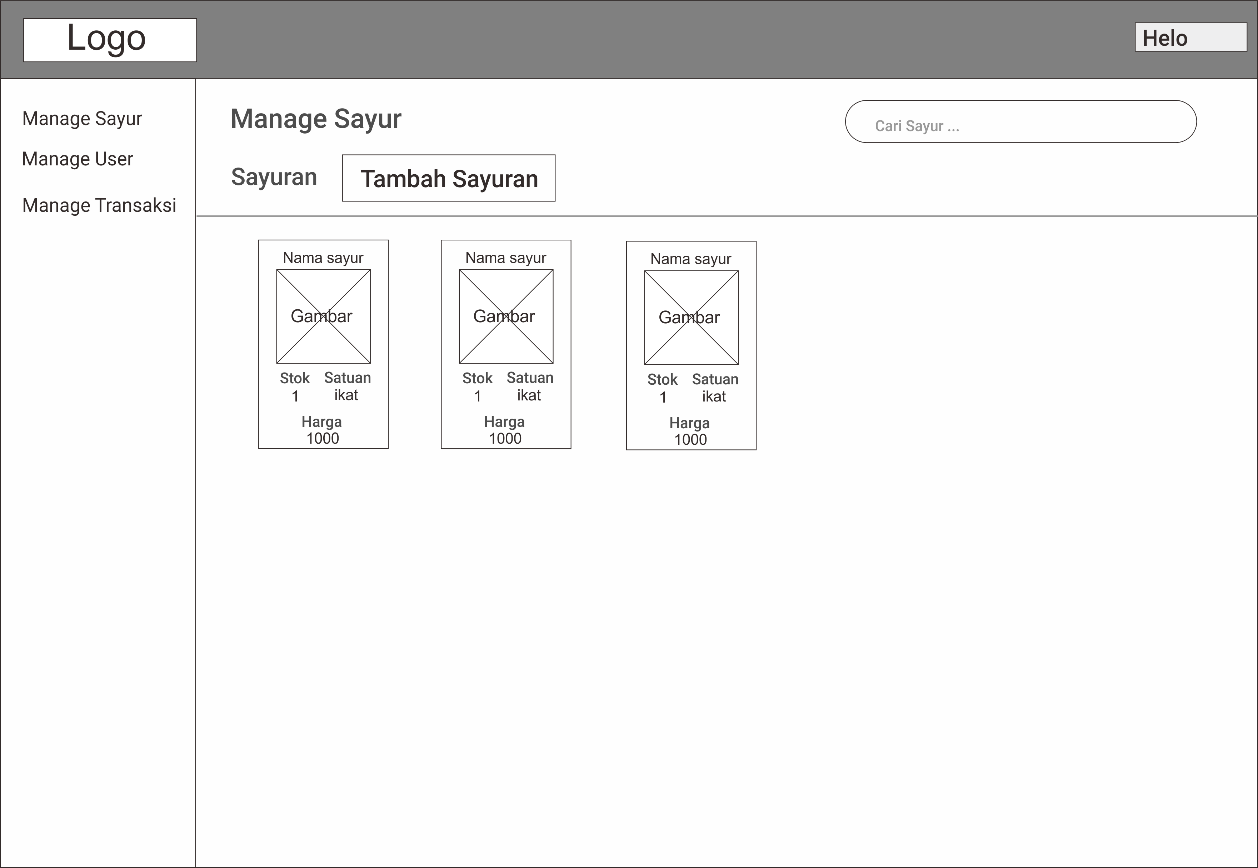
Tampilan halaman login web berisi logo, judul sistem, *field username* dan *password* serta tombol *Log In* yang digunakan *admin* untuk masuk ke dalam sistem setelah menginputkan *username* dan *password.* Pada Gambar 5.37 berikut ini menunjukkan halaman login Web*.*

##### web-login

##### Gambar 5.64 Perancangan Tampilan Halaman *Login* Web

### 5.3.3.12 Perancangan Tampilan *Manage* SayurWeb

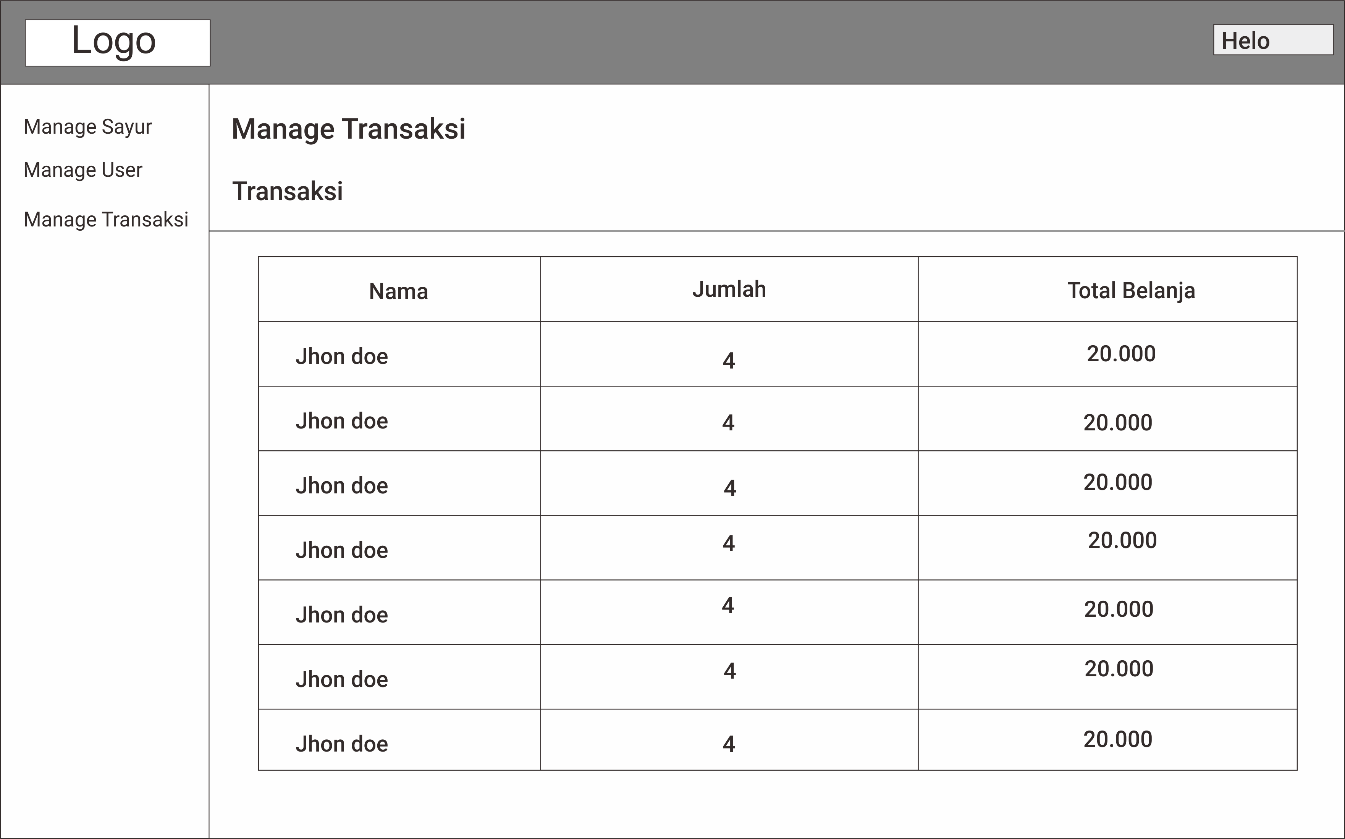
Tampilan halaman *Manage* Sayur web adalah menu ketika admin berhasi login. Halaman ini akan menampilkan data sayur yang telah dijual. Dihalaman ini admin dapat mengedit, menambah dan menghapus data sayur, dan disamping kanan juga terdapat tombol *search* untuk mencari sayur yang dijual dengan spesifik.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



##### Gambar 5.65 Perancangan Tampilan Halaman *Manage* Sayur Web

### 5.3.3.13 Perancangan Tampilan *Manage* TransaksiWeb

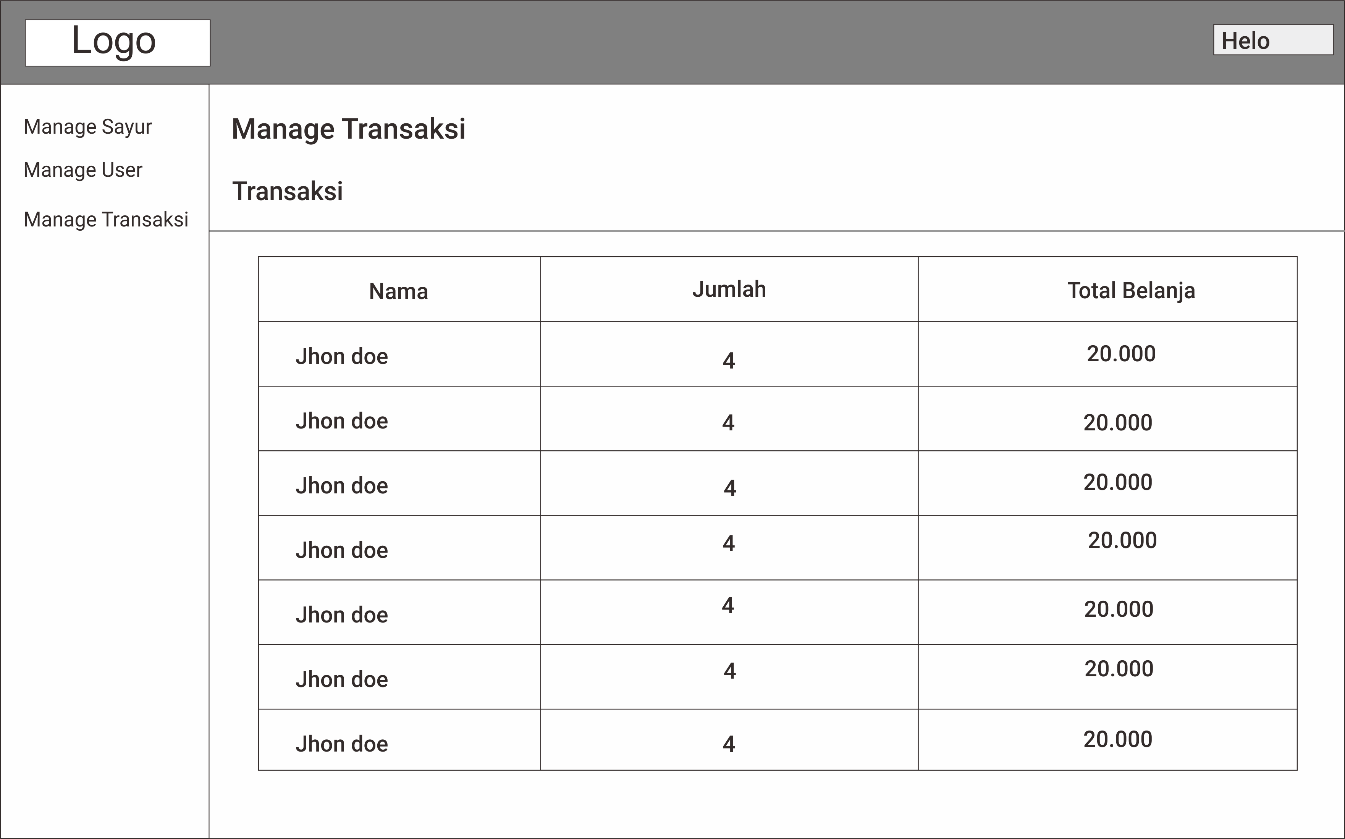
Tampilan halaman *Manage* Transaksi web adalah menu ketika admin berhasi login. Halaman ini akan menampilkan data tranasaksi yang dilkaukan oleh *user*. Dihalaman ini admin dapat mengedit dan menghapus data transaksi. Pada Gambar 5.39 berikut ini menunjukkan halaman *Manage* Transakasi Web.



##### Gambar 5.66 Perancangan Tampilan Halaman *Manage* Transakasi Web

### 5.3.3.14 Perancangan Tampilan *Manage User* Web

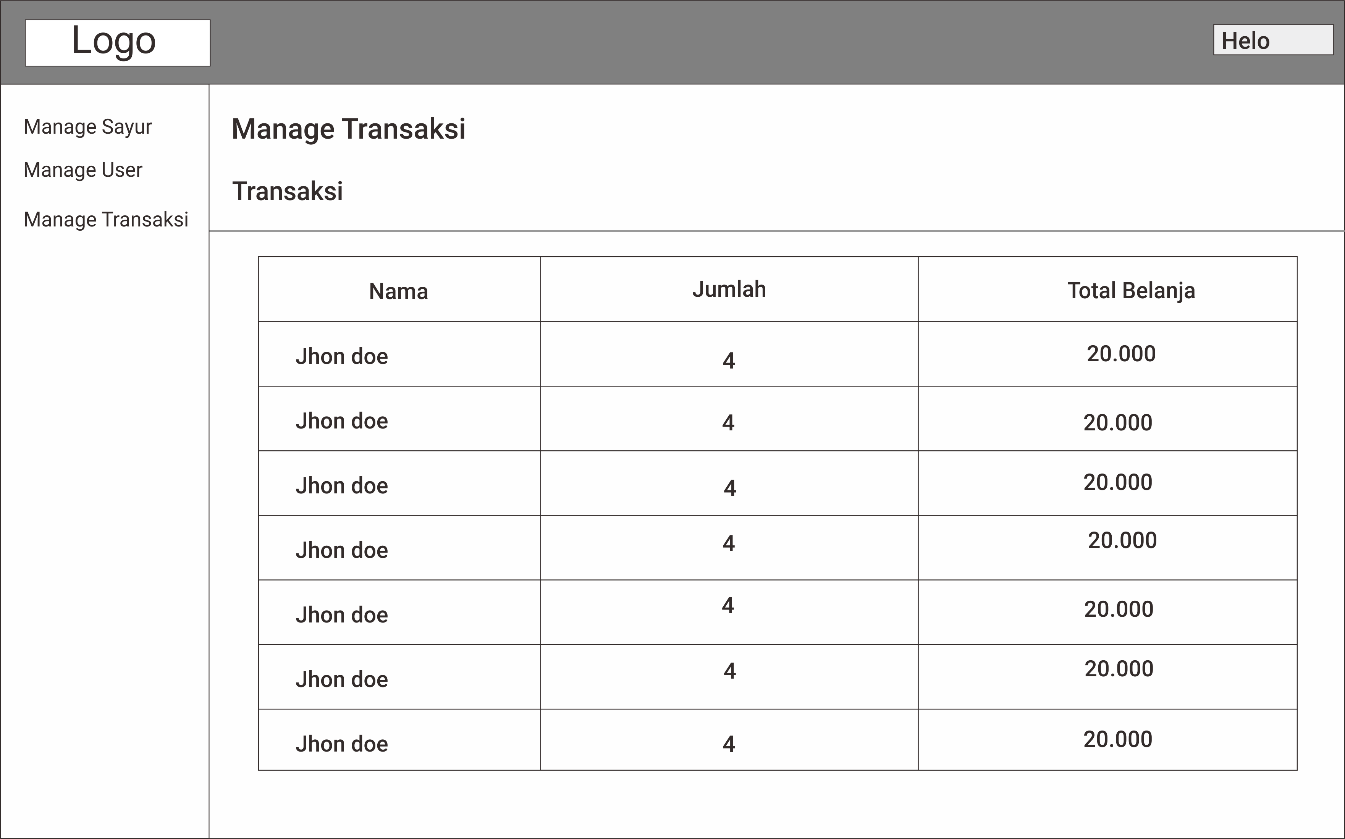
Tampilan halaman *Manage* *User* web adalah menu ketika admin berhasi login. Halaman ini akan menampilkan data *User* yang terdaftar. Dihalaman ini admin dapat menghapus data *user*, dan disamping kanan juga terdapat tombol search untuk mencari *User* yang terdaftar. Pada Gambar 5.40 berikut ini menunjukkan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



##### Gambar 5.67 Perancangan Tampilan Halaman *Manage* *User* Web

### 5.3.3.14 Perancangan Tampilan *Manage User* Web

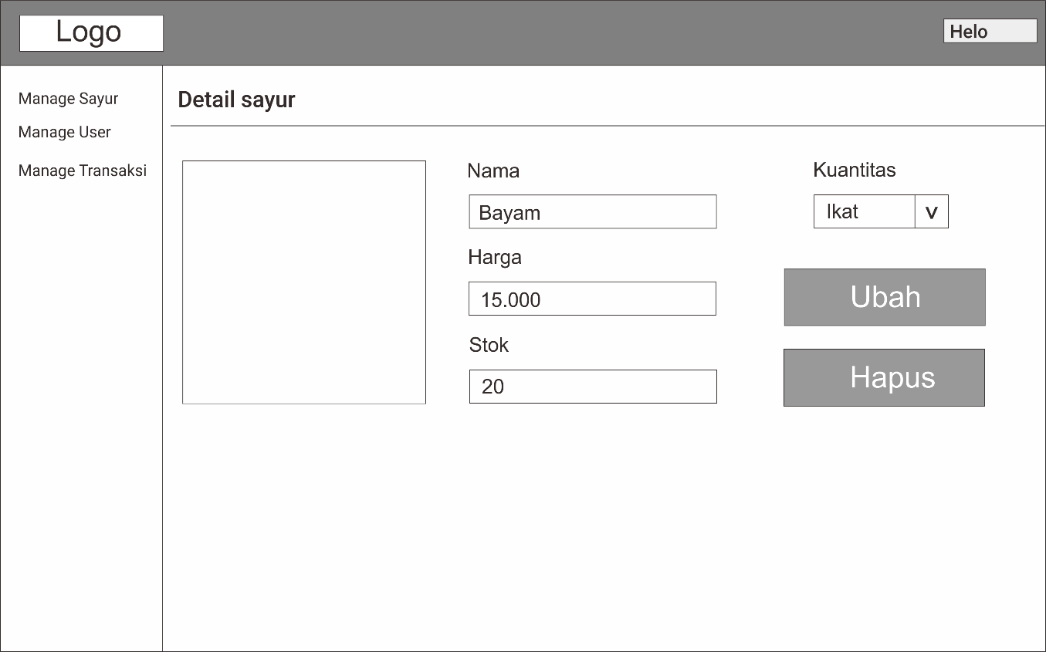
Tampilan halaman *Manage* *User* web adalah menu ketika admin berhasi *login*. Halaman ini akan menampilkan data *User* yang terdaftar. Dihalaman ini admin dapat menghapus data *user*, dan disamping kanan juga terdapat tombol search untuk mencari *User* yang terdaftar. Pada Gambar 5.40 berikut ini menunjukkan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



##### Gambar 5.68 Perancangan Tampilan Halaman *Manage* *User* Web

### 5.3.3.15 Perancangan Tampilan Detail SayurWeb

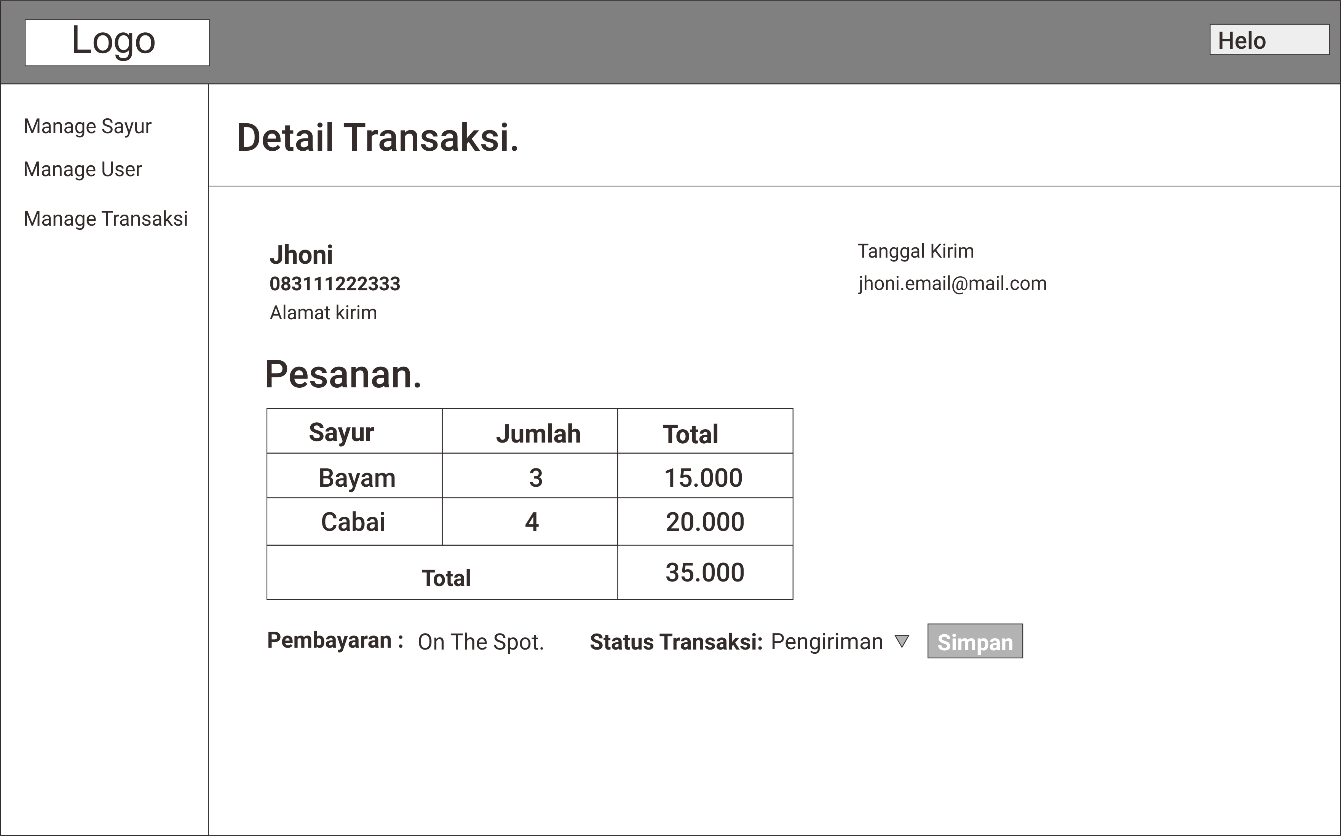
Tampilan halamandetailsayur web adalah menu ketika admin menekan sayur yang dipilih. Halaman ini akan menampilkan data sayur yang terdaftar. Dihalaman ini admin melihat detail sayur. Pada Gambar 5.41 berikut ini menunjukkan halamanDetail Sayur Web.



##### Gambar 5.69 Perancangan Tampilan Halaman Detail Sayur Web

### 5.3.3.16 Perancangan Tampilan Detail TransaksiWeb

Tampilan halamandetailTransaksi web adalah menu ketika admin menekan transaksi yang dipilih. Halaman ini akan menampilkan data transaksi yang dilakukan oleh *user*. Dihalaman ini admin melihat dan mengedit transaksi. Pada Gambar 5.42 berikut ini menunjukkan halamanDetail Transaksi Web.



##### Gambar 5.70 Perancangan Tampilan Halaman Detail Transaski Web