



**PENERAPAN METODE GAMIFIKASI PADA E-COMMERCE  
BERBASIS MOBILE APPS (STUDI KASUS USAHA MIKRO,  
KECIL DAN MENENGAH (UMKM) MACARINA JEMBER)**

**SKRIPSI**

Oleh

**Moh. Basirudin**

**NIM 152410101076**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2019**



**PENERAPAN METODE GAMIFIKASI PADA E-COMMERCE  
BERBASIS MOBILE APPS (STUDI KASUS USAHA MIKRO  
KECIL DAN MENENGAH (UMKM) MACARINA JEMBER)**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mencapai gelar Sarjana Komputer

Oleh

**Moh. Basirudin**

**NIM 152410101076**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2019**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ibunda Halimah dan Ayahanda Muhadi;
3. Saudara-saudaraku Ahmad Baedowi dan M. Shofi;
4. Keluarga besar;
5. Sahabat-sahabatku dengan dukungan dan doanya;
6. Guru-guruku baik dari pendidikan formal maupun informal;
7. Almamater Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

## MOTTO

*“Jangan menunda. Sebab satu hari kamu menunda, berarti akan ada satu hari esokmu juga yang akan tertunda”*

**- Moh. Basirudin –**

*“Berbagilah ketika kamu bisa melakukannya. Kamu juga pernah berada pada posisi ketidaktahuan yang membuatmu merasakan sakitnya belajar sendiri, karna sulitnya mencari bantuan ke teman-temanmu yang kamu anggap bisa.”*

**- Moh. Basirudin –**

*“Tidak ada balasan untuk kebaikan selain kebaikan (pula).”*

**-QS. Ar-Rahman 55:60 -**



## PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Moh. Basirudin

NIM : 152410101076

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan Metode Gamifikasi pada *E-Commerce* Berbasis *Mobile Apps* (Studi Kasus Usaha Mikro Kecil dan Menengah (UMKM) Macarina Jember)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 08 April 2019

Yang menyatakan,

Moh. Basirudin

NIM 152410101076

**SKRIPSI**

**PENERAPAN METODE GAMIFIKASI PADA E-COMMERCE  
BERBASIS MOBILE APPS (STUDI KASUS USAHA MIKRO  
KECIL DAN MENENGAH (UMKM) MACARINA JEMBER)**

Oleh

**Moh. Basirudin**

**NIM 152410101076**

Pembimbing

Dosen Pembimbing Utama : Yanuar Nurdiansyah, ST., M.Cs.

Dosen Pembimbing Pendamping : Priza Pandunata S.Kom., M.Sc.

## PENGESAHAN PEMBIMBING

Skripsi yang berjudul “Penerapan Metode Gamifikasi pada *E-Commerce* Berbasis *Mobile Apps* (Studi Kasus Usaha Mikro, Kecil dan Menengah (UMKM) Macarina Jember)”, telah diuji dan disahkan pada:

hari, tanggal : Senin, 27 Mei 2019

tempat : Fakultas Ilmu Komputer Universitas Jember.

Disetujui oleh:

Pembimbing I

Pembimbing II

Yanuar Nurdiansyah, ST., M.Cs.  
198201012010121004

Priza Pandunata, S.Kom., M.Sc.  
198301312015041001

## PENGESAHAN PENGUJI

Skripsi yang berjudul “Penerapan Metode Gamifikasi pada *E-Commerce* Berbasis *Mobile Apps* (Studi Kasus Usaha Mikro, Kecil dan Menengah (UMKM) Macarina Jember)”, telah diuji dan disahkan pada:

hari, tanggal : Senin, 27 Mei 2019

tempat : Fakultas Ilmu Komputer Universitas Jember 2019

Disetujui oleh:

Penguji I

Penguji II

Nelly Oktavia A, S.Si., MT  
198410242009122008

Tio Dharmawan, S.Kom., M.Kom  
760016851

Mengesahkan  
Dekan Fakultas Ilmu Komputer

Prof. Dr. Saiful Bukhori, ST., M.Kom  
19681113199421001

## RINGKASAN

### **Penerapan Metode Gamifikasi pada *E-Commerce* Berbasis *Mobile Apps* (Studi Kasus Usaha Mikro Kecil dan Menengah (UMKM) Macarina Jember);**

Moh. Basirudin, 152410101076; 2019, 264 halaman; Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

Macarina atau singkatan dari Macaroninya Ina merupakan salah satu dari banyaknya Usaha Mikro, Kecil dan Menengah (UMKM) yang berada di Jember dan mengandalkan inovasi kreatif dari bahan macaroni sebagai produk andalan mereka. Seiring dengan berjalannya waktu, banyak UMKM serupa yang mulai bermunculan, sehingga membuat persaingan mejadi semakin ketat. Dan untuk mengatasi hal tersebut maka dibutuhkan sebuah inovasi yang unik, dimana inovasi tersebut mampu menjadi pembeda antara UMKM Macarina dengan yang lainnya.

Seiring dengan berjalannya waktu, perkembangan Teknologi Informasi dan Komunikasi (TIK) semakin pesat dan dikenal oleh masyarakat. Perkembangan ini ditunjukkan oleh data yang dirilis *eMarketer.com* pada tahun 2015, menunjukkan perkiraan pengguna *smartphone* pada tahun 2016 berkisar 65,5 juta pengguna dan meningkat menjadi 74,9 juga pengguna di tahun 2017 dan menjadi 84,5 juta pengguna di tahun 2018 dan terus meningkat hingga menjadi 92 juta pengguna pada tahun 2019. Hal tersebut menunjukkan bahwa perkembangan penggunaan *smartphone* di Indonesia mengalami peningkatan yang sangat drastis. Dari hal tersebut didapatkan solusi untuk mengatasi permasalahan yang dihadapi oleh Macarina, yakni dengan pembuatan aplikasi yang didalamnya dimasukkan unsur-unsur gamifikasi yang akan menjadi pembeda dengan UMKM yang lainnya. Karena kebanyakan dari mereka masih menggunakan *e-marketplace* untuk melakukan transaksinya.

Aplikasi *e-commerce* berbasis *mobile apps* yang menerapkan metode gamifikasi untuk Macarina ini merupakan aplikasi untuk mempermudah *customer* yang hendak membeli produk di outlet-outlet Macarina. Karena dengan adanya

aplikasi ini, *customer* tidak perlu repot-repot untuk mengecek outlet mana saja yang masih buka dan mengetahui rasa mana yang masih tersedia di outlet tersebut.

Hasil dari penelitian ini adalah *customer* dapat memesan secara langsung produk Macarina, melakukan penukaran *M-Poin* yang sudah didapatkan dari pembelian produk menggunakan aplikasi, mengetahui rasa yang paling diminati dan juga *leaderboard customer* untuk mendapatkan *reward* berupa *M-Cash* untuk yang urutan paling atas pada tiap awal bulan. *M-Cash* merupakan uang digital yang hanya bisa digunakan untuk melakukan transaksi di UMKM Macarina. *M-Poin* merupakan poin yang didapatkan dari pembelian produk di Macarina menggunakan aplikasi. Semakin sering melakukan transaksi menggunakan aplikasi, maka akan semakin besar pula kesempatan untuk dapat menukarkannya dengan hadiah yang sudah disiapkan oleh pihak Macarina.



## PRAKATA

Puji syukur kehadiran Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Penerapan Metode Gamifikasi pada *E-Commerce* Berbasis *Mobile Apps* (Studi Kasus UMKM Macarina Jember)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember:

Penyusunan skripsi ini tidak lepas dari dukungan berbagai belah pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

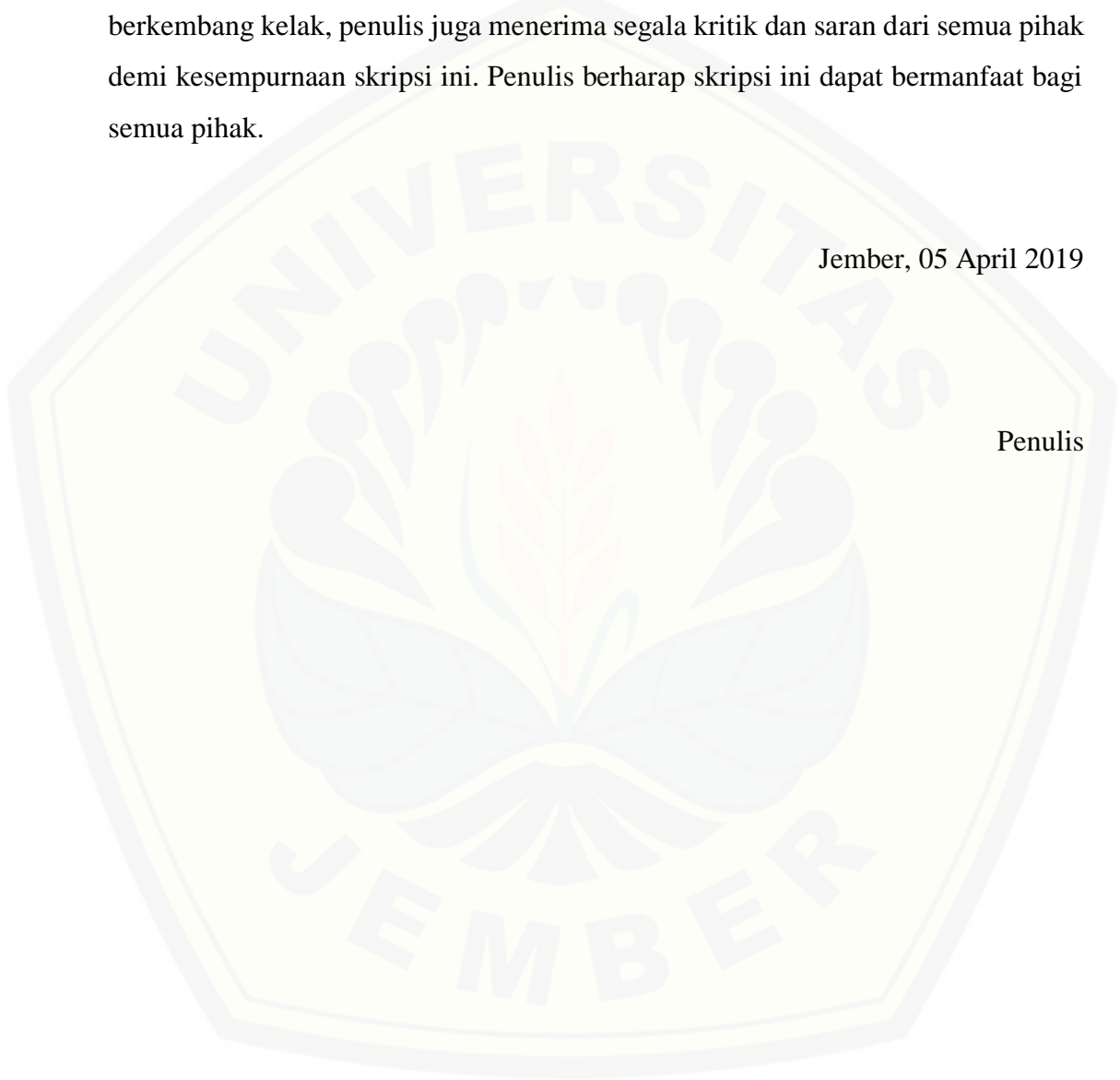
1. Dr. Saiful Bukhori, ST., M.Kom selaku dekan Fakultas Ilmu Komputer;
2. Yanuar Nurdiansyah, ST., M.Cs. selaku dosen pembimbing utama dan Priza Pandunata S.Kom., M.Sc. selaku dosen pembimbing pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
3. Fahrobyy Adnan, S.Kom., MMSI. selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Fakultas Ilmu Komputer Universitas Jember;
5. Ibunda Halimah dan Ayahanda Muhadi yang selalu mendukung serta mendoakan dan menjadi orang tua yang luar biasa;
6. Saudaraku Ahmad Baedowi dan M. Shofi yang telah memberikan dukungan semangat;
7. Keluarga besar yang telah mendukung dan mendoakan;
8. Teman-teman kelas B SELECTION;
9. Keluarga SELECTION angkatan 2015 atas motivasi yang selalu mengalir;
10. Teman-teman dari Tim ODS (Vian Elvada, Lutfi Ika P., Bratasena A., Aleq insan);

11. Teman-teman Fakultas Ilmu Komputer di semua angkatan atas bantuan dan dukungannya;
12. Semua pihak yang tidak bisa disebutkan satu persatu.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 05 April 2019

Penulis





## DAFTAR ISI

SKRIPSI.....	i
SKRIPSI.....	ii
PERSEMBAHAN.....	iii
MOTTO.....	iv
PERNYATAAN.....	v
SKRIPSI.....	vi
PENGESAHAN PEMBIMBING.....	vii
PENGESAHAN PENGUJI.....	viii
RINGKASAN.....	ix
PRAKATA.....	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL.....	xvi
DAFTAR GAMBAR.....	xvii
BAB 1. PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	4
1.3. Tujuan Penelitian.....	4
1.4. Batasan Masalah.....	4
1.5. Sistematika Penulisan.....	5
BAB 2. TINJAUAN PUSTAKA.....	6
2.1. Penelitian Terdahulu.....	6
2.2. <i>E-Commerce</i> .....	7
2.3. <i>Mobile Apps</i> .....	7

2.4.	<i>Android</i> .....	8
2.5.	Gamifikasi .....	9
BAB 3.	METODOLOGI PENELITIAN .....	12
3.1.	Tempat Penelitian .....	12
3.2.	Waktu Penelitian .....	12
3.3.	Jenis Penelitian .....	12
3.4.	Tahapan Penelitian .....	12
3.4.1.	Studi Literatur .....	12
3.4.2.	Pengumpulan Data .....	13
3.4.3.	Pemilihan Elemen Gamifikasi.....	13
3.4.4.	Pengembangan Perangkat Lunak .....	13
BAB 4.	RANCANGAN SISTEM.....	16
4.1.	Deskripsi Umum Sistem.....	16
4.1.1.	Pembelian Produk Macarina .....	16
4.1.2.	Penukaran <i>Merchandise</i> Menggunakan <i>M-Poin</i> .....	17
4.1.3.	Mendapatkan <i>M-Poin</i> .....	17
4.2.	Analisa Kebutuhan .....	18
4.2.1.	Studi Literatur .....	19
4.2.2.	Kebutuhan Fungsional .....	19
4.2.3.	Kebutuhan Non-Fungsional .....	21
4.3.	Desain Sistem .....	21
4.3.1.	<i>Input and Output Diagram</i> .....	21
4.3.2.	<i>Use case Diagram</i> .....	22
4.3.3.	<i>Scenario</i> .....	24
4.3.4.	<i>Activity diagram</i> .....	33

4.3.5.	<i>Sequence Diagram</i> .....	40
4.3.6.	<i>Class Diagram</i> .....	48
4.3.7.	<i>Database Structure</i> .....	50
4.4.	Pengujian Sistem .....	54
4.4.1.	Pengujian <i>Black Box</i> .....	55
BAB 5.	HASIL DAN PEMBAHASAN .....	65
5.1.	Pemesanan Produk dan Mendapatkan <i>M-Poin</i> .....	65
5.2.	Penukaran <i>Merchandise</i> atau hadiah Menggunakan <i>M-Poin</i> .....	70
5.3.	Perangkingan <i>Customer</i> dan Perangkingan Rasa Favorit .....	73
5.4.	Visualisasi.....	76
BAB 6.	PENUTUP .....	85
6.1.	Kesimpulan.....	85
6.2.	Saran .....	86
DAFTAR PUSTAKA	.....	87
LAMPIRAN	.....	89
A.	<i>Scenario</i> .....	89
B.	<i>Sequence diagram</i> .....	133
C.	<i>Activity diagram</i> .....	171
D.	Pengujian <i>Black Box</i> .....	187
E.	Kode Program .....	231
F.	Tampilan Aplikasi.....	257

## DAFTAR TABEL

Tabel 2.1 Daftar Versi Kode Sistem Operasi Android .....	8
Tabel 4.1 Melihat Leaderboard .....	25
Tabel 4.2 Melakukan Penukaran Hadiah .....	27
Tabel 4.3 Melakukan Pemesanan.....	28
Tabel 4.4 Verifikasi Pesanan.....	30
Tabel 4.5 Melihat Leaderboard.....	56
Tabel 4.6 Melakukan Penukaran Hadiah .....	58
Tabel 4.7 Melakukan Pemesanan.....	59
Tabel 4.8 Verifikasi Pesanan.....	61

## DAFTAR GAMBAR

Gambar 1.1 Peningkatan Pengguna Smartphone di Indonesia pada Tahun 2016 - 2019.....	1
Gambar 1.2 Penggunaan Smartphone dalam Keseharian .....	2
Gambar 3.1 Tahapan Penelitian .....	12
Gambar 3.2 Software Development Life Cycle (SDLC) Waterfall.....	13
Gambar 4.1 Flowchart Pembelian Produk Macarina .....	16
Gambar 4.2 Flowchart Penukaran Merchandise Macarina Menggnakan M-Poin	17
Gambar 4.3 Flowchart Konfirmasi Pesanan Customer oleh Karyawan .....	18
Gambar 4.4 Input and Output Diagram.....	22
Gambar 4.5 Use case Diagram.....	23
Gambar 4.6 Activity diagram Melihat Leaderboard .....	35
Gambar 4.7 Activity diagram Melakukan Penukaran Hadiah .....	37
Gambar 4.8 Activity diagram Melakukan Pemesanan.....	38
Gambar 4.9 Activity diagram Verifikasi Pesanan.....	39
Gambar 4.10 Sequence Diagram Melihat Leaderboard.....	42
Gambar 4.11 Sequence Diagram Melakukan Penukaran Hadiah .....	44
Gambar 4.12 Sequence Diagram Melakukan Pemesanan.....	46
Gambar 4.13 Sequence Diagram Verifikasi Pesanan .....	47
Gambar 4.14 Class Diagram .....	49
Gambar 5.1 Penempatan Elemen Gamifikasi Point pada Metode Pemabayaran Pemesana Produk Menggunakan Aplikasi Macarina.....	65
Gambar 5.2 Penempatan Elemen Gamifikasi Reward pada Pemesanan Produk..	66
Gambar 5.3. Coding Membangun Fitur Pemesanan Produk.....	67
Gambar 5.4 Alur Pemesanan Produk Macarina dan Mendapatkan M-Poin .....	69
Gambar 5.5 Penerapan Elemen Gamifikasi Points pada Penukaran Merchandise Menggunakan M-Poin.....	71
Gambar 5.6 Coding Membangun Fitur Penukaran Hadiah.....	72
Gambar 5.7 Penukaran Merchandise Menggunakan M-Poin .....	73

Gambar 5.8 Penempatan Elemen Gamifikasi Leaderboard pada Fitur Perangkingan Customer .....	74
Gambar 5.9 Penempatan Elemen Gamifikasi Reward pada Fitur Perangkingan Customer .....	75
Gambar 5.10 Coding Membangun Fitur Perangkingan Customer.....	76
Gambar 5.11 Tampilan Dialog Pemesanan Macarina.....	78
Gambar 5.12 Tampilan Halaman Leaderboard Customer .....	79
Gambar 5.13 Tampilan Halaman Rasa Favorit.....	80
Gambar 5.14 Tampilan Dialog Verifikasi Pesanan.....	81
Gambar 5.15 Tampilan Halaman Leaderboard .....	83
Gambar 5.16 Tampilan Dialog Ubah Reward Leaderboard .....	83
Gambar 5.17 Tampilan Dialog Kirim Reward Leaderboard .....	84

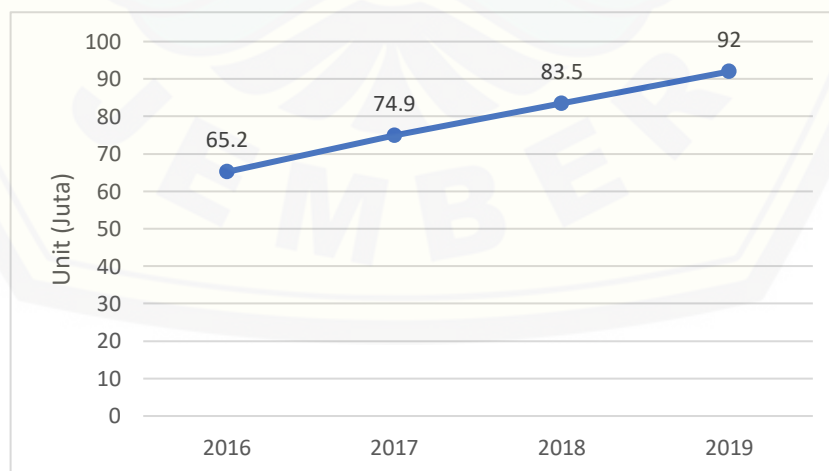


## BAB 1. PENDAHULUAN

### 1.1. Latar Belakang

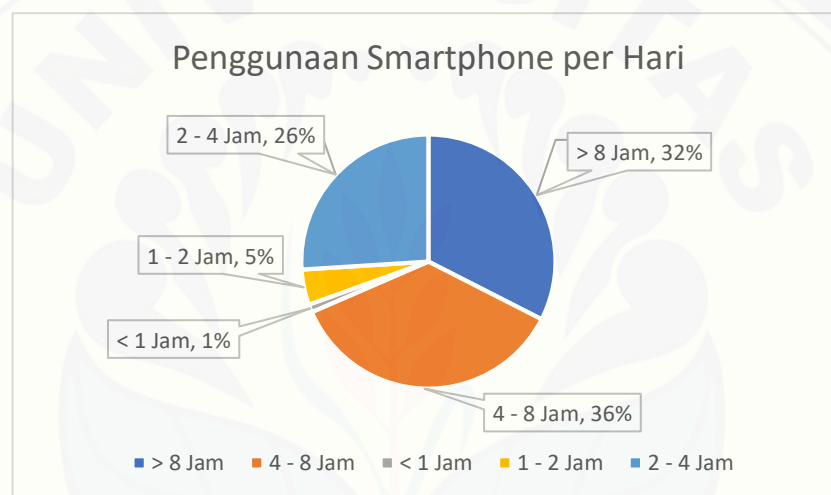
Macarina atau singkatan dari Macaroninya Ina merupakan salah satu dari banyaknya Usaha Mikro, Kecil dan Menengah (UMKM) yang berada di Jember dan mengandalkan inovasi kreatif dari bahan macaroni sebagai produk andalan mereka. Seiring dengan berjalannya waktu banyak UMKM serupa yang mulai bermunculan, sehingga membuat persaingan semakin ketat. Semakin ketatnya persaingan antar UMKM membuat Macarina harus merancang sebuah strategi atau membuat inovasi baru agar produknya tetap bisa diterima oleh para pelanggannya, yang nantinya inovasi tersebut mampu menjadi pembeda dengan UMKM lainnya.

Seiring dengan berjalannya waktu, perkembangan Teknologi Informasi dan Komunikasi (TIK) semakin pesat dan dikenal oleh masyarakat. Perangkat yang populer di kalangan masyarakat antara lain laptop, tablet, PDA dan yang populer saat ini adalah *smartphone*. Hal ini ditandai dengan jumlah pengguna *smartphone* di Indonesia yang terus meningkat dari tahun ke tahun. Berdasarkan data yang dirilis oleh eMarketer.com tahun 2015 diperkirakan pengguna *smartphone* di Indonesia pada tahun 2019 mencapai 92 juta pengguna (eMarketer, 2016). Data tersebut dapat dilihat pada Gambar 1.1.



Gambar 1.1 Peningkatan Pengguna Smartphone di Indonesia pada Tahun 2016 - 2019

Kemudahan dalam penggunaan dan ditunjang dengan koneksi yang sudah lebih cepat, membuat orang Indonesia sulit untuk lepas dari ponselnya. Berdasarkan penelitian yang dilakukan oleh *CLSA Indonesia, PT* pada tahun 2016 mengatakan bahwa sebanyak 36% pengguna *smartphone* di Indonesia menghabiskan waktu 4-8 jam dalam sehari memakai perangkat pintarnya dan juga menunjukkan bahwa lebih dari 90% pengguna *smartphone* memerlukan waktu tidak kurang dari 2 jam per hari dalam menggunakannya. Bahkan 32,5% pengguna *smartphone* ini membutuhkan lebih dari 8 jam untuk mengakses ponselnya (CLSA Indonesia, 2016). Data tersebut dapat dilihat pada Gambar 1.2.



Gambar 1.2 Penggunaan Smartphone dalam Keseharian

Perkembangan *smartphone* telah memberikan banyak manfaat bagi kehidupan sehari-hari, terutama bagi para pelaku bisnis. Misalkan saja dengan adanya penjualan secara *online* (*e-commerce*) yang merupakan sistem baru bagi para pelaku bisnis untuk memanfaatkan teknologi agar lebih efektif dan efisien dalam melakukan transaksi penjualan di dunia maya. *E-commerce* sendiri bukan hanya sekedar mekanisme menjual barang dan jasa, tapi merupakan transformasi bisnis yang mengubah cara perusahaan dalam melakukan aktifitasnya. Karena dengan adanya *e-commerce* dapat mempermudah konsumen untuk mengetahui detail dari suatu produk, baik barang ataupun jasa mulai dari harga hingga hal lain yang berkaitan dengan produk tersebut. Disamping itu layanan ini dapat diakses 24 jam *non-stop*. Hal ini dapat membantu Macarina dalam melebarkan sayap



distribusi. Banyaknya penggunaan *smartphone* berbanding lurus dengan bertambahnya calon *customer* yang menggunakan *smartphone*. Maka dari itu, cara yang paling efektif dalam menghadapi permasalahan Macarina saat ini adalah membuat aplikasi berbasis *mobile apps*.

Perkembangan *e-commerce* juga akan sangat membantu bagi para pemilik Usaha Mikro, Kecil dan Menengah (UMKM) dalam menghadapi persaingan dagang yang semakin kompleks. Sehingga *e-commerce* bisa dijadikan sebagai sarana untuk mengembangkan strategi yang efektif dan menciptakan inovasi baru. Inovasi dalam bertransaksi serta kemampuan untuk menarik dan mempertahankan pelanggan menjadi kunci dalam memenangkan persaingan. Sehingga membutuhkan inovasi yang berbeda dengan UMKM lainnya, semisal dengan diterapkannya gamifikasi pada aplikasi *e-commerce* tersebut. Gamifikasi merupakan penerapan dan penggunaan elemen-elemen *game* ke dalam konteks non-*game*, seperti elemen *points*, *badge*, *leaderboard*, *achievement*, dan lain sebagainya (Haryanto, 2017).

Hal yang menarik dari gamifikasi adalah caranya untuk dapat menarik perhatian pengguna. Metode ini membuka jalan untuk perusahaan dan pengguna agar dapat saling berinteraksi dengan cara baru. *GO-Points* dari GO-Jek adalah salah satu contohnya. *GO-Points* bekerja ketika pengguna menggunakan saldo *GO-Pay* mereka ketika menggunakan jasa GO-Jek. Dari setiap transaksi yang berhasil, pengguna akan mendapatkan token yang akan diundi untuk mendapatkan poin. Poin ini nantinya akan ditukar dengan berbagai penawaran menarik seperti diskon atau voucher.

Penerapan sistem poin pada UMKM Macarina di Jember ini memanfaatkan elemen *reward* pada bisnisnya. Semakin sering pengguna bertransaksi menggunakan aplikasi ini, maka akan semakin banyak pula poin yang akan diperoleh. Jadi, semakin besar pula kesempatan mereka untuk mendapatkan hadiah dari Macarina. Maka dari itu sangat cocok untuk menerapkan gamifikasi pada sebuah *e-commerce*, apalagi masyarakat sendiri sangat senang dengan adanya

*reward* dalam setiap aktifitasnya. Sehingga penerapan metode ini diharapkan dapat meningkatkan daya saing UMKM Macarina di Jember dengan UMKM lainnya.

### 1.2. Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang, terdapat beberapa permasalahan yang harus diselesaikan dalam penelitian ini, yaitu:

1. Bagaimana cara membuat aplikasi *e-commerce* berbasis *mobile apps*?
2. Bagaimana penerapan metode gamifikasi pada *e-commerce* berbasis *mobile apps*?

### 1.3. Tujuan Penelitian

Tujuan dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penulisan ini adalah:

1. Membangun aplikasi *e-commerce* berbasis *mobile apps* dengan penerapan metode gamifikasi untuk pemesanan produk di Macarina Jember.
2. Penerapan metode gamifikasi pada aplikasi *e-commerce* berbasis *mobile apps*.

### 1.4. Batasan Masalah

Agar tidak terjadi perluasan masalah pada penelitian ini, maka ditetapkan beberapa batasan permasalahan. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penerapan gamifikasi berbasis *mobile apps* dilakukan pada Usaha Mikro, Kecil dan Menengah (UMKM) Macarina Jember.
2. Penerapan gamifikasi berbasis *mobile apps*.
3. Diterapkan pada *smartphone* yang bersistem operasi *Android* minimal *Android 4.0.3 (Ice Cream Sandwich)*.
4. Aplikasi yang dibangun adalah aplikasi pemesanan.
5. Elemen gamifikasi yang digunakan adalah elemen *point*, *reward* dan *leaderboard*.

### 1.5. Sistematika Penulisan

Adapun sistematika penulisan dijelaskan sebagai berikut:

#### 1. Pendahuluan

Bab ini memuat uraian tentang latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan terkait penelitian yang dilakukan.

#### 2. Tinjauan Pustaka

Bab ini menjelaskan tentang kajian-kajian teori terkait dengan penelitian yang dapat mendukung penelitian yang dilakukan.

#### 3. Metodologi Penelitian

Bab ini menjelaskan tentang jenis penelitian, pengembangan sistem dan pengujian metode metode yang digunakan

#### 4. Rancangan Sistem

Bab ini menjelaskan tentang perancangan sistem yang dikembangkan. Perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, kemudian merancang *input and output diagram*, *use case diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *database structure*.

#### 5. Hasil dan Pembahasan

Bab ini menjelaskan mengenai hasil penelitian yang telah dilakukan serta pembahasan sistem yang telah dibuat. Pembahasan dilakukan guna menjelaskan dan memaparkan bagaimana penelitian ini menjawab perumusan masalah serta tujuan dari penelitian ini seperti apa yang telah ditentukan pada awal penelitian.

#### 6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

## BAB 2. TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Penelitian-penelitian terdahulu berfungsi sebagai pendukung bagi peneliti untuk melakukan penelitian baru dalam penerapan metode gamifikasi pada aplikasi e-commerce Macarina Jember. Penggalan informasi dari wacana penelitian terdahulu dilakukan sebagai upaya memperjelas variabel-variabel serta indikator-indikator yang berkaitan dengan penelitian ini, sekaligus untuk membedakan penelitian ini dengan penelitian sebelumnya. Kajian teori yang digunakan sebagai acuan dalam penelitian ini merupakan hasil penelitian dari kalangan akademis yang disajikan dalam bentuk jurnal dan skripsi.

Penelitian tentang penerapan metode gamifikasi telah banyak dilakukan, salah satunya adalah yang pernah dilakukan oleh Kardianawati, Haryanto dan Rosyidah melalui jurnal berjudul Penerapan Konsep Gamifikasi Berbasis Appreciative Inquiry untuk Peningkatan Daya Saing E-Marketplace Usaha Mikro, Kecil dan Menengah (UMKM). Hasil penelitian menunjukkan bahwa dengan diterapkannya konsep gamifikasi pada E-Marketplace mampu memberikan keunikan dan pembeda dari pesaing serta meningkatkan kemungkinan pembelian produk. Dan rangking pada produk yang sering dilihat juga mampu memotivasi penjual untuk menyediakan produknya sebaik-baiknya. Dengan hal-hal tersebut maka akan dapat meningkatkan daya saing dari E-Marketplace UMKM (Kardianawati, Haryanto, & Rosyidah, 2016).

Berdasarkan penelitian berjudul Pemanfaatan E-Commerce untuk Peningkatan Strategi Promosi dan Penjualan UMKM Tas di Kamupaten Kudus yang dilakukan oleh Noor Azizah, Danang Mahendra dan Budi Lofian menyebutkan bahwa dengan diterapkannya *e-commerce* pada UMKM tersebut mampu memperluas pangsa pasar dan juga peningkatan jumlah transaksi penjualan meskipun kenaikan omsetnya belum terlalu signifikan. Sedangkan untuk manfaat yang diperoleh dari penerapannya adalah mampu memberikan kemudahan dalam

bertransaksi dan rekapitulasi penjualan karena semua proses sudah dilakukan secara otomatis (Azizah, Mahendra, & Lofian, 2019).

## 2.2. *E-Commerce*

*Electronic Commerce* atau yang biasa disebut dengan *e-commerce* adalah sebuah konsep atau proses pembelian dan penjualan yang baru pada internet. Penerapan *e-Commerce* dimulai pada awal tahun 1970 dengan inovasi dari *Electronic Transfers (EFT)* (Nurdiansyah, Wijayanto, & Firdaus, 2018).

*E-Commerce* merupakan aktifitas yang berkaitan dengan pembelian, penjualan, pemasaran barang ataupun jasa dengan memanfaatkan system elektronik seperti internet ataupun jaringan computer. Kegiatan *e-commerce* ini merupakan aplikasi dan penerapan dari bisnis *online (e-business)*, *SCM (Supply Chain Management)*, pertukaran data elektronik (*electronic data interchange /EDI*), pemasaran *online (online marketing)*, dan sebagainya.

## 2.3. *Mobile Apps*

*Mobile apps* atau aplikasi mobile merupakan aplikasi yang dibuat untuk *smartphone* dan dapat diinstall serta dioperasikan di *smartphone* atau tablet di berbagai platform diantaranya adalah *iOS, android, windows phone, blackberry*, dsb. *Mobile apps* juga sebagai salah satu media tempat bertemunya pelanggan dengan produk/jasa dan ruang untuk pemasar produk memanjakan serta memuaskan para pelanggannya.

Berdasarkan penelitian yang berjudul Aplikasi Mobile Commerce Menggunakan PHP dan MySQL oleh Simarmata menyatakan bahwa, penggunaan mobile app sebagai sarana transaksi memiliki kelebihan dan kekurangan (Simamarta, 2006), sebagai berikut:

### 1. Kelebihan

Beberapa kelebihan diterapkannya e-commerce pada mobile app yaitu:

- a. Kepuasan pelanggan, penghematan biaya dan munculnya peluang bisnis baru.



- b. Menggunakan mobile commerce setiap waktu dan dimana saja.
- c. Mobile commerce dapat membawa penjual dan pembeli bersama-sama bertransaksi dengan mudah, sehingga memungkinkan untuk mendapatkan labayang lebih besar serta hubungan keduanya menjadi lebih dekat.

## 2. Kekurangan

Beberapa kekurangan diterapkannya e-commerce pada mobile app yaitu:

- a. Mobile device tidak bisa menawarkan grafik atau daya proses layaknya PC.
- b. Layar yang kecil dari mobile device membatasi kompleksitas aplikasi.

### 2.4. *Android*

*Android* adalah sistem operasi mobile yang berbasis *Linux* yang sudah dimodifikasi. *Android* dikembangkan dengan nama perusahaan yang sama, *Android, Inc.* Pada tahun 2005, sebagai bagian dari strategi untuk memasuki pasar mobile, *Google* membeli *Android* dan mengambil alih pengembangannya (begitu pula tim pengembangannya).

*Google* ingin agar *Android* menjadi terbuka dan bebas, karena itulah sebagian besar code *Android* dirilis di bawah lisensi *Apache open source*, yang artinya siapapun yang ingin menggunakan *Android* dapat melakukannya dengan cara mendownload seluruh source code *Android*. Selain itu, vendor (biasanya produsen hardware) dapat menambahkan ekstensi milik mereka sendiri untuk *Android* dan menyesuaikan *Android* untuk membedakan produk mereka dengan produk yang lain. Model pengembangan yang simpel membuat *Android* menjadi sangat atraktif dan menarik perhatian banyak vendor. Perkembangan versi kode *Android* dari tahun ke tahun bisa dilihat pada Tabel 2.2.

Tabel 2.1 Daftar Versi Kode Sistem Operasi *Android*

<i>Code Name</i>	<i>Version Number</i>	<i>Initial Release date</i>	<i>API Level</i>
------------------	-----------------------	-----------------------------	------------------

	1.0	23 September 2008	1
	1.1	9 Februari 2009	2
<i>Cupcake</i>	1.5	27 April 2009	3
<i>Donut</i>	1.6	15 September 2009	4
<i>Éclair</i>	2.0 – 2.1	26 Oktober 2009	5 – 7
<i>Froyo</i>	2.2 – 2.2.3	20 Mei 2010	8
<i>Gingerbread</i>	2.3 – 2.3.7	6 Desember 2010	9 – 10
<i>Honeycomb</i>	3.0 – 3.2.6	22 Februari 2011	11 – 13
<i>Ice Cream</i>	4.0 – 4.0.4	18 Oktober 2011	14 – 15
<i>Sandwich</i>			
<i>Jelly Bean</i>	4.1 – 4.3.1	9 Juli 2012	16 – 18
<i>KitKat</i>	4.4 – 4.4.4	31 Oktober 2013	19 – 20
<i>Lollipop</i>	5.0 – 5.1.1	12 Nopember 2014	21 – 22
<i>Marshmallow</i>	6.0 – 6.0.1	5 Oktober 2015	23
<i>Nougat</i>	7.0 – 7.1.2	22 Agustus 2016	24 – 25
<i>Oreo</i>	8.0 – 8.1	21 Agustus 2015	26
<i>Pie</i>	9.0	7 Maret 2018	27

## 2.5. Gamifikasi

Gamifikasi adalah sebuah proses yang bertujuan mengubah *non-gamecontext* (contoh: belajar, mengajar, pemasaran, dan sebagainya) menjadi lebih menarik dengan mengintegrasikan *game thinking*, *game design*, dan *game mechanics*. Beberapa unsur-unsur yang sering terdapat di dalam *game* antara lain (Aribowo, 2017):

1. *Point*

*Points*, ada beberapa *game* yang menyebut dengan istilah *experience points* (*XP*). *Points* dapat digunakan untuk menandai peningkatan dan dapat digunakan untuk membuka konten yang terkunci, bahkan dapat berfungsi sebagai mata uang yang dapat digunakan untuk membeli benda-benda virtual yang kita inginkan.

## 2. *Badge*

*Badges*, merupakan emblem atau lencana digital yang kita peroleh apabila telah berhasil menuntaskan misi atau tantangan tertentu. *Badges* dapat berupa pita, tropi, atau lambang lain. Pemain biasanya mendapatkan *badges* untuk mengukur pencapaian di skill tertentu yang dimaksud pada game.

## 3. *Leaderboard*

*Leaderboards* merupakan daftar nama-nama pemain peringkat atas menurut kesuksesan mereka dalam game. Hal ini merupakan penunjuk perolehan *points* sementara. Konsep yang sama yang biasanya digunakan dalam bidang olahraga – klasemen, tetapi lebih digunakan untuk *game* yang multi-player, khususnya yang dibatasi oleh waktu atau misi tertentu. *Leaderboards* bersifat real-time sehingga kita dapat mengetahui posisi kita secara langsung. Dengan demikian, kita mengetahui dengan pasti berapa jumlah *points* kita, peringkat kita, serta pemimpin (peringkat pertama) berikut perolehan *points*-nya.

## 4. *Level*

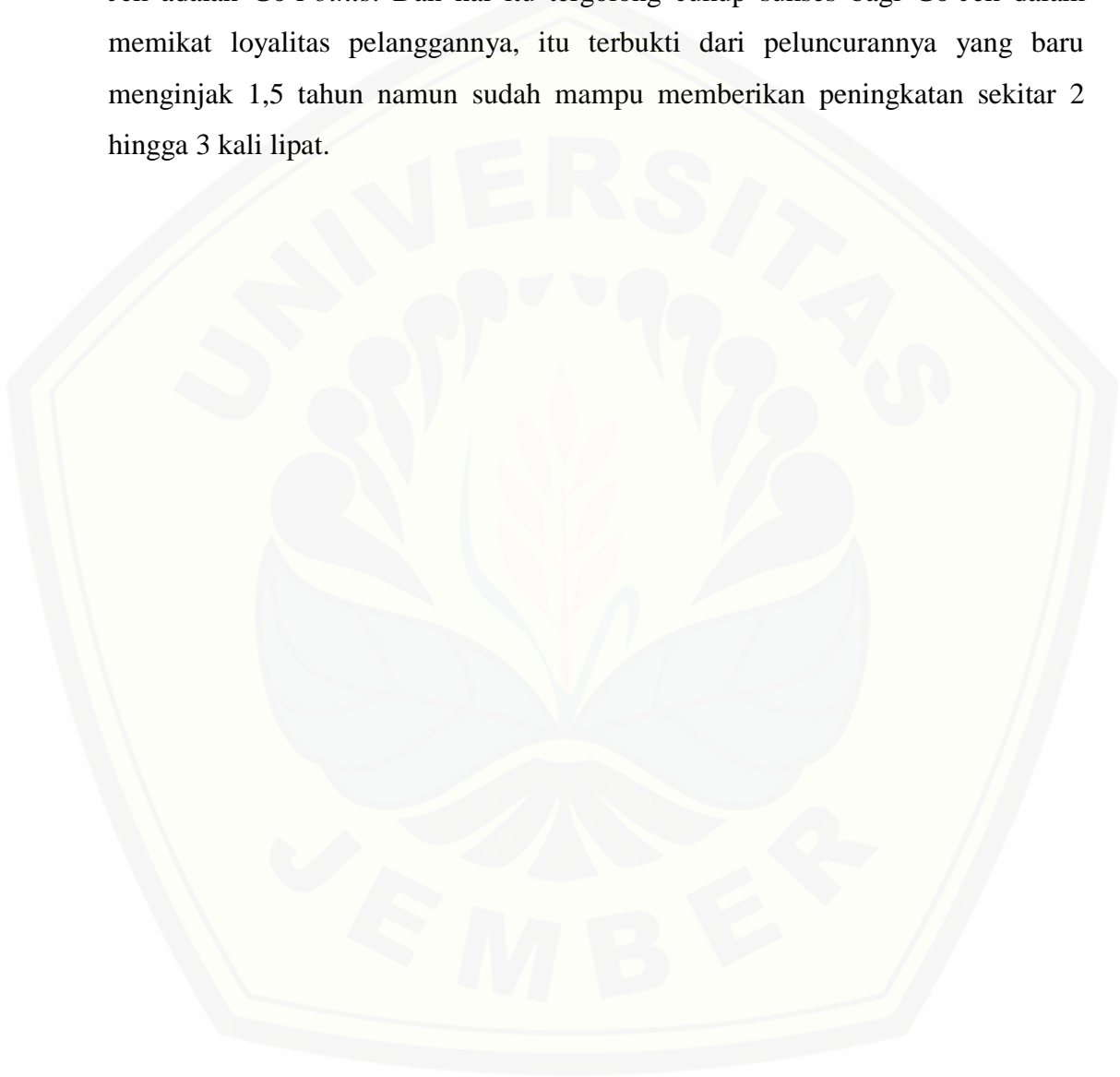
*Level* merupakan jenjang atau tingkat kesulitan. Semakin tinggi level berarti semakin tinggi pula kesulitan serta kekompleksan misi atau tugas yang harus diselesaikan. *Level 1* harus dituntaskan apabila kita ingin memainkan *level 2* (meskipun ada beberapa *game* yang membatasi dengan beberapa *level* sekaligus). Intinya, ada fitur-fitur atau *level-level* tertentu yang terkunci (*unlocked*) dan belum bisa dimainkan apabila kita belum menyelesaikan misi, tugas, atau *level* sebelumnya.

## 5. *Reward*

*Reward* merupakan sebuah hadiah yang akan didapat oleh si pengguna ketika telah menyelesaikan tantangan yang ada dalam sebuah game.



Dalam penelitian ini, elemen gamifikasi yang akan diterapkan dalam aplikasi Macarina adalah *point*, *reward* dan *leaderboard*. Dipilihnya 3 elemen gamifikasi tersebut dikarenakan dalam sebuah kutipan yang ditulis oleh Agus Tri Haryanto pada Detik Inet (Haryanto, 2017), menuliskan bahwa salah satu program loyalti Go-Jek adalah *Go-Points*. Dan hal itu tergolong cukup sukses bagi Go-Jek dalam memikat loyalitas pelanggannya, itu terbukti dari peluncurannya yang baru menginjak 1,5 tahun namun sudah mampu memberikan peningkatan sekitar 2 hingga 3 kali lipat.



## BAB 3. METODOLOGI PENELITIAN

### 3.1. Tempat Penelitian

Penelitian ini dilakukan di Usaha Mikro, Kecil dan Menengah (UMKM) Macarina Jember.

### 3.2. Waktu Penelitian

Waktu penelitian ini berlangsung kurang lebih 2 bulan, mulai bulan Desember 2018 sampai bulan Januari 2019.

### 3.3. Jenis Penelitian

Jenis penelitian yang akan dilakukan adalah penelitian terapan (*applied research*). Merancang, mendesain, dan membangun sebuah aplikasi untuk Usaha Mikro, Kecil dan Menengah (UMKM) dengan penerapan metode gamifikasi pada *e-commerce* berbasis *mobile app*.

### 3.4. Tahapan Penelitian

Tahapan penelitian merupakan urutan langkah penelitian yang dilakukan. Gambaran tahapan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

#### 3.4.1. Studi Literatur

Tahap awal dalam penelitian ini adalah studi literatur yang bertujuan untuk penyusunan dasar teori yang akan digunakan selama penelitian. Penelitian menggunakan referensi dalam buku, jurnal baik nasional maupun internasional, karya ilmiah, artikel dan penelitian – penelitian yang sudah dilakukan sebelumnya serta berbagai sumber lain yang mendukung dalam penelitian ini.

### 3.4.2. Pengumpulan Data

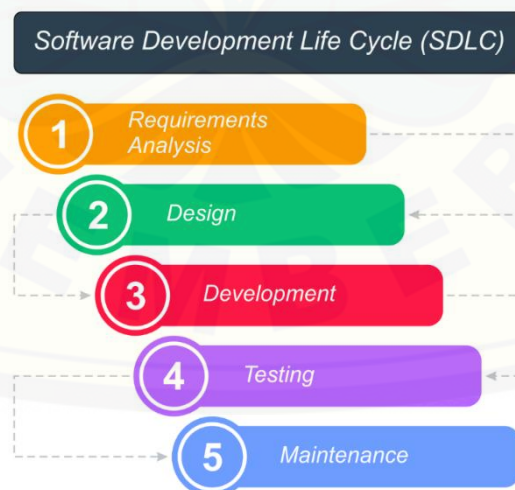
Tahap selanjutnya adalah pengumpulan data. Data yang diperoleh untuk melakukan penelitian didapat langsung dari Usaha Mikro, Kecil dan Menengah (UMKM) Macarina Jember.

### 3.4.3. Pemilihan Elemen Gamifikasi

Pemilihan elemen gamifikasi ditujukan mengetahui elemen gamifikasi yang akan digunakan pada aplikasi Macarina. Pada penelitian kali ini memakai elemen gamifikasi *point*, *leaderboard* dan *reward*.

### 3.4.4. Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak dalam penelitian ini menggunakan *Software Development Life Cycle (SDLC) waterfall*. Menurut Pressman (2010), model *SDLC Waterfall* adalah proses pengembangan perangkat lunak sekuensial dimana perkembangan terlihat seperti mengalir semakin ke bawah (mirip dengan air terjun) melalui daftar tahapan yang harus dijalankan agar berhasil membangun sebuah perangkat lunak komputer. Alur pengembangan *software* dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Software Development Life Cycle (SDLC) Waterfall*

### 1. Analisis Kebutuhan (*Required Analysis*)

Tahapan awal dalam perancangan dan pengembangan sistem ini adalah analisis kebutuhan-kebutuhan tentang bagaimana aplikasi *e-commerce* berbasis *mobile apps* yang akan diterapkan pada Usaha Mikro Kecil dan Menengah (UMKM) Macarina Jember. Pada tahapan ini peneliti melakukan beberapa studi literatur yang berkaitan dengan metode yang hendak diterapkan dan juga studi literatur lainnya yang dibutuhkan untuk pengembangan sistemnya.

### 2. Desain Sistem (*Design*)

Tahap desain sistem yang akan dibangun menggunakan *Unified Modeling Language (UML)* yang mendukung konsep pemodelan *programming* berbasis objek. Pada tahap ini akan diperoleh dokumentasi antara lain:

- a) *Input and Output Diagram* digunakan untuk mendefinisikan aktifitas dan proses.
- b) *Use case Diagram* digunakan untuk mendefinisikan fungsional sistem.
- c) *Scenario* digunakan untuk menjelaskan fitur sistem.
- d) *Database Structure* digunakan untuk menunjukkan relasi antar objek.

### 3. Implementasi (*Development*)

Tahapan implementasi adalah tahap yang akan mengkonversi desain sistem yang telah dibuat ke dalam bentuk kode program. Sistem yang akan dibangun ditulis dengan Bahasa Pemrograman *JAVA* dan menggunakan *Android Studio* sebagai *IDE (Integrated Development Environment)* pengembangan aplikasi. Sistem ini menggunakan *server* dan *database* yang dimiliki oleh *Google*, yakni *Firebase Realtime Database*.

### 4. Pengujian (*Testing*)

Tahapan selanjutnya adalah tahap pengujian (*testing*). Pada penelitian ini pengujian dilakukan menggunakan metode *Black box Testing*. *Black box Texting* merupakan sebuah pengujian yang melibatkan pengguna/*user*, dimana hanya

memperhatikan fungsionalitas yang berkaitan dengan masukan/keluaran (*I/O*) apakah sesuai dengan sistem yang dijalankan.

5. Pemeliharaan (*Maintenance*)

Pemeliharaan diadakan untuk mengatasi masalah pada sistem dilain waktu ketika aplikasi sudah dapat digunakan oleh *user*. Selama *user* menemui *bug* pada sistem, maka *user* langsung konfirmasi kepada *developer* untuk segera ditangani.



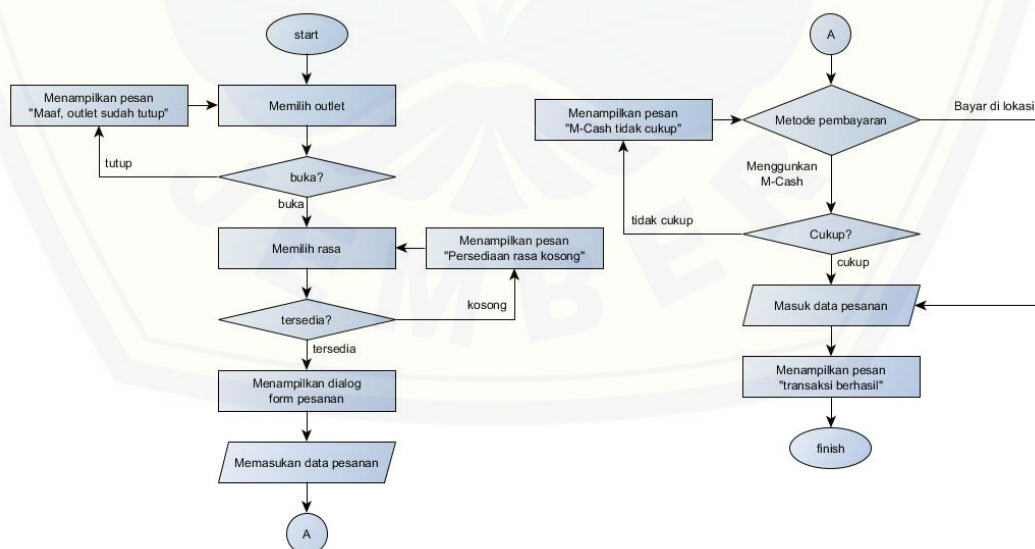
## BAB 4. RANCANGAN SISTEM

### 4.1. Deskripsi Umum Sistem

Aplikasi Macarina merupakan sebuah aplikasi berbasis *mobile apps* yang digunakan untuk pemesanan produk khusus Usaha Mikro, Kecil dan Menengah (UMKM) Macarina Jember. Pada aplikasi ini *customer* atau pelanggan disediakan 2 metode pembayaran yakni bayar di tempat dan bayar menggunakan *M-Cash*. Setelah *customer* berhasil melakukan transaksi menggunakan aplikasi ini, maka akan mendapatkan bonus berupa *M-Poin* sesuai produk yang dipesan tadi. Yang nantinya *M-Poin* ini bisa digunakan untuk melakukan penukaran dengan hadiah-hadiah yang sudah disediakan oleh pihak UMKM Macarina pada aplikasi Macarina.

#### 4.1.1. Pembelian Produk Macarina

*M-Cash* merupakan uang digital yang nantinya dapat digunakan sebagai salah satu metode pembayaran dalam pembelian produk di UMKM Macarina. *M-Cash* sendiri didapatkan dengan cara melakukan penukaran uang asli di *outlet-outlet* UMKM Macarina yang sudah tersebar di Jember. Untuk alur pemesanan produk Macarina menggunakan *M-Cash* bisa dilihat pada Gambar 4.1.

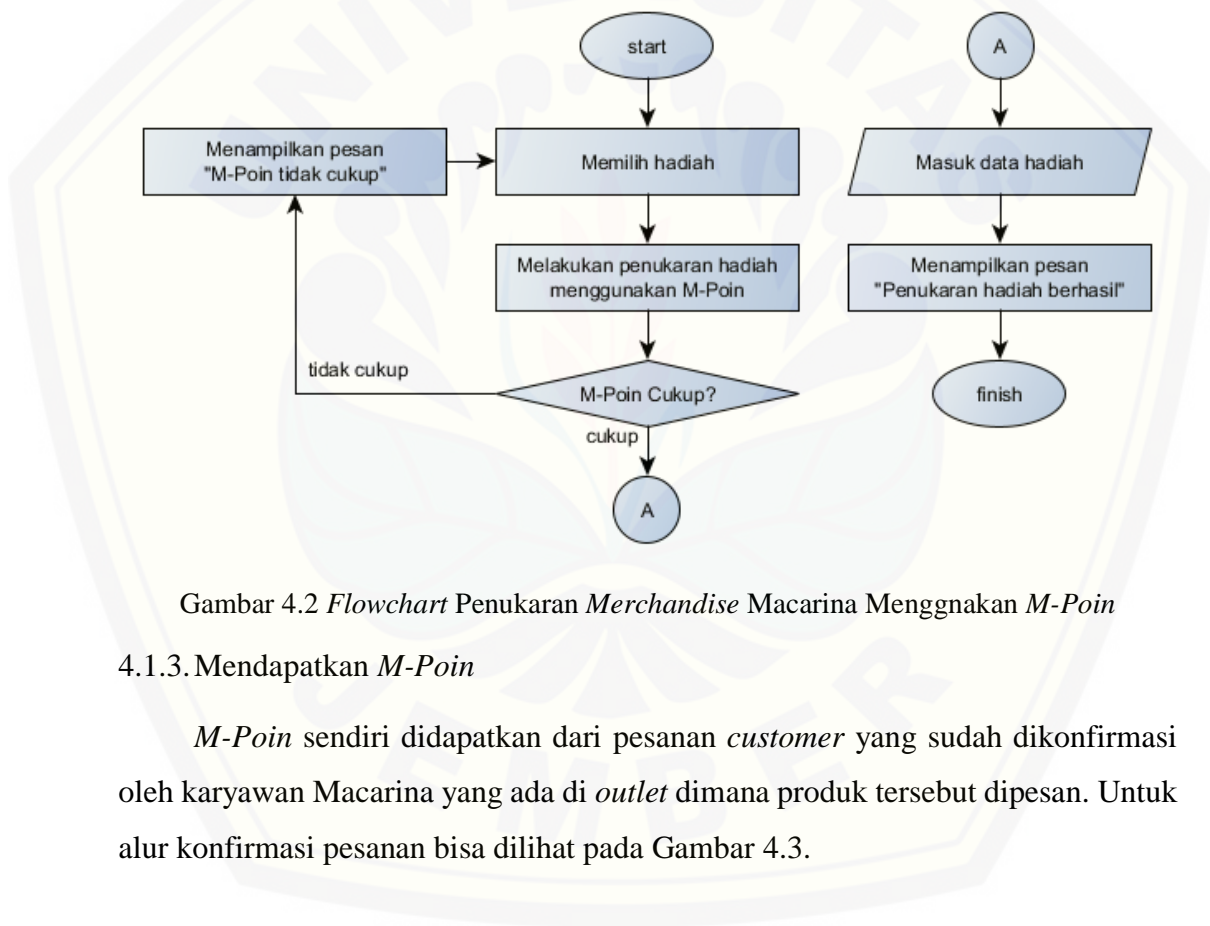


Gambar 4.1 Flowchart Pembelian Produk Macarina



#### 4.1.2. Penukaran *Merchandise* Menggunakan *M-Poin*

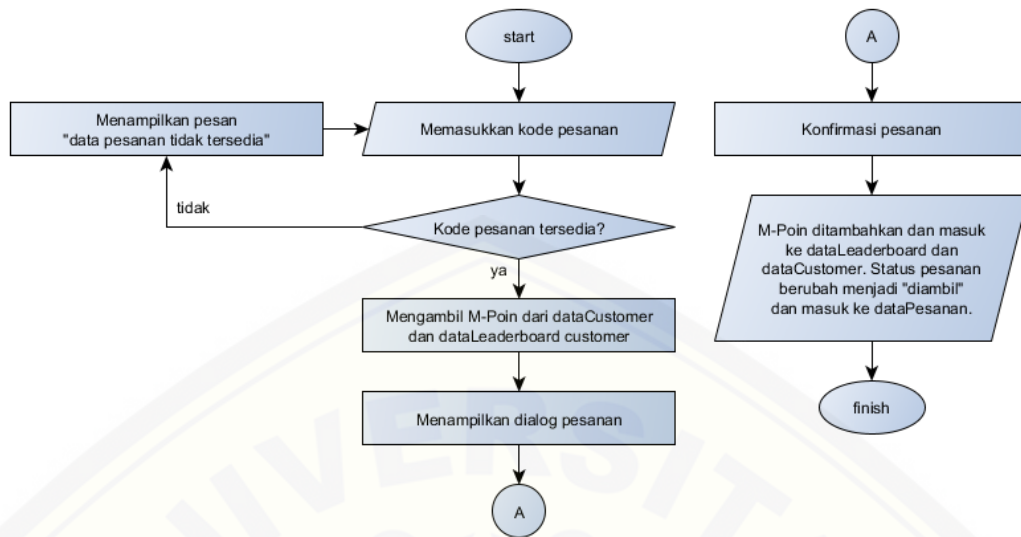
*M-Poin* merupakan poin yang nantinya dapat ditukarkan dengan *merchandise* yang sudah disediakan oleh pihak UMKM Macarina. *M-Poin* sendiri merupakan sebuah bonus dari pembelian produk menggunakan aplikasi Macarina. Jadi semakin banyak melakukan pembelian produk lewat aplikasinya, maka akan semakin banyak pula kesempatan untuk dapat menukarkannya dengan *merchandisenya*. Untuk alur penukaran *merchandise* menggunakan *M-Poin* bisa dilihat pada Gambar 4.2.



Gambar 4.2 Flowchart Penukaran *Merchandise* Macarina Menggunakan *M-Poin*

#### 4.1.3. Mendapatkan *M-Poin*

*M-Poin* sendiri didapatkan dari pesanan *customer* yang sudah dikonfirmasi oleh karyawan Macarina yang ada di *outlet* dimana produk tersebut dipesan. Untuk alur konfirmasi pesanan bisa dilihat pada Gambar 4.3.



Gambar 4.3 *Flowchart* Konfirmasi Pesanan *Customer* oleh Karyawan

#### 4.2. Analisa Kebutuhan

Kegiatan analisa kebutuhan merupakan kegiatan memahami dengan sebenarnya kebutuhan dari sistem yang akan dibangun berdasarkan dari fungsional sistem yang akan dibangun.

Alat yang digunakan dalam penelitian ini adalah *hardware* berupa *Smartphone Android* minimal 4.0.3 (*Ice Cream Sandwich*) dan satu unit laptop yang didalamnya terdapat *software* sebagai berikut:

1. *Windows 10*
2. *Browser*
3. *Android Studio*
4. Bahasa Pemrograman *Java*
5. *Realtime Firebase Database*
6. *Microsoft Office 2019*
7. *CorelDraw 2018*
8. *AdobeXD 2018*



#### 4.2.1. Studi Literatur

Studi literatur yang digunakan berasal dari jurnal, website dan penelitian terdahulu. Studi literatur mengenai penerapan metode gamifikasi, sehingga didapatkan kebutuhan fungsional dan non-fungsional yang sesuai. Kebutuhan tersebut dapat digunakan dalam melakukan peningkatan pelanggan terhadap UMKM Macarina. Melalui studi literatur yang telah dilakukan, didapatkan hasil berupa data dan informasi yang selanjutnya diolah untuk menganalisis kebutuhan fungsional dan non-fungsional.

#### 4.2.2. Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari aplikasi ini yaitu:

##### 1. *Admin*

Pada aktor *admin* ini, aktor memiliki beberapa kebutuhan fungsional yaitu:

- a. Sistem mampu mengelola data karyawan (melihat, mengubah, menghapus dan menambah).
- b. Sistem mampu mengelola data outlet (melihat, mengubah dan menambah).
- c. Sistem mampu mengelola data informasi (melihat, mengubah, menghapus dan menambah).
- d. Sistem mampu mengelola data hadiah (melihat, mengubah, menghapus dan menambah).
- e. Sistem mampu mengelola data rasa (melihat, menghapus dan menambah).
- f. Sistem mampu melakukan verifikasi hadiah *customer*.
- g. Sistem mampu melakukan *top-up M-Cash* untuk *customer*.
- h. Sistem mampu mengelola data *leaderboard* (melihat, mengubah *reward* dan mengirim *reward*).
- i. Sistem mampu mengelola data diskon (melihat dan mengubah).

## 2. Karyawan

Pada aktor karyawan ini, aktor memiliki beberapa kebutuhan fungsional yaitu:

- a. Sistem mampu mengelola data outlet (mengubah status buka).
- b. Sistem mampu mengelola data rasa (mengubah status persediaan).
- c. Sistem mampu melakukan verifikasi pesanan *customer* yang sudah selesai.
- d. Sistem mampu mengelola data pesanan (mengubah status pesanan).
- e. Sistem mampu menampilkan informasi.
- f. Sistem mampu melakukan *top-up M-Cash* untuk *customer*.

## 3. Customer

Pada aktor *customer* ini, aktor memiliki beberapa kebutuhan fungsional yaitu:

- a. Sistem mampu menampilkan data outlet (status buka, alamat, nomor telepon).
- b. Sistem mampu menampilkan data rasa, sekaligus menampilkan status ketersediaannya.
- c. Sistem mampu melakukan pemesanan produk Macarina.
- d. Sistem mampu menampilkan data informasi.
- e. Sistem mampu menampilkan data *leaderboard customer*.
- f. Sistem mampu menampilkan data hadiah.
- g. Sistem mampu menampilkan data pesanan (pesanan belum diproses, pesanan sedang diproses, pesanan ditolak, pesanan selesai diproses, pesanan sudah diambil)
- h. Sistem mampu menampilkan data rasa favorit yang sering dipesan.
- i. Sistem mampu menampilkan riwayat *top-up M-Cash*.
- j. Sistem mampu menampilkan riwayat perolehan *M-Poin*.

- k. Sistem mampu melakukan pembuatan akun baru.

#### 4.2.3. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang merupakan sekumpulan karakteristik dan properti pada sistem:

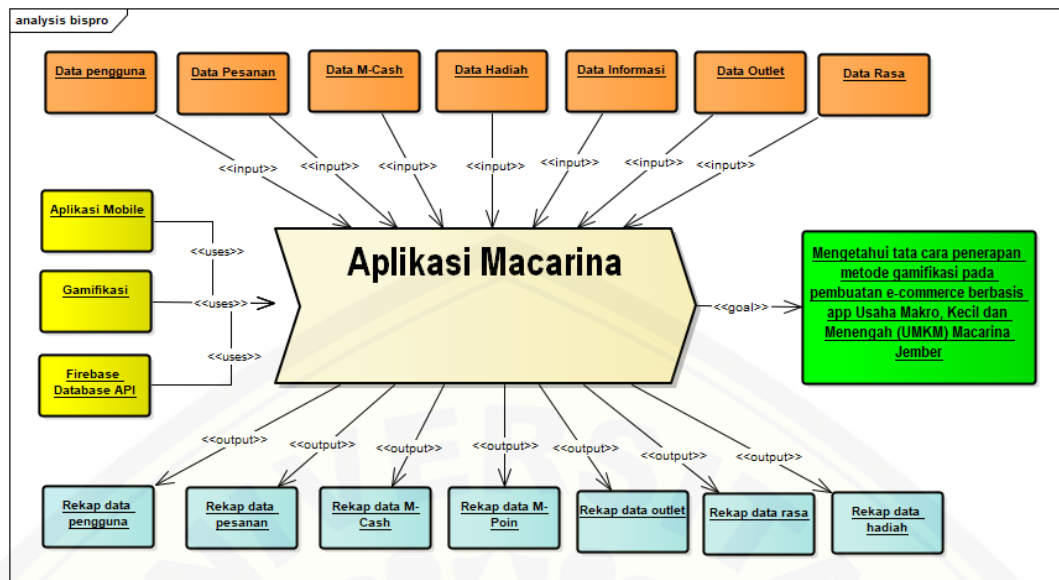
1. Sistem menggunakan autentikasi berupa *email address* dan *password*.
2. Sistem menggunakan *Firebase Realtime Database*, yakni *database* dari *Google*.
3. Pemesanan produk bisa menggunakan *M-Cash (e-Money Macarina)* atau uang tunai.
4. Hadiah yang sudah disiapkan oleh Macarina bisa didapatkan dengan menukarkannya menggunakan *M-Poin*.

### 4.3. Desain Sistem

Desain sistem yang dibuat meliputi *input and output diagram*, *use case diagram*, *scenario*, *sequence diagram*, *activity diagram*, *class diagram* dan *database structure*.

#### 4.3.1. *Input and Output Diagram*

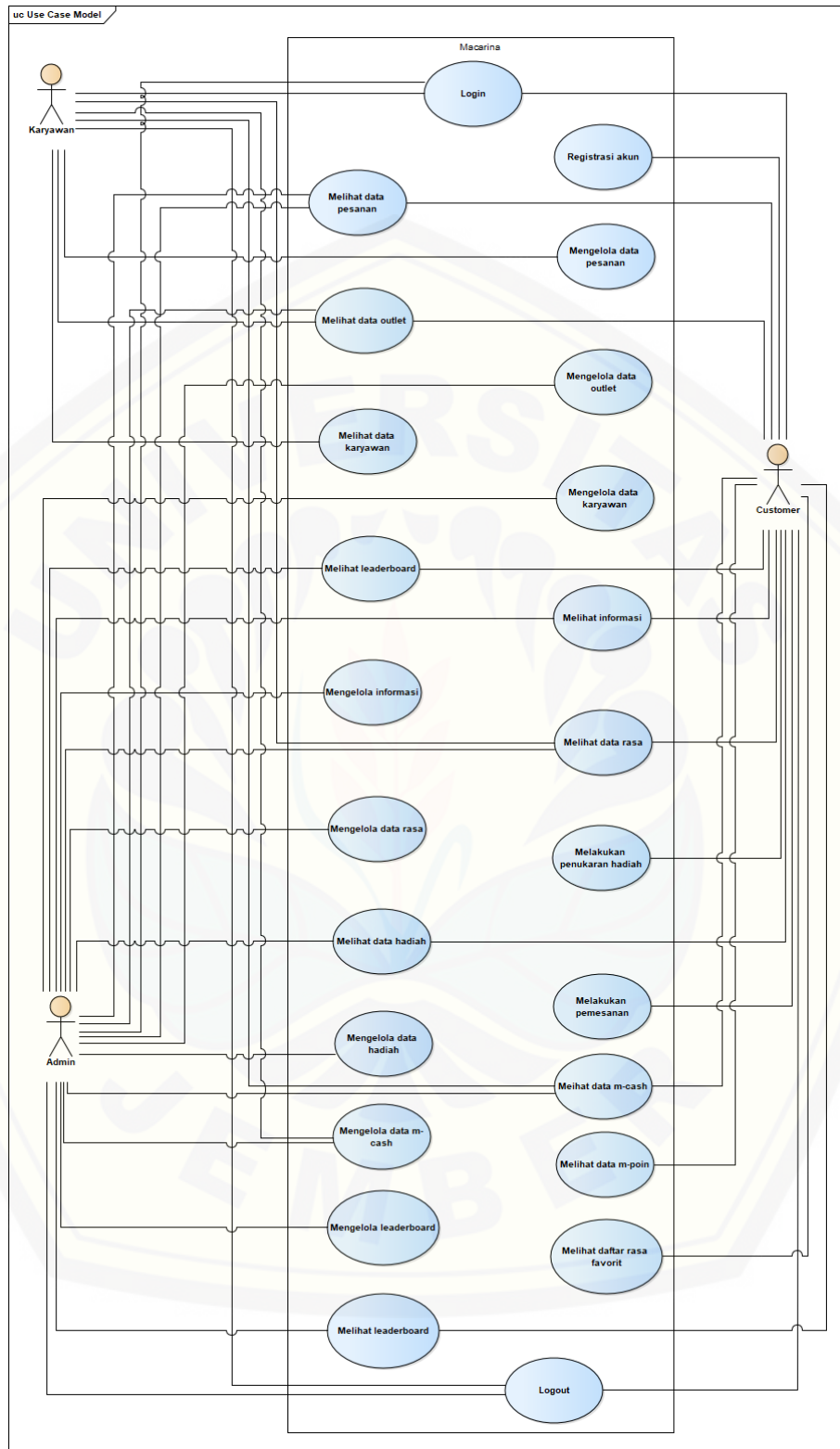
*Input and Output Diagram* merupakan suatu kumpulan dari aktivitas atau pekerjaan terstruktur yang saling berhubungan untuk menghasilkan suatu produk. *Diagram* ini digunakan untuk mengetahui masukan (*input*), keluaran (*output*) dan hasil dari sistem. *Input and output diagram* pada sistem ini dapat dilihat pada Gambar 4.4.



Gambar 4.4 Input and Output Diagram

#### 4.3.2. Use case Diagram

*Use case Diagram* menggambarkan fungsionalitas dari sebuah sistem dan interaksi antara *user* dengan sistem untuk melakukan suatu pekerjaan tertentu. *Use case Diagram* juga menggambarkan hak akses dari setiap aktor. *Use case Diagram* pada sistem ini dapat dilihat pada Gambar 4.5.



Gambar 4.5 Use case Diagram

#### 4.3.3. Scenario

*Scenario* merupakan penjabaran alur kerja sistem yang terdapat pada *use case diagram*. *Scenario* pada aplikasi *e-commerce* berbasis *mobile apps* yang menerapkan metode gamifikasi adalah sebagai berikut:

a. *Scenario Login*

*Scenario login* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor masuk ke dalam sistem. Hanya aktor yang memiliki *e-email* dan *password* yang dapat masuk. Terdapat tiga *user* dalam sistem ini, yaitu: *admin*, karyawan dan *customer*. *Scenario login* dapat dilihat pada lampiran A.

b. *Scenario registrasi*

*Scenario* registrasi merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melakukan registrasi pembuatan akun baru. Aktor dapat memasukkan nama, *email*, *password* dan nomor telepon. *Scenario* registrasi dapat dilihat pada lampiran A.

c. *Scenario* melihat data pesanan

*Scenario* melihat data pesanan merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat produk yang telah dipesan, seperti: produk yang dipesan, yang sedang diproses, yang sudah selesai diproses, yang ditolak dan riwayat pesanan. *Scenario* melihat data pesanan dapat dilihat pada lampiran A.

d. *Scenario* mengelola data pesanan

*Scenario* mengelola data pesanan merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melakukan perubahan status pada pesanan. Aktor dapat merubah status pesanan menjadi sedang diproses, ditolak ataupun sudah selesai. *Scenario* mengelola data pesanan dapat dilihat pada lampiran A.

e. *Scenario* melihat data *outlet*

*Scenario* melihat data *outlet* merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat data *outlet*. Aktor dapat melihat data *outlet*,



seperti: alamat, status (buka atau tutup) dan nomor yang dapat dihubungi pada *outlet* tersebut. *Scenario* melihat data *outlet* dapat dilihat pada lampiran A.

f. *Scenario* mengelola data *outlet*

*Scenario* mengelola data *outlet* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data *outlet*. Aktor dapat melihat, merubah dan menambah *outlet* baru dengan memilih menu *outlet*. *Scenario* mengelola data *outlet* dapat dilihat pada lampiran A.

g. *Scenario* mengelola data karyawan

*Scenario* mengelola data karyawan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data karyawan. Aktor dapat melihat, merubah dan menambah *outlet* baru dengan memilih menu *outlet*. *Scenario* mengelola data karyawan dapat dilihat pada lampiran A.

h. *Scenario* melihat *leaderboard*

*Scenario* melihat *leaderboard* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor melihat data *leaderboard*. Aktor dapat melihat *reward* yang akan didapat dan juga user yang berada pada urutan paling atas dengan perolehan poin paling tinggi pada bulan tersebut. *Scenario* melihat *leaderboard* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Melihat *Leaderboard*

No. <i>Use case</i>	UC08
Nama <i>use case</i>	Melihat <i>leaderboard</i>
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> memilih menu <i>leaderboard</i>
PostKondisi	<i>Customer</i> berhasil melihat daftar <i>leaderboard</i> dan melihat <i>reward</i> yang didapat
Aliran Normal Melihat daftar <i>leaderboard</i>	
Aksi Aktor	Reaksi Sistem

	<ol style="list-style-type: none"> <li>1. Menampilkan <i>reward</i> dan daftar <i>leaderboard</i> yang berisi: <ul style="list-style-type: none"> <li>- Poin</li> <li>- Nama</li> <li>- Foto</li> </ul> </li> </ol>
--	---

i. *Scenario* mengelola data *leaderboard*

*Scenario* mengelola data *leaderboard* merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data *leaderboard*. Aktor dapat mengubah *reward*, melihat user dengan urutan teratas dan mengirim *reward* kepada user tersebut dengan memilih menu *leaderboard*. *Scenario* mengelola data *leaderboard* dapat dilihat pada lampiran A.

j. *Scenario* melihat informasi *customer*

*Scenario* melihat informasi merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat informasi. Aktor dapat melihat informasi saat ini dan yang akan datang. *Scenario* melihat informasi *customer* dapat dilihat pada lampiran A.

k. *Scenario* melihat informasi karyawan

*Scenario* melihat informasi merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat informasi. Aktor dapat melihat informasi saat ini dan yang akan datang. *Scenario* melihat informasi karyawan dapat dilihat pada lampiran A.

l. *Scenario* mengelola informasi

*Scenario* mengelola informasi merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor mengelola informasi. Aktor dapat melihat, merubah, menghapus dan menambah informasi. *Scenario* mengelola informasi dapat dilihat pada lampiran A.

m. *Scenario* melihat data rasa *customer*

*Scenario* melihat data rasa merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat data rasa. Aktor dapat melihat rasa yang masih tersedia atau kosong. *Scenario* melihat data rasa *customer* dapat dilihat pada lampiran A.

n. *Scenario* melihat data rasa karyawan

*Scenario* melihat data rasa merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat data rasa. Aktor dapat melihat rasa yang masih tersedia atau kosong. *Scenario* melihat data rasa karyawan dapat dilihat pada lampiran A.

o. *Scenario* mengelola data rasa

*Scenario* mengelola data rasa merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor mengelola data rasa. Aktor dapat melihat, merubah, menghapus dan menambah rasa baru. *Scenario* mengelola data rasa dapat dilihat pada lampiran A.

p. *Scenario* melakukan penukaran hadiah

*Scenario* melakukan penukaran hadiah merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melakan penukaran hadiah menggunakan *M-Poin*. *Scenario* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Melakukan Penukaran Hadiah

No. <i>Use case</i>	UC16
Nama <i>use case</i>	Melakukan penukaran hadiah
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> berada pada halaman <i>home customer</i>
PostKondisi	<i>Customer</i> berhasil melakukan penukaran hadiah menggunakan <i>M-Poin</i>
Aliran Normal Menukar hadiah menggunakan <i>M-Poin</i>	
Aksi Aktor	Reaksi Sistem

1. Klik tombol tukar	
	2. Sistem menyimpan hadiah ke database
	3. Menampilkan pesan “penukaran hadiah telah berhasil dilakukan”
	4. Menampilkan halaman <i>home customer</i>
Aliran Alternatif	
1. Klik tombol tukar	
	2. Sistem mengecek <i>M-Poin customer</i>
	3. Menampilkan pesan “ <i>M-Poin</i> tidak cukup”
	4. Menampilkan halaman <i>home customer</i>

q. *Scenario* verifikasi hadiah

*Scenario* verifikasi hadiah merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melakukan verifikasi pada hadiah. *Scenario* verifikasi hadiah dapat dilihat pada lampiran A.

r. *Scenario* melihat data hadiah

*Scenario* melihat data hadiah merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat data hadiah. Aktor dapat melihat jumlah *M-Poin* yang dibutuhkan untuk dapat melakukan penukaran hadiah. *Scenario* melihat data hadiah dapat dilihat pada lampiran A.

s. *Scenario* melakukan pemesanan

*Scenario* melakukan pemesanan merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melakukan pemesanan produk. Aktor dapat memilih jenis kemasan, level dan juga metode pembayaran (di lokasi atau menggunakan *M-Cash*). *Scenario* melakukan pemesanan dapat dilihat pada Tabel 4.3.

Tabel 4.3 Melakukan Pemesanan

No. <i>Use case</i>	UC18
Nama <i>use case</i>	Melakukan pemesanan

Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> melihat persediaan rasa pada <i>outlet</i>
PostKondisi	<i>Customer</i> berhasil melakukan pemesanan
Aliran Normal Pemesanan produk	
Aksi Aktor	Reaksi Sistem
1. Klik tombol pesan	
	2. Menampilkan dialog pemesanan yang berisi: <ul style="list-style-type: none"> <li>- Rasa</li> <li>- <i>Outlet</i> tempat pesan</li> <li>- Alamat <i>outlet</i></li> <li>- Kemasan</li> <li>- Level</li> <li>- Jumlah pesanan</li> <li>- Pembayaran</li> <li>- Diskon</li> <li>- Harga</li> <li>- Harga setelah diskon</li> <li>- <i>M-Poin</i> yang akan didapatkan</li> </ul>
3. Menentukan pembayaran, jumlah, kemasan dan level	
4. Klik pesan	
	5. Sistem menyimpan data pesanan ke database
	6. Menampilkan pesan “pemesanan telah berhasil dilakukan”
	7. Menampilkan halaman daftar rasa
Aliran Alternatif Status rasa kosong	
1. Klik tombol pesan	
	2. Sistem mengecek status persediaan
	3. Menampilkan pesan “rasa kosong”
	4. Menampilkan halaman daftar rasa
Aliran Alternatif	

<i>M-Cash</i> tidak cukup	
3. Memilih metode pembayaran <i>M-Cash</i>	
4. Klik pesan	
	5. Sistem mengecek <i>M-Cash customer</i>
	6. Menampilkan pesan “ <i>M-Cash</i> tidak cukup”
	7. Menampilkan dialog pemesanan
Aliran Alternatif Koneksi internet bermasalah	
4. Klik pesan	
	5. Sistem mengecek koneksi internet
	6. Menampilkan pesan “koneksi internet bermasalah”
	7. Menampilkan dialog pemesanan

t. *Scenario* verifikasi pesanan

*Scenario* verifikasi pesanan merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melakukan verifikasi pesanan. *Scenario* verifikasi pesanan dapat dilihat pada Tabel 4.4.

Tabel 4.4 Verifikasi Pesanan

No. <i>Use case</i>	UC24
Nama <i>use case</i>	Verifikasi pesanan
Aktor	Karyawan
PreKondisi	Karyawan pilih menu verifikasi
PostKondisi	Karyawan berhasil verifikasi pesanan
Aliran Normal Verifikasi pesanan	
Aksi Aktor	Reaksi Sistem
1. Masukkan kode pesanan	
2. Klik cek	



	3. Menampilkan dialog detail pesanan
4. Klik konfirmasi	
	5. Merubah status pesanan dan menyimpannya ke database
	6. Menampilkan pesan “pesanan berhasil diverifikasi”
Aliran Alternatif Kode kosong	
2. Klik cek	
	3. mengecek kode pesanan
	4. menampilkan pesan “masukkan kode pesanan”
Aliran Alternatif Internet bermasalah	
4. klik konfirmasi	
	5. mengecek koneksi internet
	6. menampilkan pesan “koneksi internet bermasalah”
Aliran Alternatif Salah <i>outlet</i>	
4. klik konfirmasi	
	5. mengecek <i>outlet</i>
	6. menampilkan pesan “salat <i>outlet</i> ”
Aliran Alternatif Masih diproses	
4. klik konfirmasi	
	5. mengecek pesanan
	6. menampilkan pesan “masih diproses”
Aliran Alternatif Sudah diambil	
4. klik konfirmasi	
	5. mengecek pesanan

	6. menampilkan pesan “pesanan sudah diambil”
--	--

u. *Scenario* mengelola data hadiah

*Scenario* mengelola data hadiah merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor mengelola hadiah. Aktor dapat melihat, merubah, menghapus dan menambah hadiah. *Scenario* mengelola hadiah dapat dilihat pada lampiran A.

v. *Scenario* melihat data *M-Cash*

*Scenario* melihat data *M-Cash* merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat riwayat pembelian *M-Cash*. *Scenario* melihat data *M-Cash* dapat dilihat pada lampiran A.

w. *Scenario* mengelola data *M-Cash*

*Scenario* mengelola data *M-Cash* merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melakukan pengisian *M-Cash* pada user. *Scenario* mengelola data *M-Cash* dapat dilihat pada lampiran A.

x. *Scenario* melihat data *M-Poin*

*Scenario* melihat data *M-Poin* merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat riwayat *M-Poin* yang didapatkan. *Scenario* melihat data *M-Poin* dapat dilihat pada lampiran A.

y. *Scenario* *logout*

*Scenario* *logout* merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor keluar dari sistem. *Scenario* *logout* dapat dilihat pada lampiran A.

z. *Scenario* rasa favorit

*Scenario* rasa favorit merupakan alur yang menjelaskan aktor dan reaksi sistem pada saat aktor melihat daftar rasa yang paling diminati. *Scenario* rasa favorit dapat dilihat pada lampiran A.

#### 4.3.4. Activity diagram

*Activity diagram* merupakan diagram yang memodelkan proses-proses yang terjadi pada sistem dan merupakan pemodelan dari *scenario* yang sudah ditetapkan sebelumnya. *Activity diagram* dalam sistem ini antara lain sebagai berikut:

a. *Activity diagram login*

*Activity diagram login* dilakukan oleh *customer*, karyawan dan *admin*. *Activity diagram login* menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan autentifikasi hak akses semua aktor dalam menggunakan sistem. *Activity diagram login* bisa dilihat pada lampiran C.

b. *Activity diagram registrasi*

*Activity diagram registrasi* dilakukan oleh *customer*. *Activity diagram registrasi* menjelaskan tentang bagaimana sistem dapat melakukan pembuatan akun bagi *customer* baru. *Activity diagram registrasi* bisa dilihat pada lampiran C.

c. *Activity diagram melihat data pesanan*

*Activity diagram melihat data pesanan* dilakukan oleh *customer*. *Activity diagram melihat data pesanan* menjelaskan tentang bagaimana sistem dapat melihat data pesanan yang dimiliki *customer*, meliputi pesanan yang belum diproses, pesanan sedang diproses, pesanan ditolak, pesanan selesai diproses atau pesanan yang siap ambil dan pesanan yang sudah diambil. *Activity diagram melihat data pesanan* bisa dilihat pada lampiran C.

d. *Activity diagram mengelola data pesanan*

*Activity diagram mengelola data pesanan* dilakukan oleh karyawan. *Activity diagram mengelola data pesanan* menjelaskan tentang bagaimana sistem dapat mengelola data pesanan, meliputi melihat data pesanan belum diproses, pesanan sedang diproses, pesanan sudah selesai, pesanan ditolak, pesanan sudah diambil dan konfirmasi pesanan diproses atau ditolak. *Activity diagram mengelola data pesanan* bisa dilihat pada lampiran C.

e. *Activity diagram melihat data outlet*

*Activity diagram* melihat data outlet dilakukan oleh *customer*. *Activity diagram* melihat data outlet menjelaskan tentang bagaimana sistem dapat melihat data outlet, seperti alamat dan status buka outlet. *Activity diagram* melihat data outlet bisa dilihat pada lampiran C.

f. *Activity diagram* mengelola data outlet

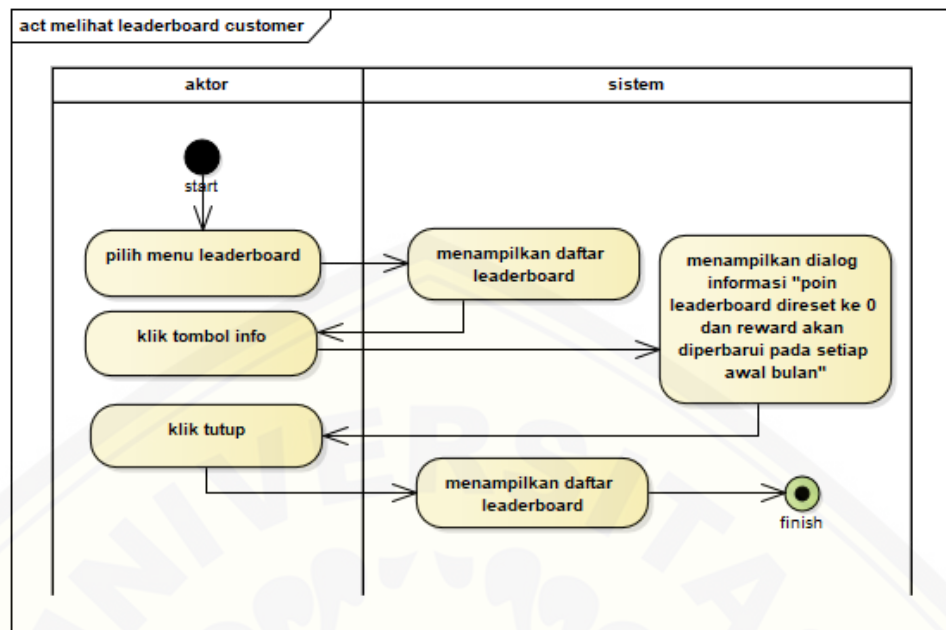
*Activity diagram* mengelola data outlet dilakukan oleh *admin*. *Activity diagram* mengelola data outlet menjelaskan tentang bagaimana sistem dapat mengelola data outlet, meliputi melihat, merubah dan menambah data outlet. *Activity diagram* mengelola data outlet bisa dilihat pada lampiran C.

g. *Activity diagram* mengelola data karyawan

*Activity diagram* mengelola data karyawan dilakukan oleh *admin*. *Activity diagram* mengelola data karyawan menjelaskan tentang bagaimana sistem dapat mengelola data karyawan, meliputi melihat, merubah, menambah dan menghapus data karyawan. *Activity diagram* mengelola data karyawan bisa dilihat pada lampiran C.

h. *Activity diagram* melihat *leaderboard*

*Activity diagram* melihat *leaderboard* dilakukan oleh *customer* dan *admin*. *Activity diagram* melihat *leaderboard* menjelaskan tentang bagaimana sistem dapat melihat urutan *customer* dari banyaknya poin yang sudah dikumpulkan pada bulan tertentu. *Activity diagram* melihat *leaderboard* bisa dilihat pada Gambar 4.6.



Gambar 4.6 Activity diagram Melihat Leaderboard

i. Activity diagram mengelola data leaderboard

Activity diagram mengelola data leaderboard dilakukan oleh admin. Activity diagram mengelola data leaderboard menjelaskan tentang bagaimana sistem dapat mengelola data leaderboard, meliputi merubah reward atau hadiah dan mengirimkan reward. Activity diagram mengelola data leaderboard bisa dilihat pada lampiran C.

j. Activity diagram melihat informasi customer

Activity diagram melihat informasi dilakukan oleh customer. Activity diagram melihat informasi menjelaskan tentang bagaimana sistem dapat melihat informasi yang diberikan oleh pihak Macarina. Activity diagram melihat informasi customer bisa dilihat pada lampiran C.

k. Activity diagram melihat informasi karyawan

Activity diagram melihat informasi dilakukan oleh karyawan. Activity diagram melihat informasi menjelaskan tentang bagaimana sistem dapat melihat informasi yang diberikan oleh pihak Macarina. Activity diagram melihat informasi karyawan bisa dilihat pada lampiran C.



l. *Activity diagram* mengelola informasi

*Activity diagram* mengelola informasi dilakukan oleh *admin*. *Activity diagram* mengelola informasi menjelaskan tentang bagaimana sistem dapat mengelola informasi, meliputi melihat, merubah, menghapus dan menambah informasi. *Activity diagram* mengelola informasi bisa dilihat pada lampiran C.

m. *Activity diagram* melihat data rasa *customer*

*Activity diagram* melihat data rasa dilakukan oleh *customer*. *Activity diagram* melihat data rasa menjelaskan tentang bagaimana sistem dapat melihat status rasa yang tersedia pada outlet tertentu. *Activity diagram* melihat data rasa *customer* bisa dilihat pada lampiran C.

n. *Activity diagram* melihat data rasa karyawan

*Activity diagram* melihat data rasa dilakukan oleh karyawan. *Activity diagram* melihat data rasa menjelaskan tentang bagaimana sistem dapat melihat dan merubah status persediaan rasa pada outlet tertentu. *Activity diagram* melihat data rasa karyawan bisa dilihat pada lampiran C.

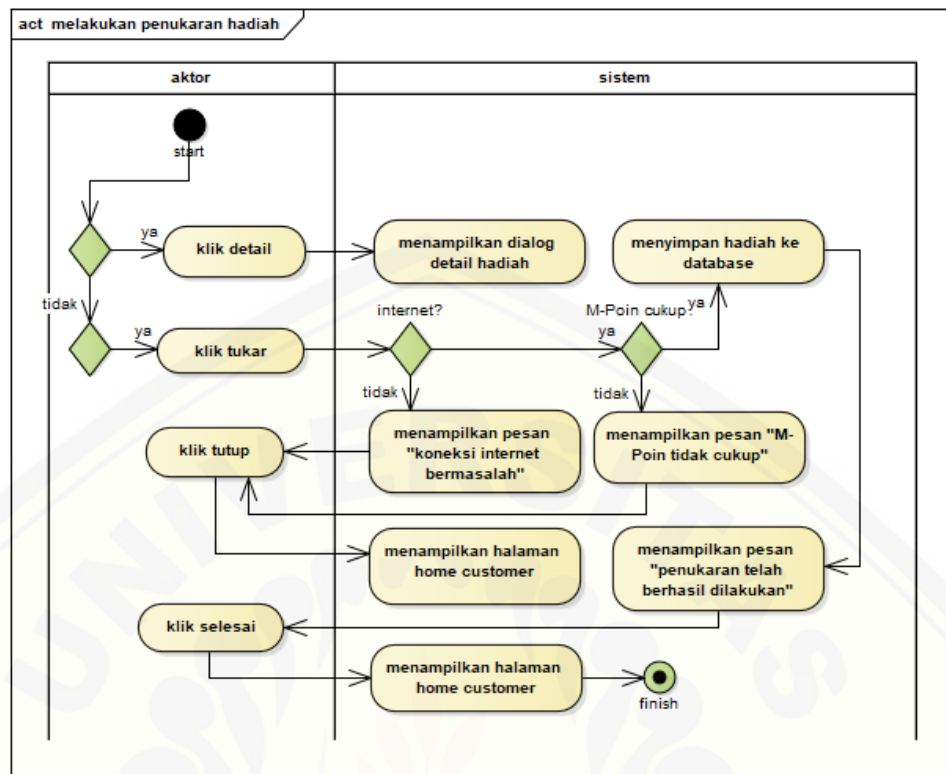
o. *Activity diagram* mengelola data rasa

*Activity diagram* mengelola data rasa dilakukan oleh *admin*. *Activity diagram* mengelola data rasa menjelaskan tentang bagaimana sistem dapat mengelola meliputi melihat, merubah status, menghapus dan menambah data rasa. *Activity diagram* mengelola data rasa bisa dilihat pada lampiran C.

p. *Activity diagram* melakukan penukaran hadiah

*Activity diagram* melakukan penukaran hadiah dilakukan oleh *customer*. *Activity diagram* melakukan penukaran hadiah menjelaskan tentang bagaimana sistem dapat melakukan penukaran hadiah menggunakan *M-Poin* yang dimiliki oleh *customer*. *Activity diagram* melakukan penukaran hadiah bisa dilihat pada Gambar 4.7.





Gambar 4.7 Activity diagram Melakukan Penukaran Hadiah

q. Activity diagram verifikasi hadiah

Activity diagram verifikasi hadiah dilakukan oleh *admin*. Activity diagram verifikasi hadiah menjelaskan tentang bagaimana sistem dapat melakukan verifikasi hadiah yang sudah dimiliki oleh *customer* sebelumnya. Activity diagram verifikasi hadiah bisa dilihat pada lampiran C.

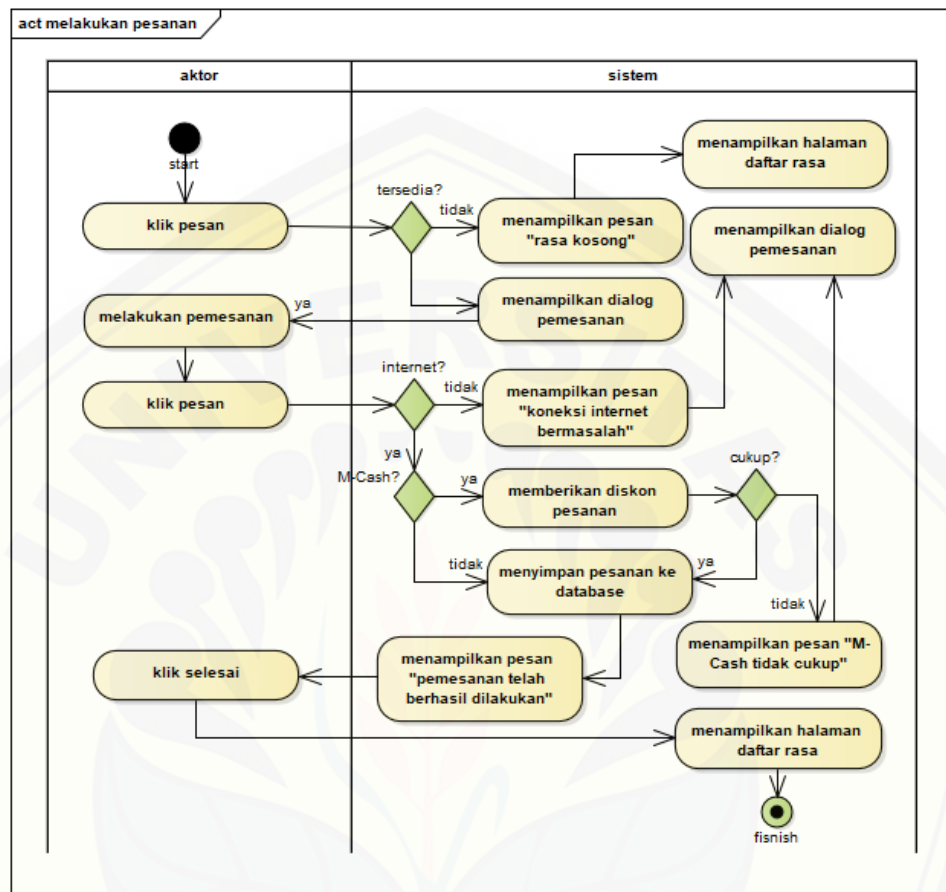
r. Activity diagram melihat data hadiah

Activity diagram melihat data hadiah dilakukan oleh *customer*. Activity diagram melihat data hadiah menjelaskan tentang bagaimana sistem dapat melihat hadiah yang sudah disediakan oleh pihak *Macarina*. Activity diagram melihat data hadiah bisa dilihat pada lampiran C.

s. Activity diagram melakukan pemesanan

Activity diagram melakukan pemesanan dilakukan oleh *customer*. Activity diagram melakukan pemesanan menjelaskan tentang bagaimana sistem dapat

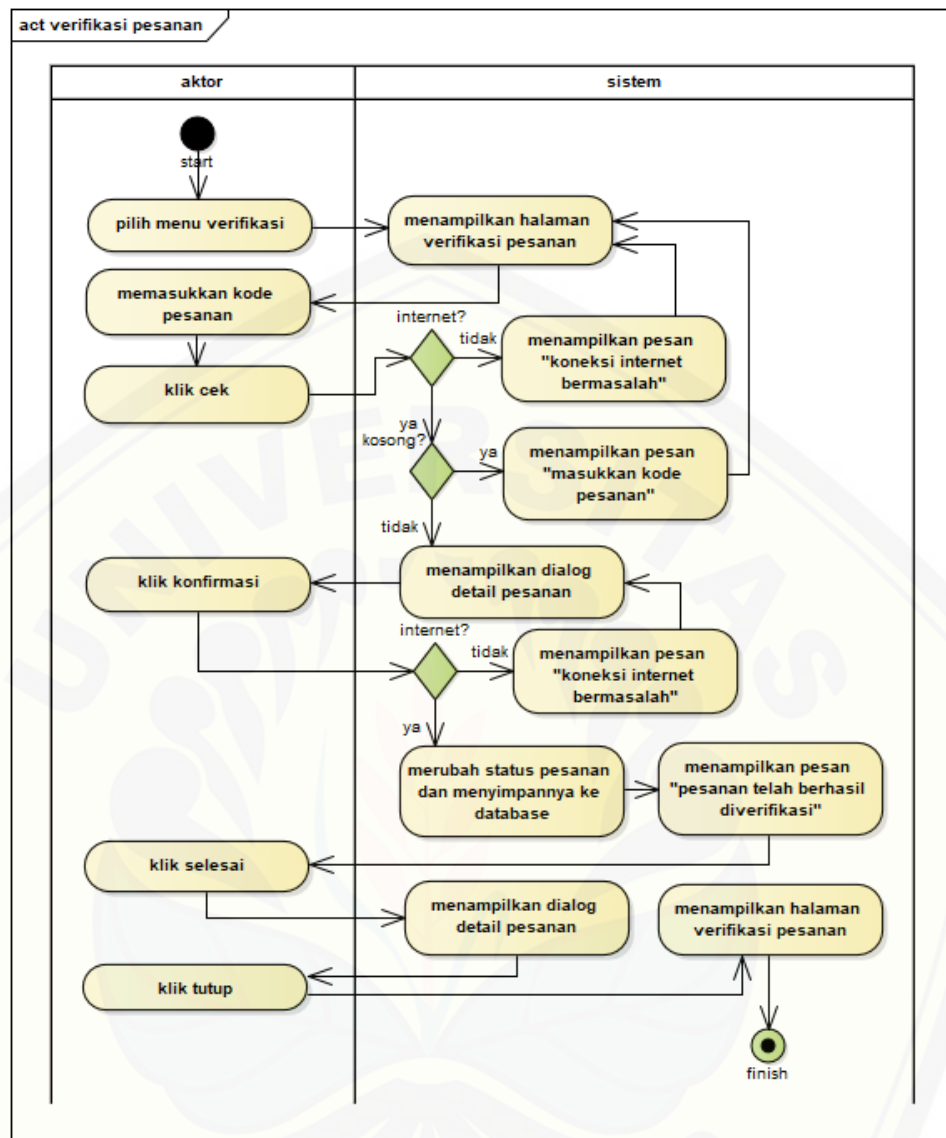
melakukan pemesanan produk Macarina. *Activity diagram* melakukan pemesanan bisa dilihat pada Gambar 4.8.



Gambar 4.8 *Activity diagram* Melakukan Pemesanan

t. *Activity diagram* verifikasi pesanan

*Activity diagram* verifikasi pesanan dilakukan oleh karyawan. *Activity diagram* verifikasi pesanan menjelaskan tentang bagaimana sistem dapat melakukan verifikasi pesanan yang sudah selesai diproses. *Activity diagram* verifikasi pesanan bisa dilihat pada Gambar 4.9.



Gambar 4.9 Activity diagram Verifikasi Pesanan

u. *Activity diagram* mengelola data hadiah

*Activity diagram* mengelola data hadiah dilakukan oleh *admin*. *Activity diagram* mengelola data hadiah menjelaskan tentang bagaimana sistem dapat mengelola data hadiah, meliputi melihat, merubah, menghapus dan menambah data hadiah. *Activity diagram* mengelola data hadiah bisa dilihat pada lampiran C.

v. *Activity diagram* melihat data *M-Cash*

*Activity diagram* melihat data *M-Cash* dilakukan oleh *customer*. *Activity diagram* melihat data *M-Cash* menjelaskan tentang bagaimana sistem dapat melihat

riwayat *top-up M-Cash* yang pernah dilakukan oleh *customer*. *Activity diagram* melihat data *M-Cash* bisa dilihat pada lampiran C.

w. *Activity diagram* mengelola data *M-Cash*

*Activity diagram* mengelola data *M-Cash* dilakukan oleh karyawan dan *admin*. *Activity diagram* mengelola data *M-Cash* menjelaskan tentang bagaimana sistem dapat melihat riwayat *top-up M-Cash* dan melakukan *top-up M-Cash* kepada *customer*. *Activity diagram* mengelola data *M-Cash* bisa dilihat pada lampiran C.

x. *Activity diagram* melihat data *M-Poin*

*Activity diagram* melihat data *M-Poin* dilakukan oleh *customer*. *Activity diagram* melihat data *M-Poin* menjelaskan tentang bagaimana sistem dapat melihat riwayat *M-Poin* yang sudah didapatkan oleh *customer* dari transaksi menggunakan aplikasi Macarina. *Activity diagram* melihat data *M-Poin* bisa dilihat pada lampiran C.

y. *Activity diagram* *logout*

*Activity diagram* *logout* dilakukan oleh *customer*, karyawan dan *admin*. *Activity diagram* *logout* menjelaskan tentang bagaimana sistem dapat melakukan keluar dari sistem oleh semua aktor. *Activity diagram* *logout* bisa dilihat pada lampiran C.

z. *Activity diagram* rasa favorit

*Activity diagram* rasa favorit dilakukan oleh *customer*. *Activity diagram* rasa favorit menjelaskan tentang bagaimana sistem dapat melihat rasa yang sering dipesan menggunakan aplikasi Macarina. *Activity diagram* rasa favorit bisa dilihat pada lampiran C.

#### 4.3.5. *Sequence Diagram*

*Sequence diagram* merupakan *diagram* yang menggambarkan interaksi antar kelas yang dilakukan sesuai dengan alurnya masing-masing. *Sequence diagram* merupakan desain perancangan yang mendekati proses pengkodean, maka dari itu di dalam *Sequence diagram* bahasa yang dituliskan merupakan bahasa pengkodean,

dimana bahasa pengkodean tersebut menjelaskan jalannya alur dalam fitur secara berurutan. *Sequence diagram* dalam sistem ini antara lain sebagai berikut:

a. *Sequence diagram login*

Interaksi antar *class* dan *function* pada *class login* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

b. *Sequence diagram registrasi*

Interaksi antar *class* dan *function* pada *class registrasi* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

c. *Sequence diagram melihat data pesanan*

Interaksi antar *class* dan *function* pada *class melihat data pesanan* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

d. *Sequence diagram mengelola data pesanan*

Interaksi antar *class* dan *function* pada *class mengelola data pesanan* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

e. *Sequence diagram melihat data outlet*

Interaksi antar *class* dan *function* pada *class melihat data outlet* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

f. *Sequence diagram mengelola data outlet*

Interaksi antar *class* dan *function* pada *class mengelola data outlet* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

g. *Sequence diagram mengelola data karyawan*

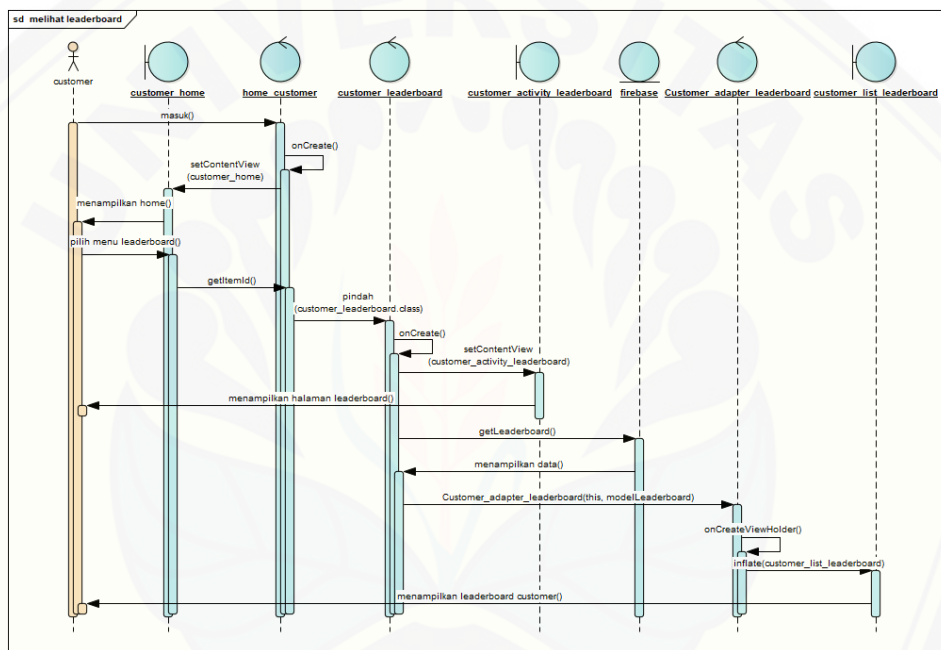
Interaksi antar *class* dan *function* pada *class mengelola data karyawan* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

h. *Sequence diagram melihat leaderboard*

Pada *sequence diagram melihat leaderboard* yang menjadi aktor adalah *customer*. Objek *customer\_home* akan menjalankan fungsi pindah untuk berpindah



ke halaman *customer\_leaderboard*, dan sekaligus pada halaman perancangan ini diterapkan elemen gamifikasi *leaderboard*. Kemudian objek *customer\_leaderboard* akan menjalankan fungsi *getLeaderboard()* yang berfungsi mengambil data *customer* dari *database* untuk ditaruh pada *customer\_adapter\_leaderboard* agar nantinya data tersebut dapat ditampilkan pada *customer\_leaderboard* untuk dilihat oleh *customer*. Interaksi antar *class* dan *function* pada *class* melihat *leaderboard* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada Gambar 4.10.



Gambar 4.10 *Sequence Diagram* Melihat *Leaderboard*

i. *Sequence diagram* mengelola data *leaderboard*

Interaksi antar *class* dan *function* pada *class* mengelola data *leaderboard* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

j. *Sequence diagram* melihat informasi *customer*

Interaksi antar *class* dan *function* pada *class* melihat informasi *customer* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

k. *Sequence diagram* melihat informasi karyawan



Interaksi antar *class* dan *function* pada *class* melihat informasi karyawan yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

l. *Sequence diagram* mengelola informasi

Interaksi antar *class* dan *function* pada *class* mengelola informasi yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

m. *Sequence diagram* melihat data rasa *customer*

Interaksi antar *class* dan *function* pada *class* melihat data rasa *customer* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

n. *Sequence diagram* melihat data rasa karyawan

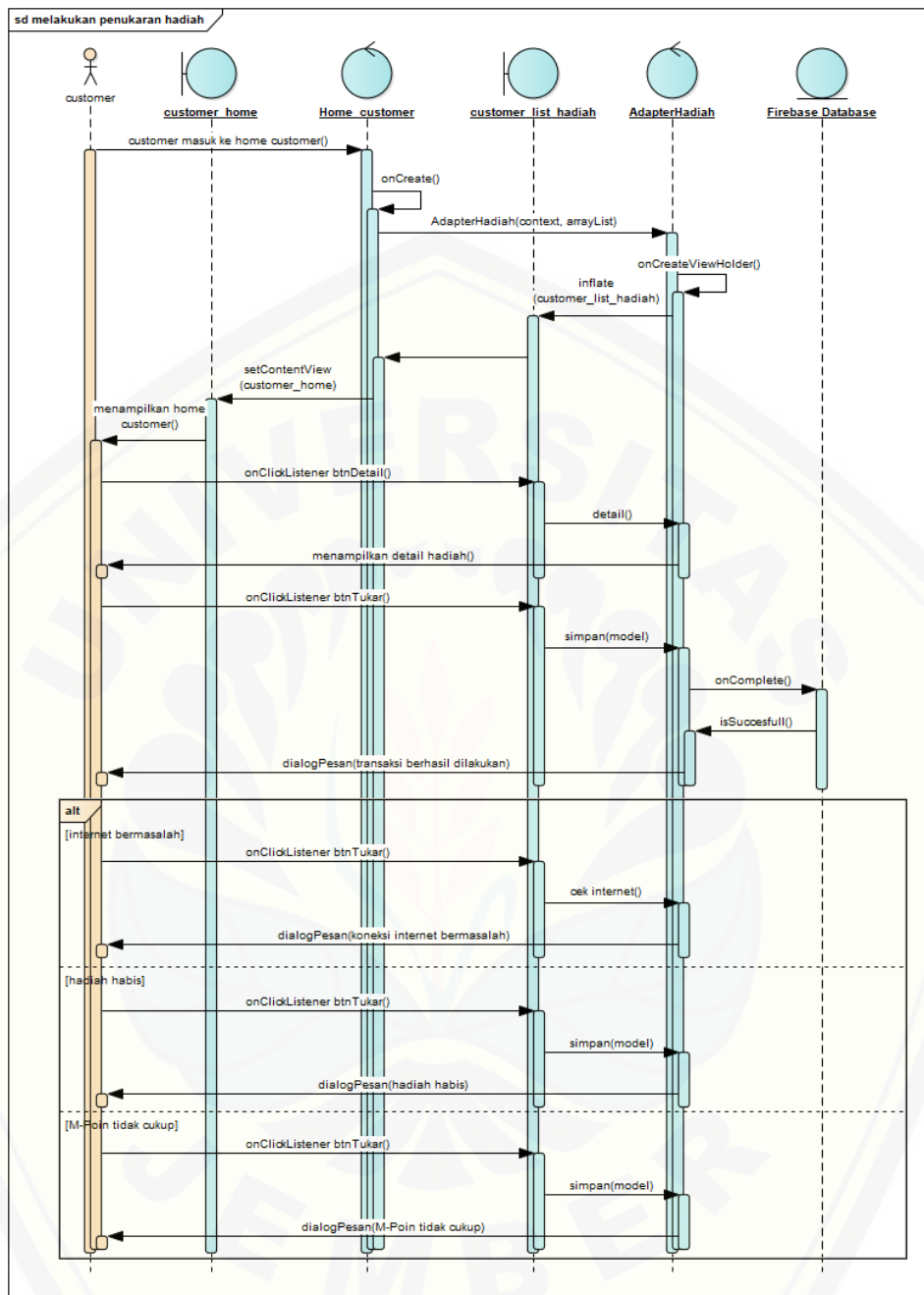
Interaksi antar *class* dan *function* pada *class* melihat data rasa karyawan yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

o. *Sequence diagram* mengelola data rasa

Interaksi antar *class* dan *function* pada *class* mengelola data rasa yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

p. *Sequence diagram* melakukan penukaran hadiah

Pada *sequence diagram* melakukan penukaran hadiah yang menjadi aktor adalah *customer*, dan sekaligus pada objek *AdapterHadiah()* ditempatkan elemen gamifikasi *points*. Objek *customer\_home* akan memanggil *class AdapterHadiah()* untuk menampilkan data hadiah agar dapat dilihat *customer*. Kemudian aktor akan memilih *btnTukar()* untuk menjalankan fungsi *simpan()* pada *AdapterHadiah()*, jika berhasil maka akan menjalankan fungsi *onComplete()* untuk pengiriman data ke *database*. Interaksi antar *class* dan *function* pada *class* melakukan penukaran hadiah yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada Gambar 4.11.



Gambar 4.11 *Sequence Diagram* Melakukan Penukaran Hadiah

q. *Sequence diagram* verifikasi hadiah

Interaksi antar *class* dan *function* pada *class* verifikasi hadiah yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

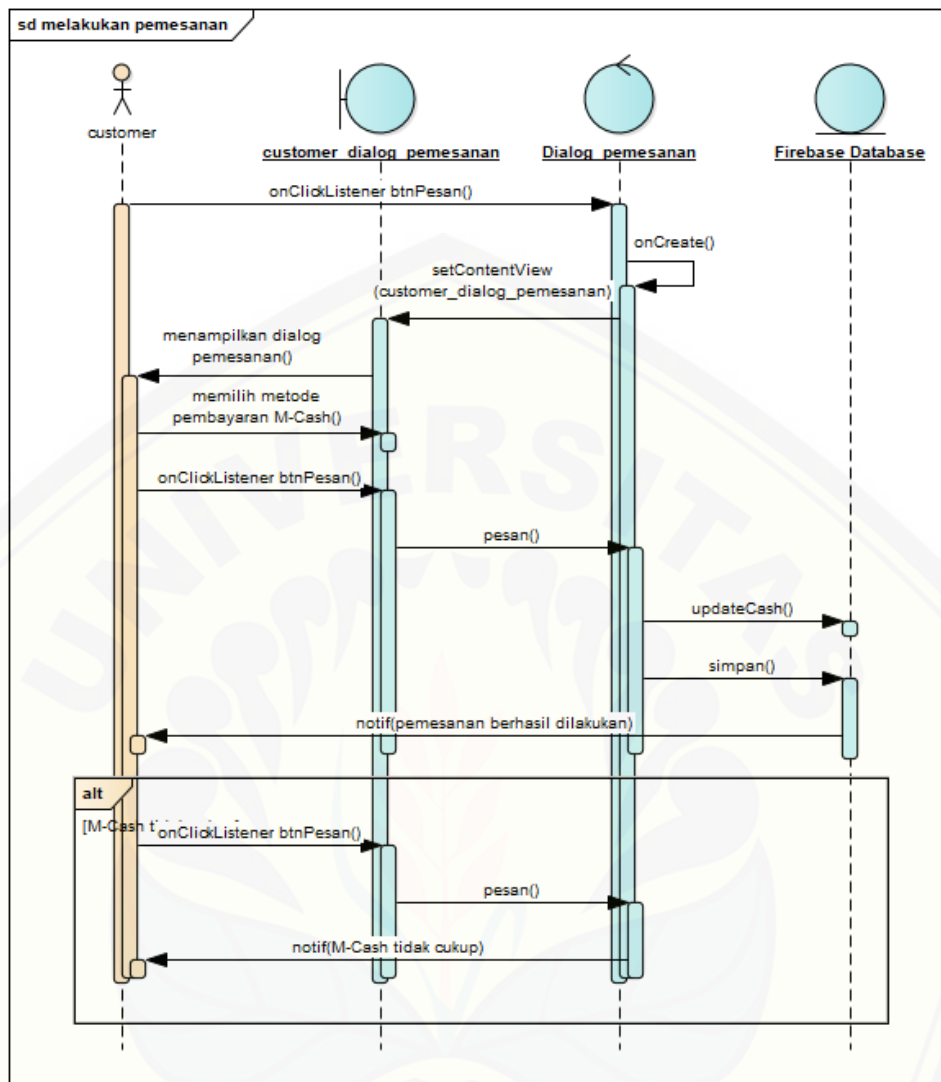
r. *Sequence diagram* melihat data hadiah

Interaksi antar *class* dan *function* pada *class* melihat data hadiah yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

s. *Sequence diagram* melakukan pemesanan

Pada *sequence diagram* melakukan pemesanan yang menjadi aktor adalah *customer*, dan sekaligus pada objek *dialog\_pemesanan* ditempatkan elemen gamifikasi *points* dan *reward*. objek *dialog\_pemesanan* akan menjalankan fungsi *simpan()* ketika aktor memilih *btnPesan()*. Setelah itu akan dilanjutkan dengan menjalankan fungsi *updateCash()* dan *simpan()* ke *database* untuk melakukan pembaruan *M-Cash* yang dimiliki oleh aktor dan sekaligus melakukan penyimpanan data pesanan baru milik aktor.

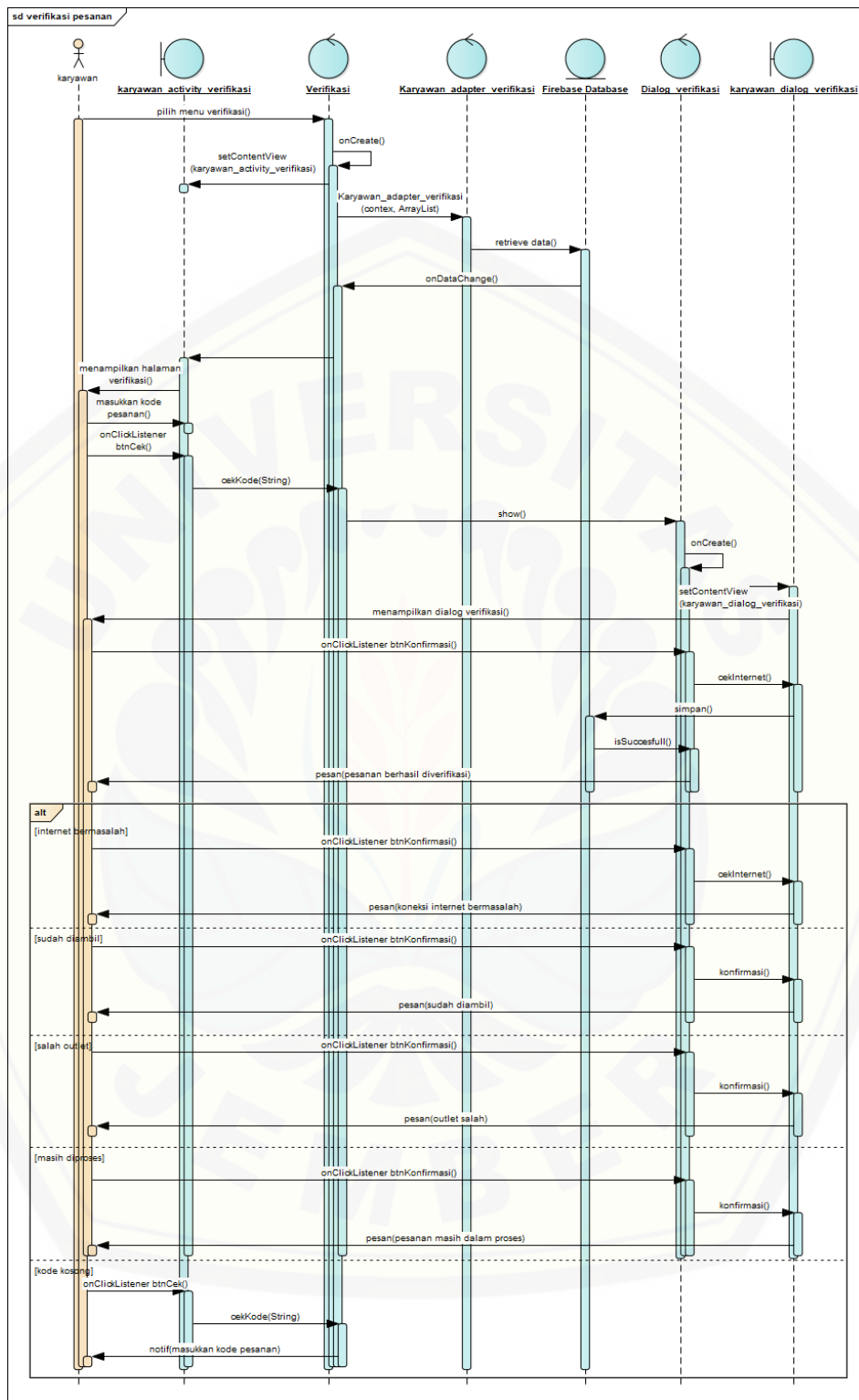
Interaksi antar *class* dan *function* pada *class* melakukan pemesanan yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada Gambar 4.12.



Gambar 4.12 Sequence Diagram Melakukan Pemesanan

t. Sequence diagram verifikasi pesanan

Pada *sequence diagram* verifikasi pesanan yang menjadi aktor adalah karyawan. Aktor akan memasukkan kode pesanan, kemudian akan dilakukan pengecekan oleh objek *verifikasi* terhadap kode tersebut. Jika ada, maka akan menjalankan fungsi *show()* untuk menampilkan *dialog\_verifikasi*. Lalu aktor akan memilih *btnKonfirmasi()* untuk menjalankan fungsi *simpan()* pada *dialog\_verifikasi* yang berfungsi untuk melakukan pembaruan data pesanan pada *database* berdasarkan kode pesanan tersebut. Interaksi antar *class* dan *function* pada *class* verifikasi pesanan yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada Gambar 4.13.



Gambar 4.13 Sequence Diagram Verifikasi Pesanan

u. Sequence diagram mengelola data hadiah

Interaksi antar *class* dan *function* pada *class* mengelola data hadiah yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

v. *Sequence diagram* melihat data *M-Cash*

Interaksi antar *class* dan *function* pada *class* melihat data *M-Cash* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

w. *Sequence diagram* mengelola data *M-Cash*

Interaksi antar *class* dan *function* pada *class* mengelola data *M-Cash* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

x. *Sequence diagram* melihat data *M-Poin*

Interaksi antar *class* dan *function* pada *class* melihat data *M-Poin* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

y. *Sequence diagram* logout

Interaksi antar *class* dan *function* pada *class* logout yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

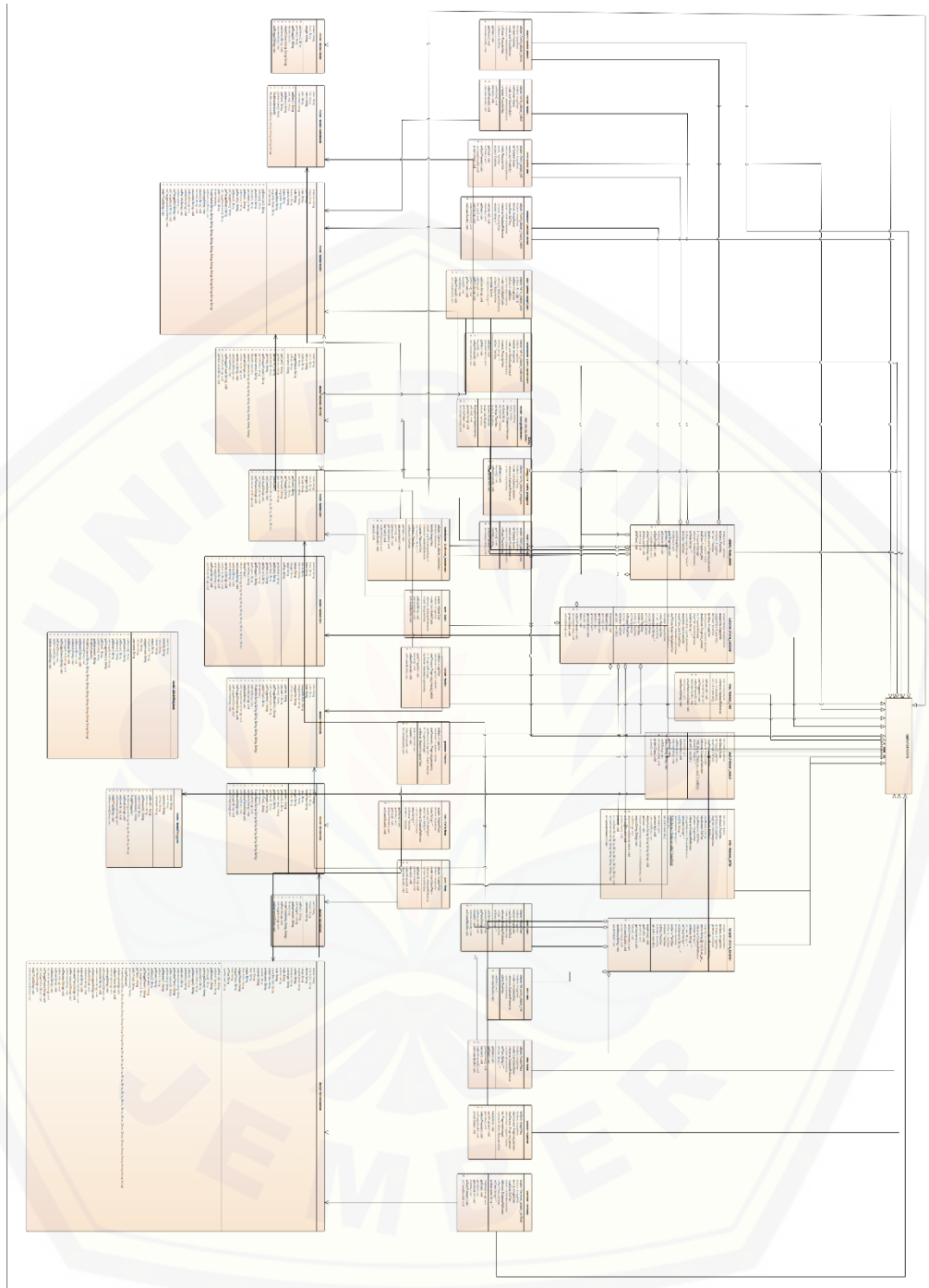
z. *Sequence diagram* rasa favorit

Interaksi antar *class* dan *function* pada *class* rasa favorit yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B.

#### 4.3.6. Class Diagram

*Class diagram* merupakan diagram untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya digunakan. *Class diagram* yang digunakan pada aplikasi *e-commerce* berbasis *mobile apps* dapat dilihat pada Gambar 4.14.





Gambar 4.14 *Class Diagram*

Untuk *class diagram* lebih lengkapnya dapat dilihat pada Lampiran G.

#### 4.3.7. Database Structure

Pada penelitian ini Struktur penyimpanan data pada aplikasi Macarina tidak menggunakan ERD (*Entity Relationship Diagram*) dikarenakan untuk penyimpanan data menggunakan *JSON Object*, maka dari itu peneliti menggunakan *JSON Tree* untuk struktur penyimpanan datanya. Struktur penyimpanan data pada penelitian ini adalah sebagai berikut:

```
Aplikasi Macarina: {
  "dataCash": {
    "push": {
      "cash": "String",
      "harga": "String",
      "tanggal": "String",
      "tempat": "String",
      "username": "String"
    }
  },
  "dataCustomer": {
    "username": {
      "cash": "String",
      "nama": "String",
      "noUrut": "String",
      "nomor": "String",
      "tanggalMasuk": "String",
      "urlFoto": "String",
      "username": "String"
    }
  },
  "dataDiskon": {
    "kode": {
      "diskon": "String",
      "kode": "String",
      "tanggal": "String"
    }
  },
  "dataHadiah": {
    "kode": {
      "deskripsi": "String",
      "jenis": "String",
      "judul": "String",
      "harga": "String",
```

```
        "jumlah": "String",
        "kode": "String",
        "kondisi": "String",
        "pemilik": "String",
        "status": "String",
        "tanggalBeli": "String",
        "tanggalBuat": "String",
        "tanggalVerifikasi": "String",
        "urlFoto": "String"
    }
},
"dataHadiahCustomer": {
    "kode": {
        "deskripsi": "String",
        "jenis": "String",
        "judul": "String",
        "harga": "String",
        "jumlah": "String",
        "kode": "String",
        "kondisi": "String",
        "pemilik": "String",
        "status": "String",
        "tanggalBeli": "String",
        "tanggalBuat": "String",
        "tanggalVerifikasi": "String",
        "urlFoto": "String"
    }
},
"dataInfo": {
    "kode": {
        "info": "String",
        "kode": "String",
        "judul": "String",
        "tanggalBerakhir": "String",
        "tanggalInfo": "String",
        "tanggalDibuat": "String",
        "urlFoto": "String"
    }
},
"dataKaryawan": {
    "username": {
        "alamat": "String",
```

```
        "kelamin": "String",
        "nama": "String",
        "nomor": "String",
        "status": "String",
        "tanggalLahir": "String",
        "tanggalMasuk": "String",
        "toko": "String",
        "urlFoto": "String",
        "username": "String"
    },
    "dataLeaderboard": {
        "username": {
            "desc": "String",
            "foto": "String",
            "nama": "String",
            "poin": "String",
            "username": "String"
        }
    },
    "dataMacarina": {
        "leaderboard": {
            "leaderboard": "String"
        },
        "rasa": {
            "kode": "String",
            "kondisi": "String",
            "rasa": "String",
            "status": "String",
            "tanggal": "String",
            "urlFoto": "String"
        },
        "toko": {
            "diskon": "String",
            "urutan": "String"
        }
    },
    "dataPengguna": {
        "UID": {
            "email": "String",
            "password": "String",
            "urutan": "String",
```

```
        "level": "String",
        "username": "String"
    }
},
"dataPesanan": {
    "kode": {
        "urlFoto": "String",
        "kode": "String",
        "toko": "String",
        "userPembeli": "String",
        "pembeli": "String",
        "nomorPembeli": "String",
        "nomorToko": "String",
        "rasa": "String",
        "alamat": "String",
        "kemasan": "String",
        "level": "String",
        "jumlah": "String",
        "pembayaran": "String",
        "diskon": "String",
        "harga": "String",
        "hargaDiskon": "String",
        "poin": "String",
        "status": "String",
        "kondisi": "String",
        "tanggalPesanan": "String",
        "tanggalTukar": "String",
        "bulanTukar": "String",
        "tahunTukar": "String",
        "keterangan": "String",
        "pesanan": "String"
    }
},
"dataPoin": {
    "push": {
        "tanggal": "String",
        "poin": "String",
        "username": "String"
    }
},
"dataRasa": {
    "kode": {
```

```
        "urlFoto": "String",
        "kode": "String",
        "rasa": "String",
        "status": "String",
        "kondisi": "String",
        "tanggal": "String",
        "namaToko": "String"
    }
},
"dataToko": {
    "kode": {
        "kode": "String",
        "urlFoto": "String",
        "nama": "String",
        "alamat": "String",
        "status": "String",
        "buka": "String",
        "tutup": "String",
        "nomor": "String",
        "tanggal": "String"
    }
},
"dataFavorit": {
    "kode": {
        "desc": "String",
        "kode": "String",
        "rasa": "String",
        "terjual": "String",
        "url": "String"
    }
}
}
```

#### 4.4. Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *black box*.



#### 4.4.1. Pengujian *Black Box*

Pengujian *black box* berfungsi untuk menguji sistem dari sisi fungsionalitas sistem dengan tujuan mengetahui apakah fitur-fitur, inputan dan keluaran sistem sesuai dengan kebutuhan pengguna. Fitur-fitur yang diujikan mencakup fitur melihat, menambah, mengubah dan menghapus data serta keluaran yang diharapkan dari sistem saat fitur tersebut dijalankan. Hasil pengujian *black box* pada fitur dapat dilihat pada tabel berikut ini:

a. *Login*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur *login* dapat dilihat pada lampiran D.

b. Registrasi

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur registrasi dapat dilihat pada lampiran D.

c. Melihat data pesanan

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat data pesanan dapat dilihat pada lampiran D.

d. Mengelola data pesanan

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur mengelola data pesanan dapat dilihat pada lampiran D.

e. Melihat data *outlet*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat data *outlet* dapat dilihat pada lampiran D.

f. Mengelola data *outlet*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur mengelola data *outlet* dapat dilihat pada lampiran D.

## g. Mengelola data karyawan

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur mengelola data karyawan dapat dilihat pada lampiran D.

h. Melihat *leaderboard*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat *leaderboard* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Melihat *Leaderboard*

No. <i>Use case</i>	UC09		
Nama <i>use case</i>	Melihat <i>leaderboard</i>		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> memilih menu <i>leaderboard</i>		
PostKondisi	<i>Customer</i> berhasil melihat daftar <i>leaderboard</i> dan melihat <i>reward</i> yang didapat		
Aliran Normal Melihat daftar <i>leaderboard</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	2. Menampilkan <i>reward</i> dan daftar <i>leaderboard</i> .	√	

i. Mengelola data *leaderboard*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur mengelola data *leaderboard* dapat dilihat pada lampiran D.

j. Melihat informasi *customer*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat informasi *customer* dapat dilihat pada lampiran D.

k. Melihat informasi karyawan

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat informasi *customer* dapat dilihat pada lampiran D.

l. Mengelola informasi

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur mengelola informasi dapat dilihat pada lampiran D.

m. Melihat data rasa *customer*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat data rasa *customer* dapat dilihat pada lampiran D.

n. Melihat data rasa karyawan

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat data rasa karyawan dapat dilihat pada lampiran D.

o. Mengelola data rasa

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur mengelola data rasa dapat dilihat pada lampiran D.

## p. Melakukan penukaran hadiah

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melakukan penukaran hadiah dapat dilihat pada Tabel 4.6.

Tabel 4.6 Melakukan Penukaran Hadiah

No. Use case	UC17		
Nama use case	Melakukan penukaran hadiah		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> berada pada halaman <i>home customer</i>		
PostKondisi	<i>Customer</i> berhasil melakukan penukaran hadiah menggunakan <i>M-Poin</i>		
Aliran Normal Menukar hadiah menggunakan <i>M-Poin</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
5. Klik tombol tukar			
	6. Sistem menyimpan hadiah ke database	√	
	7. Menampilkan pesan “penukaran hadiah telah berhasil dilakukan”	√	
	8. Menampilkan halaman <i>home customer</i>	√	
Aliran Alternatif <i>M-Poin</i> kurang			
5. Klik tombol tukar			
	6. Sistem mengecek <i>M-Poin customer</i>	√	
	7. Menampilkan pesan “ <i>M-Poin</i> tidak cukup”	√	

	8. Menampilkan halaman <i>home customer</i>	√	
--	---	---	--

q. Verifikasi hadiah

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur verifikasi hadiah dapat dilihat pada lampiran D.

r. Melihat data hadiah

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat data hadiah dapat dilihat pada lampiran D.

s. Melakukan pemesanan

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melakukan pemesanan dapat dilihat pada Tabel 4.7.

Tabel 4.7 Melakukan Pemesanan

No. Use case	UC19		
Nama use case	Melakukan pemesanan		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> melihat persediaan rasa pada <i>outlet</i>		
PostKondisi	<i>Customer</i> berhasil melakukan pemesanan		
Aliran Normal Pemesanan produk			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
8. Klik tombol pesan			
	9. Menampilkan dialog pemesanan	√	
10. Menentukan pembayaran, jumlah, kemasan dan level			

11. Klik pesan			
	12. Sistem menyimpan data pesanan ke database	√	
	13. Menampilkan pesan “pemesanan telah berhasil dilakukan”	√	
	14. Menampilkan halaman daftar rasa	√	
Aliran Alternatif Status rasa kosong			
8. Klik tombol pesan			
	9. Sistem mengecek status persediaan	√	
	10. Menampilkan pesan “rasa kosong”	√	
	11. Menampilkan halaman daftar rasa	√	
Aliran Alternatif <i>M-Cash</i> tidak cukup			
5. Memilih metode pembayaran <i>M-Cash</i>			
6. Klik pesan			
	12. Sistem mengecek <i>M-Cash customer</i>	√	
	13. Menampilkan pesan “ <i>M-Cash</i> tidak cukup”	√	
	14. Menampilkan dialog pemesanan	√	
Aliran Alternatif Koneksi internet bermasalah			
8. Klik pesan			
	9. Sistem mengecek koneksi internet	√	



	10. Menampilkan pesan “koneksi internet bermasalah”	√	
	11. Menampilkan dialog pemesanan	√	

t. Verifikasi pesanan

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur verifikasi pesanan dapat dilihat pada Tabel 4.8.

Tabel 4.8 Verifikasi Pesanan

No. <i>Use case</i>	UC24		
Nama <i>use case</i>	Verifikasi pesanan		
Aktor	Karyawan		
PreKondisi	Karyawan pilih menu verifikasi		
PostKondisi	Karyawan berhasil verifikasi pesanan		
Aliran Normal Verifikasi pesanan			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Masukkan kode pesanan			
2. Klik cek			
	3. Menampilkan dialog detail pesanan	√	
4. Klik konfirmasi			
	5. Merubah status pesanan dan menyimpannya ke database	√	
	6. Menampilkan pesan “pesanan	√	

	berhasil diverifikasi”		
Aliran Alternatif Kode kosong			
2. Klik cek			
	3. mengecek kode pesanan	√	
	4. menampilkan pesan “masukkan kode pesanan”	√	
Aliran Alternatif Internet bermasalah			
4. klik konfirmasi			
	5. mengecek koneksi internet	√	
	6. menampilkan pesan “koneksi internet bermasalah”	√	
Aliran Alternatif Salah <i>outlet</i>			
4. klik konfirmasi			
	5. mengecek <i>outlet</i>	√	
	6. menampilkan pesan “salat <i>outlet</i> ”	√	
Aliran Alternatif Masih diproses			
4. klik konfirmasi			
	5. mengecek pesanan	√	
	6. menampilkan pesan “masih diproses”	√	
Aliran Alternatif Sudah diambil			

4. klik konfirmasi			
	5. mengecek pesanan	√	
	6. menampilkan pesan “pesanan sudah diambil”	√	

u. Mengelola data hadiah

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur mengelola data hadiah dapat dilihat pada lampiran D.

v. Melihat data *M-Cash*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat data *M-Cash* dapat dilihat pada lampiran D.

w. Mengelola data *M-Cash*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur mengelola data *M-Cash* dapat dilihat pada lampiran D.

x. Melihat data *M-Poin*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur melihat data *M-Poin* dapat dilihat pada lampiran D.

y. *Logout*

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur *logout* dapat dilihat pada lampiran D.

z. Rasa favorit

Alur dari penjelasan use case skenario setelah di evaluasi apakah sesuai dengan tampilan luar atau interface-nya. Detail dari pengujian *black box* pada fitur rasa favorit dapat dilihat pada lampiran D.



## BAB 6. PENUTUP

### 6.1. Kesimpulan

Hasil dari penelitian yang dilakukan dapat disimpulkan sebagai berikut:

1. Peneliti telah berhasil membuat sebuah aplikasi *e-commerce* berbasis *mobile apps* yang diterapkan pada smartphone dengan operasi sistem Android. Dan juga berhasil menerapkan metode gamifikasi pada aplikasi tersebut.
2. Aplikasi mampu memberikan informasi pasti mengenai outlet (status buka, alamat dan nomor yang dapat dihubungi), rasa (status rasa yang tersedia pada outlet tertentu), pesanan (mulai dari pesanan yang belum diproses hingga pesanan sudah siap ambil), informasi dari pihak Macarina, hadiah (persediaan beserta harga *M-Poin*) dan juga penampilan rasa favorit yang akan memudahkan *customer* baru untuk memilih rasa produk yang sering dipesan. Informasi outlet dan rasa itu paling penting, dikarenakan *customer* tidak perlu lagi pergi mengecek ke tiap outlet untuk mengetahui status buka outlet ataupun ketersediaan rasa pada outlet tertentu.
3. Aplikasi *e-commerce* Macarina berbasis *mobile app* ini dibuat dengan cara menerapkan 3 elemen gamifikasi didalamnya, yakni *points*, *reward* dan *leaderboard*.
4. Penerapan elemen-elemen gamifikasi pada Aplikasi Macarina ditempatkan pada beberapa fitur, seperti perangsingan *customer* dengan diterapkannya elemen *leaderboard* dan *reward*, perangsingan rasa favorit dengan diterapkannya elemen *leaderboard*, pemesanan menggunakan *M-Cash* dengan diterapkannya elemen *points* dan *reward* dan juga penukaran hadiah menggunakan *M-Poin* dengan diterapkannya elemen *points*.

5. Aplikasi *e-commerce* berbasis *mobile apps* dengan tiga aktor yaitu *admin*, karyawan dan *customer*. Aktor *admin* untuk mengelola sistem, aktor karyawan untuk pegawai macarina pada tiap outlet dan *customer* sebagai pembeli sekaligus spesifikasi pada metode. Desain sistem yang dibuat yaitu *input and output diagram*, *use case diagram*, *scenario*, *sequence diagram*, *activity diagram*, *class diagram* dan *database structure*.

## 6.2. Saran

1. Penambahan keamanan pada *M-Cash*, karena *M-Cash* merupakan data pribadi yang sangat urgen bagi *customer*.
2. Penambahan metode *login* menggunakan akun sosial media, seperti: *facebook*, *instagram*, *twitter*, *gmail*, dll. Untk mempermudah keaslian akun *customer*.
3. Validasi nomor telepon untuk pengecekan nomor pribadi.
4. Penelitian lanjutan lebih difokuskan terhadap seberapa efektif dan efisiennya penerapan elemen-elemen gamifikasi dalam menarik pelanggan.
5. Tidak digunakannya aspek lainnya bukan karena aspek tersebut tidak bisa diimplentasikan pada macarina melainkan perlunya observasi lebih lanjut untuk mengetahui sesuai atau tidaknya aspek gamifikasi yang akan ditambahkan kembali. Usaha Mikro Kecil dan Menengah (UMKM) lainnya juga dapat menggunakan aspek aspek gamifikasi yang diterapkan pada penelitian ini atau bahkan mengembangkan kembali aspek yang ada sehingga kemajuan UMKM dapat lebih cepat dan sesuai target yang diharapkan.
6. Penambahan algoritma pada perolehan poin. Hal ini ditujukan agar tidak ada yang dirugikan antar kedua belah pihak.



## DAFTAR PUSTAKA

- Aribowo, E. K. (2017). Gamification: Adaptasi Game dalam Dunia Pendidikan. 8-10.
- Azizah, N., Mahendra, D., & Lofian, B. (2019). Pemanfaatan E-Commerce untuk Peningkatan Strategi Promosi dan Penjualan UMKM Tas di Kamupaten Kudus. *Jurnal Pengabdian kepada Masyarakat*, 4.
- Aini, Qurotul., Rahardja, Untung., Moeins, Anoesyirwan., Apriani, Dewi Mariana. (2018). Penerapan Gamifikasi pada Sistem Informasi Penilaian Ujian Mahasiswa untuk Meningkatkan Kinerja Dosen. *SYNTAX Jurnal Informatika*. 7(1): 33-43.
- Takdir, Muhammad. (2017). Penerapan Konsep Gamifikasi dalam Pembelajaran Matematika dalam Meningkatkan Motivasi Belajar Matematika Siswa. *Jurnal Penelitian Pendidikan INSANI*. 20(1): 1-6.
- Sary, Bety Wulan., Utami, Ema., Al-Fatta, Hanif. (2015). Penerapan Konsep Gamification pada Pembelajaran Tenses Bahasa Inggris Berbasis Web. *Jurnal Ilmiah SISFOTENIKA*. 5(2): 155-166.
- Tamara, Dony., Neyfa, Bella Chintya. (2016). Rancang Bangun Aplikasi E-Learning Moonlay Academy Dengan Metode Gamifikasi Dan Algoritma Knuth Shuffleperancangan Aplikasi E-Canteen Berbasis Android Dengan Menggunakan Metode Object Oriented Analysis & Design (OOAD). *Jurnal Penelitian Komunikasi dan Opini Publik*. 20(1): 89-93.
- Setyowati, Lis. (2017). Pengajaran Literasi Informasi Dengan Konsep Gamification Di Perpustakaan Perguruan Tinggi. *Jurnal Perpustakaan*. 8(1): 39-50.
- CLSA Indonesia, P. (2016, Februari 11). *33% Pengguna Smartphone Akses Perangkatnya Lebih Dari 8 Jam/hari*. Retrieved from [databoks.katadata.co.id: https://databoks.katadata.co.id/datapublish/2016/11/02/33-pengguna-smartphone-akses-perangkatnya-lebih-dari-8-jam-per-hari](https://databoks.katadata.co.id/datapublish/2016/11/02/33-pengguna-smartphone-akses-perangkatnya-lebih-dari-8-jam-per-hari)

Wiryaputra, Samuel., Hansun, Seng., Wiratama, Yustinus Widya. (2016). Rancang Bangun Aplikasi E-Learning Moonlay Academy dengan Metode Gamifikasi Dan Algoritma Knuth Shuffle. *Jurnal Teknik dan Ilmu Komputer*. 5(19): 305-317.

eMarketer. (2016, Agustus 8). *Pengguna Smartphone di Indonesia 2016-2019*. Dikutip pada 11 Maret 2019 dari databoks.katadata.co.id: <https://databoks.katadata.co.id/datapublish/2016/08/08/pengguna-smartphone-di-indonesia-2016-2019>

Kardianawati, A., Haryanto, H., & Rosyidah, U. (2016). Penerapan Konsep Gamifikasi Appreciative pada EMarketplace UMKM. 4.

Nurdiansyah, Y., Wijayanto, F., & Firdaus. (2018). The Design of E-Commerce in the Shrimp Paste Industry using the Method of Structured Analysis and Design Technique (SADT) to Increase Marketing. *MATEC Web of Conferences*, 7.

Haryanto, Agus Tri. (2018, 19 September). *GoPoints, Cara Go-Jek Tingkatkan Loyalitas Pengguna*. Dikutip pada 11 Maret 2019 dari Detik Inet: <https://inet.detik.com/cyberlife/d-4219455/Go-Points-cara-Go-Jek-tingkatkan-loyalitas-pengguna>

Simamarta (2006). *Aplikasi Mobile Commerce Menggunakan PHP dan MySQL*

## LAMPIRAN

### A. Scenario

#### A.1. Login

No. Use case	UC01
Nama use case	<i>Login</i>
Aktor	<i>Customer, admin, karyawan</i>
PreKondisi	<i>User telah memiliki email dan password dulu agar bisa login</i>
PostKondisi	<i>User berhasil login sesuai hak akses</i>
Aliran Normal <i>login</i>	
Aksi Aktor	Reaksi Sistem
1. Masukkan <i>email</i> dan <i>password</i>	
2. Klik tombol masuk	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan halaman <i>home</i> sesuai dengan hak akses
Aliran Alternatif Tidak ada koneksi internet	
2. Klik tombol masuk	
	3. Sistem mengecek koneksi internet
	4. Menampilkan pesan “Koneksi internet Anda bermasalah, mohon untuk mengeceknya kembali”
	5. Menampilkan halaman <i>login</i>
Aliran Alternatif <i>email</i> dan <i>password</i> kosong	
2. Klik tombol masuk	
	3. Sistem mengecek <i>email</i> dan <i>password</i>

	4. Menampilkan pesan “ <i>email dan password</i> tidak boleh kosong”
	5. Menampilkan halaman <i>login</i>
Aliran Alternatif <i>email</i> atau <i>password</i> kosong	
2. Klik tombol masuk	
	3. Sistem mengecek <i>email</i> atau <i>password</i>
	4. Menampilkan pesan “ <i>email/password</i> tidak boleh kosong”
	5. Menampilkan halaman <i>login</i>
Aliran Alternatif <i>email</i> tidak valid	
2. Klik tombol masuk	
	3. Sistem mengecek validasi <i>email</i>
	4. Menampilkan pesan “ <i>email</i> tidak valid”
	5. Menampilkan halaman <i>login</i>

## A.2. Registrasi akun

No. <i>Use case</i>	UC02
Nama <i>use case</i>	Registrasi akun
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> belum mempunyai akun untuk bisa <i>login</i> ke dalam sistem
PostKondisi	<i>Customer</i> berhasil membuat akun dan akan diarahkan ke halaman <i>home customer</i>
Aliran Normal Proses <i>login</i> sistem	
Aksi Aktor	Reaksi Sistem
1. Pilih foto dan masukkan - Nama, <i>email</i> , <i>password</i> , nomor hp	

2. Klik tombol daftar	
	3. Sistem mengecek foto, nama, <i>email</i> , <i>password</i> dan nomor hp
	4. Menampilkan halaman <i>home customer</i>
Aliran Alternatif Tidak ada koneksi internet	
2. Klik tombol daftar	
	3. Sistem mengecek koneksi internet
	4. Menampilkan pesan “koneksi internet bermasalah”
	5. Menampilkan halaman daftar
Aliran Alternatif Belum memilih foto	
2. Klik tombol daftar	
	3. Sistem mengecek url foto
	4. Menampilkan pesan “Pilih foto terlebih dahulu”
	5. Menampilkan halaman daftar
Aliran Alternatif Nama, <i>email</i> , <i>password</i> atau nomor hp kosong	
2. Klik tombol daftar	
	3. Mengecek nama, <i>email</i> , <i>password</i> atau nomor hp
	4. Menampilkan pesan “Data tidak boleh kosong”
	5. Menampilkan halaman daftar
Aliran Alternatif Panjang karakter <i>password</i>	
2. Klik tombol daftar	
	3. Mengecek panjang karakter <i>password</i>
	4. Menampilkan pesan “ <i>password</i> min 6 karakter”

	5. Menampilkan halaman daftar
Aliran Alternatif Validasi nomor hp	
2. Klik tombol daftar	
	3. Sistem mengecek nomor hp
	4. Menampilkan pesan “nomor hp tidak valid”
	5. Menampilkan halaman daftar
Aliran Alternatif Validasi <i>email</i>	
2. Klik tombol daftar	
	6. Sistem mengecek <i>email</i>
	7. Menampilkan pesan “ <i>email</i> tidak valid”
	8. Menampilkan halaman daftar

### A.3. Melihat data pesanan

No. <i>Use case</i>	UC03
Nama <i>use case</i>	Melihat data pesanan
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> memilih menu pesanan
PostKondisi	<i>Customer</i> berhasil melihat data pesanan
Aliran Normal Melihat pesanan belum diproses	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan halaman <i>home customer</i>
2. Klik menu pesanan	
3. Pilih pesanan belum diproses	
	4. Menampilkan daftar pesanan yang belum diproses yang berisi: <ul style="list-style-type: none"> <li>- Jumlah pesanan</li> <li>- Kode pesanan</li> </ul>



	<ul style="list-style-type: none"> <li>- Rasa</li> <li>- <i>Outlet</i> tempat pesan</li> <li>- Nomor wa <i>outlet</i></li> </ul>
5. Klik detail pesanan	
	<p>6. Menampilkan dialog detail pesanan yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama <i>outlet</i> tempat pemesanan</li> <li>- Kondisi penukaran</li> <li>- Rasa</li> <li>- Jumlah</li> <li>- Kemasan</li> <li>- Level</li> <li>- Harga satuan</li> <li>- Diskon</li> <li>- Harga setelah diskon</li> <li>- Poin yang akan didapat</li> <li>- Pembayaran yang digunakan</li> <li>- Status pesanan</li> <li>- Tanggal pemesanan</li> <li>- Tanggal pengambilan</li> <li>- Alamat <i>outlet</i></li> <li>- keterangan</li> </ul>
<p>Aliran Normal Melihat pesanan diproses</p>	
2. Pilih tab diproses pada <i>home customer</i>	
	<p>3. Menampilkan daftar pesanan sedang diproses yang berisi:</p> <ul style="list-style-type: none"> <li>- Jumlah pesanan</li> <li>- Kode pesanan</li> <li>- Rasa</li> <li>- <i>Outlet</i> tempat pesan</li> <li>- Nomor wa <i>outlet</i></li> </ul>
4. Pilih detail pesanan	
	<p>5. Menampilkan dialog detail pesanan sedang diproses yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama <i>outlet</i> tempat pemesanan</li> </ul>

	<ul style="list-style-type: none"> <li>- Kondisi penukaran</li> <li>- Rasa</li> <li>- Jumlah</li> <li>- Kemasan</li> <li>- Level</li> <li>- Harga satuan</li> <li>- Diskon</li> <li>- Harga setelah diskon</li> <li>- Poin yang akan didapat</li> <li>- Pembayaran yang digunakan</li> <li>- Status pesanan</li> <li>- Tanggal pemesanan</li> <li>- Tanggal pengambilan</li> <li>- Alamat <i>outlet</i></li> <li>- keterangan</li> </ul>
<p>Aliran Normal</p> <p>Melihat pesanan selesai diproses</p>	
2. Pilih tab selesai pada <i>home customer</i>	
	<p>3. Menampilkan daftar pesanan selesai diproses yang berisi:</p> <ul style="list-style-type: none"> <li>- Jumlah pesanan</li> <li>- Kode pesanan</li> <li>- Rasa</li> <li>- <i>Outlet</i> tempat pesan</li> <li>- Nomor wa <i>outlet</i></li> </ul>
4. Klik detail pesanan	
	<p>5. Menampilkan dialog detail pesanan selesai diproses yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama <i>outlet</i> tempat pemesanan</li> <li>- Kondisi penukaran</li> <li>- Rasa</li> <li>- Jumlah</li> <li>- Kemasan</li> <li>- Level</li> <li>- Harga satuan</li> <li>- Diskon</li> <li>- Harga setelah diskon</li> </ul>

	<ul style="list-style-type: none"> <li>- Poin yang akan didapat</li> <li>- Pembayaran yang digunakan</li> <li>- Status pesanan</li> <li>- Tanggal pemesanan</li> <li>- Tanggal pengambilan</li> <li>- Alamat <i>outlet</i></li> <li>- keterangan</li> </ul>
Aliran Normal Melihat pesanan ditolak	
3. Pilih ditolak	
	4. Melihat daftar pesanan yang tertolak yang berisi: <ul style="list-style-type: none"> <li>- Jumlah pesanan</li> <li>- Kode pesanan</li> <li>- Rasa</li> <li>- <i>Outlet</i> tempat pesan</li> <li>- Nomor wa <i>outlet</i></li> </ul>
5. Klik detail pesanan	
	6. Menampilkan dialog detail pesanan yang berisi: <ul style="list-style-type: none"> <li>- Nama <i>outlet</i> tempat pemesanan</li> <li>- Kondisi penukaran</li> <li>- Rasa</li> <li>- Jumlah</li> <li>- Kemasan</li> <li>- Level</li> <li>- Harga satuan</li> <li>- Diskon</li> <li>- Harga setelah diskon</li> <li>- Poin yang akan didapat</li> <li>- Pembayaran yang digunakan</li> <li>- Status pesanan</li> <li>- Tanggal pemesanan</li> <li>- Tanggal pengambilan</li> <li>- Alamat <i>outlet</i></li> <li>- keterangan</li> </ul>
Aliran Normal Melihat riwayat pesanan	

3. Pilih riwayat	
	<p>4. Menampilkan daftar riwayat pesanan yang berisi:</p> <ul style="list-style-type: none"> <li>- Jumlah pesanan</li> <li>- Kode pesanan</li> <li>- Rasa</li> <li>- <i>Outlet</i> tempat pesan</li> <li>- Nomor wa <i>outlet</i></li> </ul>
5. Klik detail pesanan	
	<p>6. Menampilkan dialog detail riwayat pesanan yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama <i>outlet</i> tempat pemesanan</li> <li>- Kondisi penukaran</li> <li>- Rasa</li> <li>- Jumlah</li> <li>- Kemasan</li> <li>- Level</li> <li>- Harga satuan</li> <li>- Diskon</li> <li>- Harga setelah diskon</li> <li>- Poin yang akan didapat</li> <li>- Pembayaran yang digunakan</li> <li>- Status pesanan</li> <li>- Tanggal pemesanan</li> <li>- Tanggal pengambilan</li> <li>- Alamat <i>outlet</i></li> <li>- Keterangan</li> </ul>

#### A.4. Mengelola data pesanan

No. <i>Use case</i>	UC04
Nama <i>use case</i>	Mengelola data pesanan
Aktor	karyawan
PreKondisi	Karyawan memilih menu pesanan
PostKondisi	Karyawan berhasil melihat daftar pesanan pada <i>outlet</i> nya

Aliran Normal Melihat data pesanan belum diproses	
Aksi Aktor	Reaksi Sistem
	7. Menampilkan halaman <i>home</i> karyawan
8. Pilih menu pesanan	
9. Pilih pesanan	
	10. Menampilkan daftar pesanan belum diproses yang berisi: <ul style="list-style-type: none"> <li>- Rasa</li> <li>- Jumlah</li> <li>- Level</li> <li>- kemasan</li> </ul>
11. klik detail pesanan	
	12. Menampilkan dialog konfirmasi pesanan dengan pilihan “terima” atau “tolak”
13. klik terima	
	14. Sistem mengubah status pesanan menjadi “sedang diproses”
	15. Menampilkan halaman pesanan belum diproses
Aliran Normal Melihat data pesanan diproses	
3. Pilih diproses	
	4. Menampilkan daftar pesanan sedang diproses yang berisi <ul style="list-style-type: none"> <li>- Rasa</li> <li>- Jumlah</li> <li>- Level</li> <li>- kemasan</li> </ul>
5. klik detail pesanan	
	6. Menampilkan dialog konfirmasi pesanan dengan pilihan “sudah” atau “belum”
7. Klik sudah	
	8. Sistem merubah status pesanan menjadi “sudah selesai”

	9. Menampilkan halaman pesanan diproses
Aliran Normal Meihat data pesanan selesai diproses	
3. Pilih selesai	
	4. Menampilkan daftar pesanan sudah selesai diproses yang berisi: <ul style="list-style-type: none"> <li>- Rasa</li> <li>- Jumlah</li> <li>- Level</li> <li>- kemasan</li> </ul>
5. klik detail pesanan	
	6. Menampilkan dialog detail pesanan yang berisi: <ul style="list-style-type: none"> <li>- Nama <i>outlet</i> tempat pemesanan</li> <li>- Kondisi penukaran</li> <li>- Rasa</li> <li>- Jumlah</li> <li>- Kemasan</li> <li>- Level</li> <li>- Harga satuan</li> <li>- Diskon</li> <li>- Harga setelah diskon</li> <li>- Poin yang akan didapat</li> <li>- Pembayaran yang digunakan</li> <li>- Status pesanan</li> <li>- Tanggal pemesanan</li> <li>- Tanggal pengambilan</li> <li>- Alamat <i>outlet</i></li> <li>- Keterangan</li> </ul>
Aliran Normal Meihat data pesanan ditolak	
5. Pilih menu	
6. Klik riwayat	
7. Pilih ditolak	
	8. Menampilkan daftar pesanan ditolak yang berisi:



	<ul style="list-style-type: none"> <li>- Rasa</li> <li>- Jumlah</li> <li>- Level</li> <li>- kemasan</li> </ul>
9. klik detail pesanan	
	<p>10. Menampilkan dialog detail pesanan ditolak yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama <i>outlet</i> tempat pemesanan</li> <li>- Kondisi penukaran</li> <li>- Rasa</li> <li>- Jumlah</li> <li>- Kemasan</li> <li>- Level</li> <li>- Harga satuan</li> <li>- Diskon</li> <li>- Harga setelah diskon</li> <li>- Poin yang akan didapat</li> <li>- Pembayaran yang digunakan</li> <li>- Status pesanan</li> <li>- Tanggal pemesanan</li> <li>- Tanggal pengambilan</li> <li>- Alamat <i>outlet</i></li> <li>- Keterangan</li> </ul>
<p>Aliran Normal Melihat data riwayat pesanan</p>	
7. Pilih diambil	
	<p>8. Menampilkan daftar pesanan sudah diambil yang berisi:</p> <ul style="list-style-type: none"> <li>- Rasa</li> <li>- Jumlah</li> <li>- Level</li> <li>- kemasan</li> </ul>
9. klik detail pesanan	
	<p>10. Menampilkan dialog detail pesanan selesai yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama <i>outlet</i> tempat pemesanan</li> <li>- Kondisi penukaran</li> </ul>

	<ul style="list-style-type: none"> <li>- Rasa</li> <li>- Jumlah</li> <li>- Kemasan</li> <li>- Level</li> <li>- Harga satuan</li> <li>- Diskon</li> <li>- Harga setelah diskon</li> <li>- Poin yang akan didapat</li> <li>- Pembayaran yang digunakan</li> <li>- Status pesanan</li> <li>- Tanggal pemesanan</li> <li>- Tanggal pengambilan</li> <li>- Alamat <i>outlet</i></li> <li>- Keterangan</li> </ul>
Aliran Normal Menolak pesanan	
7. Pilih tolak	
	8. Sistem mengubah status pesanan dari “dipesan” menjadi “ditolak”
	9. Menutup dialog konfirmasi
Aliran Normal Pesanan belum selesai diproses	
7. Pilih belum	
	8. Menutup dialog konfirmasi
Aliran Alternatif Keterangan kosong saat melakukan penolakan pesanan	
7. Pilih tolak	
	8. Sistem mengecek keterangan
	6. Menampilkan pesan “Keterangan tidak boleh kosong”

A.5. Melihat data *outlet*

No. <i>Use case</i>	UC05
Nama <i>use case</i>	Melihat data <i>outlet</i>

Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> melihat <i>outlet</i> pada <i>home customer</i>
PostKondisi	<i>Customer</i> mengetahui status <i>outlet</i> yang “buka” atau “tutup”
Aliran Normal Melihat data <i>outlet</i>	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan halaman <i>home customer</i>
2. Melihat daftar <i>outlet</i>	
3. Klik tombol detail <i>outlet</i>	
	4. Menampilkan halaman rasa yang berisi: <ul style="list-style-type: none"> <li>- Alamat <i>outlet</i></li> <li>- Rasa yang “tersedia” ataupun “kosong” dari <i>outlet</i> tersebut</li> </ul>
Aliran Alternatif Status <i>outlet</i> tutup	
2. Melihat daftar <i>outlet</i>	
3. Klik tombol detail <i>outlet</i>	
	4. Sistem mengecek status <i>outlet</i>
	5. Menampilkan pesan “ <i>outlet</i> sudah tutup”
	6. Menampilkan halaman <i>home customer</i>

#### A.6. Mengelola data *outlet*

No. <i>Use case</i>	UC06
Nama <i>use case</i>	Mengelola data <i>outlet</i>
Aktor	<i>Admin</i>
PreKondisi	<i>Admin</i> melihat daftar <i>outlet</i> di menu <i>outlet</i>
PostKondisi	<i>Admin</i> berhasil melihat daftar <i>outlet</i>
Aliran Normal Melihat data <i>outlet</i>	
Aksi Aktor	Reaksi Sistem

	<ol style="list-style-type: none"> <li>1. Menampilkan daftar <i>outlet</i> yang berisi: <ul style="list-style-type: none"> <li>- Nama <i>outlet</i></li> <li>- Waktu buka &amp; tutup</li> <li>- Alamat <i>outlet</i></li> </ul> </li> </ol>
2. Klik detail <i>outlet</i>	
	<ol style="list-style-type: none"> <li>3. Menampilkan dialog detail <i>outlet</i> yang berisi: <ul style="list-style-type: none"> <li>- Nama <i>outlet</i></li> <li>- Nomor telepon</li> <li>- Waktu buka &amp; tutup</li> <li>- Alamat <i>outlet</i></li> </ul> </li> </ol>
<p>Aliran Normal</p> <p>Mengubah data <i>outlet</i></p>	
2. Klik detail <i>outlet</i>	
	<ol style="list-style-type: none"> <li>3. Menampilkan dialog detail <i>outlet</i> yang berisi: <ul style="list-style-type: none"> <li>- Nama <i>outlet</i></li> <li>- Nomor telepon</li> <li>- Waktu buka &amp; tutup</li> <li>- Alamat <i>outlet</i></li> </ul> </li> </ol>
4. Klik ubah	
	<ol style="list-style-type: none"> <li>5. Menampilkan halaman form ubah <i>outlet</i> yang berisi: <ul style="list-style-type: none"> <li>- Foto</li> <li>- Nama <i>outlet</i></li> <li>- Waktu buka &amp; tutup</li> <li>- Status <i>outlet</i></li> <li>- Nomor telepon</li> <li>- Alamat <i>outlet</i></li> </ul> </li> </ol>
6. Klik simpan	
	<ol style="list-style-type: none"> <li>7. Sistem akan merubah data <i>outlet</i> pada database</li> </ol>
	<ol style="list-style-type: none"> <li>8. Menampilkan pesan “data <i>outlet</i> telah berhasil diperbarui”</li> </ol>
<p>Aliran Normal</p> <p>Menambah <i>outlet</i> baru</p>	

2. Klik tambah	
	<p>3. Menampilkan halaman form tambah <i>outlet</i> yang berisi:</p> <ul style="list-style-type: none"> <li>- Foto</li> <li>- Nama <i>outlet</i></li> <li>- Waktu buka &amp; tutup</li> <li>- Status <i>outlet</i></li> <li>- Nomor telepon</li> <li>- Alamat <i>outlet</i></li> </ul>
4. Klik simpan	
	5. Sistem akan menambahkan data baru <i>outlet</i> ke database
	6. Menampilkan pesan “data <i>outlet</i> baru telah berhasil ditambahkan”
<p>Aliran Alternatif Koneksi internet bermasalah</p>	
4. Klik simpan	
	5. Sistem melakukan pengecekan koneksi internet
	6. Menampilkan pesan “Koneksi internet bermasalah”
	7. Menampilkan halaman tambah/ubah <i>outlet</i>
<p>Aliran Alternatif Belum milih foto <i>outlet</i></p>	
4. Klik simpan	
	5. Sistem melakukan pengecekan foto
	6. Menampilkan pesan “pilih foto terlebih dahulu”
	7. Menampilkan halaman form tambah/ubah <i>outlet</i>
<p>Aliran Alternatif Nomor telepon atau alamat kosong</p>	
4. Klik simpan	

	5. Sistem melakukan pengecekan terhadap nomor telepon atau alamat
	6. Menampilkan pesan “data tidak boleh kosong”
	7. Menampilkan halaman form tambah/ubah <i>outlet</i>
Aliran Alternatif Validasi nomor telepon	
4. Klik simpan	
	5. Sistem melakukan validasi terhadap nomor telepon
	8. Menampilkan pesan “nomor telepon tidak valid”
	9. Menampilkan halaman form tambah/ubah <i>outlet</i>

## A.7. Mengelola data karyawan

No. <i>Use case</i>	UC07
Nama <i>use case</i>	Mengelola data karyawan
Aktor	<i>Admin</i>
PreKondisi	<i>Admin</i> memilih menu karyawan
PostKondisi	<i>Admin</i> berhasil melihat, mengaktifkan atau menon-aktifkan, mengubah, menambah dan menghapus data karyawan
Aliran Normal Melihat data karyawan	
Aksi Aktor	Reaksi Sistem
1. Klik detail karyawan	
	2. Menampilkan dialog detail karyawan yang berisi: <ul style="list-style-type: none"> <li>- Nama</li> <li>- Nomor hp</li> </ul>



	<ul style="list-style-type: none"> <li>- Tanggal lahir</li> <li>- Jenis kelamin</li> <li>- Tanggal masuk</li> <li>- Penempatan</li> <li>- Status</li> <li>- Alamat</li> </ul>
<p>Aliran Normal Mengaktifkan karyawan</p>	
5. Klik detail karyawan	
	<p>2. Menampilkan dialog detail karyawan yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama</li> <li>- Nomor hp</li> <li>- Tanggal lahir</li> <li>- Jenis kelamin</li> <li>- Tanggal masuk</li> <li>- Penempatan</li> <li>- Status</li> <li>- Alamat</li> </ul>
3. Klik aktifkan/non-aktifkan	
	<p>4. Mengubah status karyawan yang sebelumnya “aktif” menjadi “tidak aktif” dan sebaliknya.</p>
	<p>5. Menampilkan pesan “status karyawan telah berhasil diperbarui”</p>
	<p>6. Menampilkan dialog detail karyawan yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama</li> <li>- Nomor hp</li> <li>- Tanggal lahir</li> <li>- Jenis kelamin</li> <li>- Tanggal masuk</li> <li>- Penempatan</li> <li>- Status</li> <li>- Alamat</li> </ul>
<p>Aliran Normal Mengubah data karyawan</p>	

7. Klik detail karyawan	
	<p>2. Menampilkan dialog detail karyawan yang berisi:</p> <ul style="list-style-type: none"> <li>- Nama</li> <li>- Nomor hp</li> <li>- Tanggal lahir</li> <li>- Jenis kelamin</li> <li>- Tanggal masuk</li> <li>- Penempatan</li> <li>- Status</li> <li>- Alamat</li> </ul>
3. Klik ubah	
	<p>4. Menampilkan halaman form ubah karyawan yang berisi:</p> <ul style="list-style-type: none"> <li>- foto</li> <li>- Nama (<i>text</i>)</li> <li>- Alamat (<i>text</i>)</li> <li>- Nomor hp (<i>number</i>)</li> <li>- Status</li> <li>- Jenis kelamin</li> <li>- Penempatan</li> <li>- Tanggal lahir</li> </ul>
5. Mengubah data-data karyawan	
6. Klik simpan	
	7. Sistem melakukan penyimpanan perubahan data karyawan ke dalam database
	8. Menampilkan pesan “data karyawan berhasil diperbarui”
	9. Menampilkan halaman daftar karyawan
Aliran Alternatif Koneksi internet bermasalah	
6. Klik simpan	
	7. Sistem mengecek koneksi internet

	8. Menampilkan pesan “koneksi internet bermasalah”
	9. Menampilkan halaman form ubah data karyawan
Aliran Alternatif Nama, nomor hp atau alamat kosong	
6. Klik simpan	
	7. Sistem mengecek nama, nomor hp dan alamat
	8. Menampilkan pesan “data tidak boleh kosong”
	9. Menampilkan halaman form ubah karyawan
Aliran Alternatif Nomor hp tidak valid	
6. Klik simpan	
	7. Sistem mengecek nomor hp
	8. Menampilkan pesan “nomor hp tidak valid”
	9. Menampilkan halaman form ubah karyawan
Aliran Normal Menghapus data karyawan	
1. Klik detail karyawan	
	2. Menampilkan dialog detail karyawan yang berisi: <ul style="list-style-type: none"> <li>- Nama</li> <li>- Nomor hp</li> <li>- Tanggal lahir</li> <li>- Jenis kelamin</li> <li>- Tanggal masuk</li> <li>- Penempatan</li> <li>- Status</li> <li>- Alamat</li> </ul>
3. Klik hapus	

	4. Sistem menghapus data karyawan dari database
	5. Menampilkan pesan “data karyawan berhasil dihapus”
Aliran Alternatif Koneksi internet bermasalah	
3. Klik hapus	
	4. Sistem mengecek koneksi internet
	5. Menampilkan pesan “koneksi internet bermasalah”
	6. Menampilkan dialog detail karyawan
Aliran Normal Tambah data karyawan	
1. Klik simpan	
	2. Menampilkan halaman form tambah karyawan yang berisi: <ul style="list-style-type: none"> <li>- foto</li> <li>- Nama (<i>text</i>)</li> <li>- Alamat (<i>text</i>)</li> <li>- Nomor hp (<i>number</i>)</li> <li>- Status</li> <li>- Jenis kelamin</li> <li>- Penempatan</li> <li>- Tanggal lahir</li> </ul>
3. Memasukkan data-data karyawan	
4. Klik simpan	
	5. Sistem menyimpan data karyawan baru ke database
	6. Menampilkan pesan “data karyawan baru telah berhasil ditambahkan”
	7. Menampilkan halaman daftar karyawan
Aliran Alternatif Nama, nomor hp atau alamat kosong	

4. Klik simpan	
	5. Sistem mengecek nama, nomor dan alamat
	6. Menampilkan pesan “data tidak boleh kosong”
	7. Menampilkan halaman form tambah karyawan
Aliran Alternatif Koneksi internet bermasalah	
4. Klik simpan	
	5. Sistem mengecek koneksi internet
	6. Menampilkan pesan “koneksi internet bermasalah”
	7. Menampilkan halaman form tambah karyawan
Aliran Alternatif Validasi nomor hp	
4. Klik simpan	
	5. Sistem mengecek nomor hp
	6. Menampilkan pesan “nomor hp tidak valid”
	7. Menampilkan halaman form tambah karyawan

#### A.8. Melihat *leaderboard*

No. <i>Use case</i>	UC08
Nama <i>use case</i>	Melihat <i>leaderboard</i>
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> memilih menu <i>leaderboard</i>
PostKondisi	<i>Customer</i> berhasil melihat daftar <i>leaderboard</i> dan melihat <i>reward</i> yang didapat
Aliran Normal Melihat daftar <i>leaderboard</i>	

Aksi Aktor	Reaksi Sistem
	3. Menampilkan <i>reward</i> dan daftar <i>leaderboard</i> yang berisi: <ul style="list-style-type: none"> <li>- Poin</li> <li>- Nama</li> <li>- foto</li> </ul>

#### A.9. Mengelola *leaderboard*

No. Use case	UC09
Nama use case	Mengelola <i>leaderboard</i>
Aktor	<i>Admin</i>
PreKondisi	<i>Admin</i> memilih menu <i>leaderboard</i>
PostKondisi	<i>Admin</i> berhasil melihat daftar <i>leaderboard</i> , melihat <i>reward</i> dan merubah <i>reward</i>
Aliran Normal Melihat daftar <i>leaderboard</i>	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan <i>reward</i> dan daftar <i>leaderboard</i> yang berisi: <ul style="list-style-type: none"> <li>- Foto</li> <li>- Nama</li> <li>- poin</li> </ul>
Aliran Normal Mengubah <i>reward</i>	
6. klik ubah <i>reward</i>	
	7. menampilkan dialog form ubah <i>reward</i> yang berisi: <ul style="list-style-type: none"> <li>- form <i>reward</i></li> </ul>
4. memasukkan jumlah <i>reward</i>	
5. klik simpan	
	6. sistem menyimpan perubahan <i>reward</i> ke dalam database



	7. menampilkan pesan “ <i>reward</i> berhasil diperbarui”
	8. menampilkan daftar <i>leaderboard</i>
Aliran Alternatif Koneksi internet bermasalah	
5. klik simpan	
	6. sistem mengecek koneksi internet
	7. menampilkan pesan “koneksi internet bermasalah”
	8. menampilkan dialog form <i>reward</i>
Aliran Alternatif <i>Reward</i> tidak valid	
5. klik simpan	
	6. sistem mengecek <i>reward</i>
	7. menampilkan pesan “ <i>reward</i> tidak valid”
	8. menampilkan dialog form <i>reward</i>

A.10. Melihat informasi *customer*

No. <i>Use case</i>	UC10
Nama <i>use case</i>	Melihat informasi
Aktor	<i>Customer</i>
PreKondisi	Masuk halaman <i>home customer</i>
PostKondisi	<i>Customer</i> melihat daftar informasi
Aliran Normal Melihat informasi	
Aksi Aktor	Reaksi Sistem
1. Klik tombol detail	
	2. Menampilkan dialog detail informasi yang berisi: <ul style="list-style-type: none"> <li>- Judul informasi</li> <li>- Deskripsi informasi</li> </ul>

	<ul style="list-style-type: none"> <li>- Tanggal mulai</li> <li>- Tanggal berakhir</li> </ul>
--	---

## A.11. Melihat informasi karyawan

No. Use case	UC11
Nama use case	Melihat informasi
Aktor	Karyawan
PreKondisi	Karyawan memilih menu informasi
PostKondisi	Karyawan berhasil melihat daftar informasi
Aliran Normal Melihat informasi	
Aksi Aktor	Reaksi Sistem
1. Klik tombol detail	
	2. Menampilkan dialog detail informasi yang berisi: <ul style="list-style-type: none"> <li>- Judul informasi</li> <li>- Deskripsi informasi</li> <li>- Tanggal mulai</li> <li>- Tanggal berakhir</li> </ul>

## A.12. Mengelola informasi

No. Use case	UC12
Nama use case	Mengelola informasi
Aktor	<i>Admin</i>
PreKondisi	<i>Admin</i> memilih menu informasi
PostKondisi	<i>Admin</i> berhasil melihat, mengubah, menghapus dan menambah informasi
Aliran Normal Melihat informasi	
Aksi Aktor	Reaksi Sistem
1. Klik tombol detail	

	<p>2. Menampilkan dialog detail informasi yang berisi:</p> <ul style="list-style-type: none"> <li>- Judul</li> <li>- Tanggal dibuat</li> <li>- Tanggal mulai</li> <li>- Tanggal berakhir</li> <li>- deskripsi</li> </ul>
<p>Aliran Normal Mengubah informasi</p>	
1. klik tombol detail	
	<p>2. Menampilkan dialog detail informasi yang berisi:</p> <ul style="list-style-type: none"> <li>- Judul</li> <li>- Tanggal dibuat</li> <li>- Tanggal mulai</li> <li>- Tanggal berakhir</li> <li>- deskripsi</li> </ul>
3. klik ubah	
	<p>4. menampilkan halaman form ubah informasi yang berisi:</p> <ul style="list-style-type: none"> <li>- foto</li> <li>- judul (teks)</li> <li>- deskripsi (teks)</li> <li>- tanggal mulai</li> <li>- tanggal berakhir</li> </ul>
5. klik simpan	
	6. sistem menyimpan perubahan informasi ke dalam database
	7. menampilkan pesan “informasi telah berhasil diperbarui”
	8. menampilkan halaman daftar informasi
<p>Aliran Alternatif Koneksi internet bermasalah</p>	
5. klik simpan	
	6. sistem mengecek koneksi internet

	7. menampilkan pesan “koneksi internet bermasalah”
	8. menampilkan halaman form ubah informasi
Aliran Alternatif Judul atau deskripsi kosong	
5. klik simpan	
	6. sistem melakukan pengecekan terhadap judul dan deskripsi
	7. menampilkan pesan “data tidak boleh kosong”
	8. menampilkan halaman form ubah informasi
Aliran Alternatif Foto kosong	
5. klik simpan	
	6. sistem mengecek foto
	7. menampilkan pesan “pilih foto terlebih dahulu”
	8. menampilkan halaman form ubah informasi
Aliran Normal Menghapus informasi	
1. klik tombol detail	
	2. Menampilkan dialog detail informasi yang berisi: <ul style="list-style-type: none"> <li>- Judul</li> <li>- Tanggal dibuat</li> <li>- Tanggal mulai</li> <li>- Tanggal berakhir</li> <li>- deskripsi</li> </ul>
3. klik hapus	
	4. sistem menghapus data informasi dari database

	5. menampilkan pesan “informasi berhasil dihapus”
	6. menampilkan halaman daftar informasi
Aliran Alternatif Koneksi internet bermasalah	
3. klik hapus	
	4. sistem mengecek koneksi internet
	5. menampilkan pesan “koneksi internet bermasalah”
	6. menampilkan dialog detail informasi

A.13. Melihat data rasa *customer*

No. <i>Use case</i>	UC13
Nama <i>use case</i>	Melihat data rasa
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> memilih <i>outlet</i> yang buka
PostKondisi	<i>Customer</i> berhasil melihat daftar rasa pada <i>outlet</i> tersebut
Aliran Normal Melihat daftar rasa	
Aksi Aktor	Reaksi Sistem
1. Klik tombol detail <i>outlet</i>	
	2. Menampilkan daftar rasa yang berisi: <ul style="list-style-type: none"> <li>- Rasa</li> <li>- Status persediaan</li> </ul>
Aliran Alternatif <i>Outlet</i> tutup	
1. Klik tombol detail <i>outlet</i>	
	2. Sistem mengecek status <i>outlet</i>
	3. Menampilkan pesan “ <i>outlet</i> sudah tutup, silahkan pilih <i>outlet</i> yang lainnya”

	4. Menampilkan halaman <i>home customer</i>
--	---

## A.14. Melihat data rasa karyawan

No. <i>Use case</i>	UC14
Nama <i>use case</i>	Melihat data rasa
Aktor	Karyawan
PreKondisi	karyawan memilih menu rasa
PostKondisi	Karyawan berhasil melihat daftar rasa dan mengubah status tersedia
Aliran Normal Melihat daftar rasa	
Aksi Aktor	Reaksi Sistem
3. Klik tombol detail	
	4. Menampilkan dialog ubah status yang berisi: - Status persediaan rasa
5. Klik tombol tersedia/kosong	
	6. Sistem menyimpan perubahan status persediaan rasa ke database
	7. Menampilkan pesan “status persediaan telah berhasil diperbarui”
	8. Menampilkan halaman daftar rasa
Aliran Alternatif Koneksi internet bermasalah	
5. Klik tombol tersedia/kosong	
	6. Sistem mengecek status <i>outlet</i>
	7. Menampilkan pesan “ <i>outlet</i> sudah tutup, silahkan pilih <i>outlet</i> yang lainnya”
	8. Menampilkan halaman <i>home customer</i>

## A.15. Mengelola data rasa



No. <i>Use case</i>	UC15
Nama <i>use case</i>	Mengelola data rasa
Aktor	<i>Admin</i>
PreKondisi	<i>Admin</i> memilih menu rasa
PostKondisi	<i>Admin</i> berhasil melihat, mengubah status persediaan, menghapus dan menambah rasa baru
Aliran Normal Melihat daftar rasa	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan daftar rasa yang berisi: <ul style="list-style-type: none"> <li>- Rasa</li> <li>- Status persediaan</li> </ul>
Aliran Normal Mengubah status persediaan	
	1. Menampilkan daftar rasa yang berisi <ul style="list-style-type: none"> <li>- Rasa</li> <li>- Status persediaan</li> </ul>
2. Klik tombol detail	
	3. Menampilkan dialog detail rasa yang berisi: <ul style="list-style-type: none"> <li>- Rasa</li> <li>- Persediaan</li> </ul>
4. Klik tersedia/kosong	
	5. Sistem menyimpan perubahan status persediaan ke database
	6. Menampilkan pesan “status persediaan berhasil diperbarui”
	7. Menampilkan halaman daftar rasa
Aliran Alternatif Koneksi internet bermasalah	
8. Klik tersedia/kosong	
	9. Sistem mengecek koneksi internet

	10. Menampilkan pesan “koneksi internet bermasalah”
	11. Menampilkan dialog detail rasa
Aliran Normal Meghapus rasa	
4. Klik hapus	
	5. Sistem menghapus rasa pada database
	6. Menampilkan pesan “data rasa berhasil dihapus”
	7. Menampilkan halaman daftar rasa
Aliran Alternatif Koneksi internet bermasalah	
4. Klik hapus	
	5. Sistem mengecek koneksi internet
	6. Menampilkan pesan “koneksi internet bermasalah”
	7. Menampilkan dialog detail rasa
Aliran Normal Menambah rasa	
2. Klik tombol tambah	
	3. Menampilkan halaman form tambah rasa yang berisi: - Rasa (teks)
4. Klik simpan	
	5. Sistem menyimpan rasa ke database
	6. Menampilkan pesan “rasa baru telah berhasil ditambahkan”
	7. Menampilkan daftar rasa
Aliran Alternatif Koneksi internet bermasalah	
4. Klik simpan	
	5. Sistem mengecek koneksi internet

	6. Menampilkan pesan “koneksi internet bermasalah”
	7. Menampilkan halaman form tambah rasa
Aliran Alternatif Belum pilih foto	
4. Klik simpan	
	5. Sistem mengecek foto
	6. Menampilkan pesan “pilih foto terlebih dahulu”
	7. Menampilkan halaman form tambah rasa
Aliran Alternatif Rasa kosong	
4. Klik simpan	
	5. Sistem mengecek rasa
	6. Menampilkan pesan “data tidak boleh kosong”
	7. Menampilkan halaman form tambah rasa
Aliran Alternatif Rasa sudah tersedia	
4. Klik simpan	
	5. Sistem mengecek rasa ke database
	6. Menampilkan pesan “rasa sudah tersedia”
	7. Menampilkan halaman form tambah rasa

## A.16. Melakukan penukaran hadiah

No. Use case	UC16
Nama use case	Melakukan penukaran data hadiah
Aktor	Customer
PreKondisi	Customer berada pada halaman home customer

PostKondisi	<i>Customer</i> berhasil melakukan penukaran hadiah menggunakan <i>M-Poin</i>
Aliran Normal Menukar hadiah menggunakan <i>M-Poin</i>	
Aksi Aktor	Reaksi Sistem
9. Klik tombol tukar	
	10. Sistem menyimpan hadiah ke database
	11. Menampilkan pesan “penukaran hadiah telah berhasil dilakukan”
	12. Menampilkan halaman <i>home customer</i>
Aliran Alternatif	
9. Klik tombol tukar	
	10. Sistem mengecek <i>M-Poin customer</i>
	11. Menampilkan pesan “ <i>M-Poin</i> tidak cukup”
	12. Menampilkan halaman <i>home customer</i>

## A.17. Melihat data hadiah

No. <i>Use case</i>	UC17
Nama <i>use case</i>	Melihat data hadiah
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> berada pada halaman <i>home customer</i>
PostKondisi	<i>Customer</i> berhasil melihat daftar hadiah
Aliran Normal Melihat daftar hadiah	
Aksi Aktor	Reaksi Sistem
13. Klik tombol detail	
	14. Menampilkan dialog detail hadiah yang berisi: <ul style="list-style-type: none"> <li>- Nama</li> <li>- Jumlah tersedia</li> <li>- Harga (<i>M-Poin</i>)</li> </ul>

	<ul style="list-style-type: none"> <li>- Merek</li> <li>- Deskripsi</li> </ul>
--	--

## A.18. Melakukan pemesanan

No. <i>Use case</i>	UC18
Nama <i>use case</i>	Melakukan pemesanan
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> melihat persediaan rasa pada <i>outlet</i>
PostKondisi	<i>Customer</i> berhasil melakukan pemesanan
Aliran Normal Pemesanan produk	
Aksi Aktor	Reaksi Sistem
15. Klik tombol pesan	16. Menampilkan dialog pemesanan yang berisi: <ul style="list-style-type: none"> <li>- Rasa</li> <li>- <i>Outlet</i> tempat pesan</li> <li>- Alamat <i>outlet</i></li> <li>- Kemasan</li> <li>- Level</li> <li>- Jumlah pesanan</li> <li>- Pembayaran</li> <li>- Diskon</li> <li>- Harga</li> <li>- Harga setelah diskon</li> <li>- <i>M-Poin</i> yang akan didapatkan</li> </ul>
17. Menentukan pembayaran, jumlah, kemasan dan level	
18. Klik pesan	
	19. Sistem menyimpan data pesanan ke database
	20. Menampilkan pesan “pemesanan telah berhasil dilakukan”
	21. Menampilkan halaman daftar rasa

Aliran Alternatif Status rasa kosong	
15. Klik tombol pesan	
	16. Sistem mengecek status persediaan
	17. Menampilkan pesan “rasa kosong”
	18. Menampilkan halaman daftar rasa
Aliran Alternatif <i>M-Cash</i> tidak cukup	
7. Memilih metode pembayaran <i>M-Cash</i>	
8. Klik pesan	
	19. Sistem mengecek <i>M-Cash customer</i>
	20. Menampilkan pesan “ <i>M-Cash</i> tidak cukup”
	21. Menampilkan dialog pemesanan
Aliran Alternatif Koneksi internet bermasalah	
12. Klik pesan	
	13. Sistem mengecek koneksi internet
	14. Menampilkan pesan “koneksi internet bermasalah”
	15. Menampilkan dialog pemesanan

## A.19. Mengelola data hadiah

No. <i>Use case</i>	UC19
Nama <i>use case</i>	Mengelola data hadiah
Aktor	<i>Admin</i>
PreKondisi	<i>Admin</i> memilih menu hadiah
PostKondisi	<i>Admin</i> berhasil melihat, mengubah, menghapus dan menambah data hadiah
Aliran Normal Melihat data hadiah	



Aksi Aktor	Reaksi Sistem
1. Klik tombol detail	
	2. Menampilkan dialog detail hadiah yang berisi: <ul style="list-style-type: none"> <li>- Tanggal buat</li> <li>- Nama</li> <li>- Harga (<i>M-Poin</i>)</li> <li>- Jumlah</li> <li>- Merek</li> <li>- Deskripsi</li> </ul>
Aliran Normal hapus data hadiah	
1. Klik tombol detail	
	2. Menampilkan dialog detail hadiah yang berisi: <ul style="list-style-type: none"> <li>- Tanggal buat</li> <li>- Nama</li> <li>- Harga (<i>M-Poin</i>)</li> <li>- Jumlah</li> <li>- Merek</li> <li>- Deskripsi</li> </ul>
3. Klik hapus	
	4. Sistem menghapus data hadiah dari database
	5. Menampilkan pesan “data hadiah berhasil dihapus”
	6. Menampilkan halaman daftar hadiah
Aliran Alternatif Koneksi internet bermasalah	
3. Klik hapus	
	4. Sistem mengecek koneksi internet
	5. Menampilkan pesan “koneksi internet bermasalah”
	6. Menampilkan halaman daftar hadiah
Aliran Normal	

Ubah hadiah	
7. Klik tombol detail	
	2. Menampilkan dialog detail hadiah yang berisi: <ul style="list-style-type: none"> <li>- Tanggal buat</li> <li>- Nama</li> <li>- Harga (<i>M-Poin</i>)</li> <li>- Jumlah</li> <li>- Merek</li> <li>- Deskripsi</li> </ul>
3. Klik ubah	
	4. Menampilkan halaman form ubah hadiah yang berisi: <ul style="list-style-type: none"> <li>- Foto</li> <li>- Nama (teks)</li> <li>- Deskripsi (teks)</li> <li>- Harga (angka)</li> <li>- Merek (teks)</li> <li>- Jumlah (angka)</li> </ul>
5. Melakukan perubahan pada data hadiah	
6. Klik simpan	
	7. Sistem menyimpan perubahan data hadiah ke database
	8. Menampilkan pesan “data hadiah berhasil diperbarui”
	9. Menampilkan halaman daftar hadiah
Aliran Alternatif Koneksi internet bermasalah	
6. Klik simpan	
	7. Sistem mengecek koneksi internet
	8. Menampilkan pesan “koneksi internet bermasalah”
	9. Menampilkan halaman form ubah hadiah
Aliran Alternatif	

Belum memilih foto	
6. Klik simpan	
	7. Sistem mengecek foto
	8. Menampilkan pesan “pilih foto terlebih dahulu”
	9. Menampilkan halaman form ubah hadiah
Aliran Alternatif Nama, harga, jumlah, deskripsi dan merek kosong	
6. Klik simpan	
	7. Sistem mengecek nama, harga, jumlah, deskripsi dan merek
	8. Menampilkan pesan “data tidak boleh kosong”
	9. Menampilkan halaman form ubah hadiah
Aliran Alternatif Harga tidak valid	
6. Klik simpan	
	7. Sistem mengecek harga
	8. Menampilkan pesan “harga tidak valid”
	9. Menampilkan halaman form ubah hadiah
Aliran Alternatif Jumlah tidak valid	
6. Klik simpan	
	7. Sistem mengecek jumlah
	8. Menampilkan pesan “jumlah tidak valid”
	9. Menampilkan halaman form ubah hadiah

#### A.20. Melihat data *M-Cash*

No. <i>Use case</i>	UC20
Nama <i>use case</i>	Melihat data <i>M-Cash</i>
Aktor	<i>Customer</i>

PreKondisi	<i>Customer</i> memilih menu <i>M-Cash</i>
PostKondisi	<i>Customer</i> berhasil melihat riwayat transaksi <i>M-Cash</i>
Aliran Normal Melihat riwayat <i>M-Cash</i>	
Aksi Aktor	Reaksi Sistem
	<ol style="list-style-type: none"> <li>Menampilkan daftar riwayat transaksi <i>M-Cash</i> yang berisi: <ul style="list-style-type: none"> <li>Tanggal transaksi</li> <li><i>M-Cash</i></li> <li>Harga</li> </ul> </li> </ol>

A.21. Mengelola data *M-Cash*

No. <i>Use case</i>	UC21
Nama <i>use case</i>	Mengelola data <i>M-Cash</i>
Aktor	<i>Admin</i> , karyawan
PreKondisi	<i>User</i> memilih menu <i>M-Cash</i>
PostKondisi	<i>User</i> berhasil melihat riwayat transaksi dan menambahkan <i>M-Cash</i>
Aliran Normal Melihat riwayat transaksi <i>M-Cash</i>	
Aksi Aktor	Reaksi Sistem
	<ol style="list-style-type: none"> <li>Menampilkan daftar riwayat transaksi <i>M-Cash</i> yang berisi: <ul style="list-style-type: none"> <li>Tanggal transaksi</li> <li><i>M-Cash</i></li> <li>Harga</li> </ul> </li> </ol>
Aliran Normal Menambah <i>M-Cash</i>	
	<ol style="list-style-type: none"> <li>Menampilkan daftar riwayat transaksi <i>M-Cash</i> yang berisi: <ul style="list-style-type: none"> <li>Tanggal transaksi</li> <li><i>M-Cash</i></li> <li>Harga</li> </ul> </li> </ol>

2. Memasukkan <i>username customer</i>	
3. Klik tombol cek	
	4. Menampilkan dialog form <i>M-Cash</i> yang berisi: - Nominal (angka)
5. Memasukkan transaksi nominal	
6. Klik konfirmasi	
	7. Sistem menambahkan <i>M-Cash</i> ke <i>customer</i>
	8. Menampilkan pesan “transaksi <i>M-Cash</i> telah berhasil dilakukan”
	9. Menampilkan dialog form <i>M-Cash</i>
10. Klik selesai	
	11. Menampilkan halaman daftar riwayat transaksi <i>M-Cash</i>
Aliran Alternatif Nominal tidak valid	
6. Klik konfirmasi	
	7. Sistem mengecek nominal
	8. Menampilkan pesan “nominal tidak valid”
	9. Menampilkan dialog <i>M-Cash</i>
Aliran Alternatif Nominal kosong	
6. Klik konfirmasi	
	7. Sistem mengecek nominal
	8. Menampilkan pesan “data tidak boleh kosong”
	9. Menampilkan dialog <i>M-Cash</i>
Aliran Alternatif Koneksi internet bermasalah	

6. Klik konfirmasi	
	7. Sistem mengecek koneksi internet
	8. Menampilkan pesan “koneksi internet bermasalah”

A.22. Melihat data *M-Poin*

No. Use case	UC22
Nama use case	Melihat data <i>M-Poin</i>
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> memilih menu <i>M-Poin</i>
PostKondisi	<i>Customer</i> berhasil melihat riwayat <i>M-Poin</i> yang didapatkan
Aliran Normal Melihat riwayat <i>M-Poin</i>	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan daftar riwayat <i>M-Poin</i> diperoleh yang berisi: <ul style="list-style-type: none"> <li>- Tanggal diperoleh</li> <li>- <i>M-Poin</i></li> </ul>

## A.23. Logout

No. Use case	UC23
Nama use case	<i>Logout</i>
Aktor	<i>Customer, karyawan, admin</i>
PreKondisi	<i>User</i> telah <i>login</i> terlebih dahulu kedalam sistem dan berada halaman <i>home</i> masing-masing <i>user</i>
PostKondisi	<i>User</i> keluar dari sistem dan akan diteruskan ke halaman <i>login</i>
Aliran Normal	



Aksi Aktor	Reaksi Sistem
1. Pilih menu	
2. Pilih keluar	
	3. Menampilkan pesan “ <i>logout</i> berhasil”
	4. Menampilkan halaman <i>login</i>
Aliran Alternatif Koneksi internet bermasalah	
2. Pilih keluar	
	3. Sistem mengecek koneksi internet
	4. Menampilkan pesan “Koneksi internet bermasalah”
	5. Menampilkan halaman <i>home</i>

## A.24. Verifikasi pesanan

No. <i>Use case</i>	UC24
Nama <i>use case</i>	Verifikasi pesanan
Aktor	Karyawan
PreKondisi	Karyawan pilih menu verifikasi
PostKondisi	Karyawan berhasil verifikasi pesanan
Aliran Normal Verifikasi pesanan	
Aksi Aktor	Reaksi Sistem
7. Masukkan kode pesanan	
8. Klik cek	
	9. Menampilkan dialog detail pesanan
10. Klik konfirmasi	
	11. Merubah status pesanan dan menyimpannya ke database
	12. Menampilkan pesan “pesanan berhasil diverifikasi”

Aliran Alternatif Kode kosong	
5. Klik cek	
	6. mengecek kode pesanan
	7. menampilkan pesan “masukkan kode pesanan”
Aliran Alternatif Internet bermasalah	
7. klik konfirmasi	
	8. mengecek koneksi internet
	9. menampilkan pesan “koneksi internet bermasalah”
Aliran Alternatif Salah <i>outlet</i>	
7. klik konfirmasi	
	8. mengecek <i>outlet</i>
	9. menampilkan pesan “salat <i>outlet</i> ”
Aliran Alternatif Masih diproses	
7. klik konfirmasi	
	8. mengecek pesanan
	9. menampilkan pesan “masih diproses”
Aliran Alternatif Sudah diambil	
7. klik konfirmasi	
	8. mengecek pesanan
	9. menampilkan pesan “pesanan sudah diambil”

## A.25. Verifikasi hadiah

No. <i>Use case</i>	UC24
---------------------	------

Nama <i>use case</i>	Verifikasi hadiah
Aktor	<i>Admin</i>
PreKondisi	<i>Admin</i> pilih menu verifikasi
PostKondisi	<i>Admin</i> berhasil verifikasi hadiah
Aliran Normal Verifikasi hadiah	
Aksi Aktor	Reaksi Sistem
13. Masukkan kode hadiah	
14. Klik cek	
	15. Menampilkan dialog detail hadiah
16. Klik konfirmasi	
	17. Merubah status pesanan dan menyimpannya ke database
	18. Menampilkan pesan “pesanan berhasil diverifikasi”
Aliran Alternatif Kode kosong	
8. Klik cek	
	9. mengecek kode pesanan
	10. menampilkan pesan “masukkan kode hadiah”
Aliran Alternatif Internet bermasalah	
10. klik konfirmasi	
	11. mengecek koneksi internet
	12. menampilkan pesan “koneksi internet bermasalah”
Aliran Alternatif Sudah ditukar	
10. klik konfirmasi	
	11. mengecek pesanan

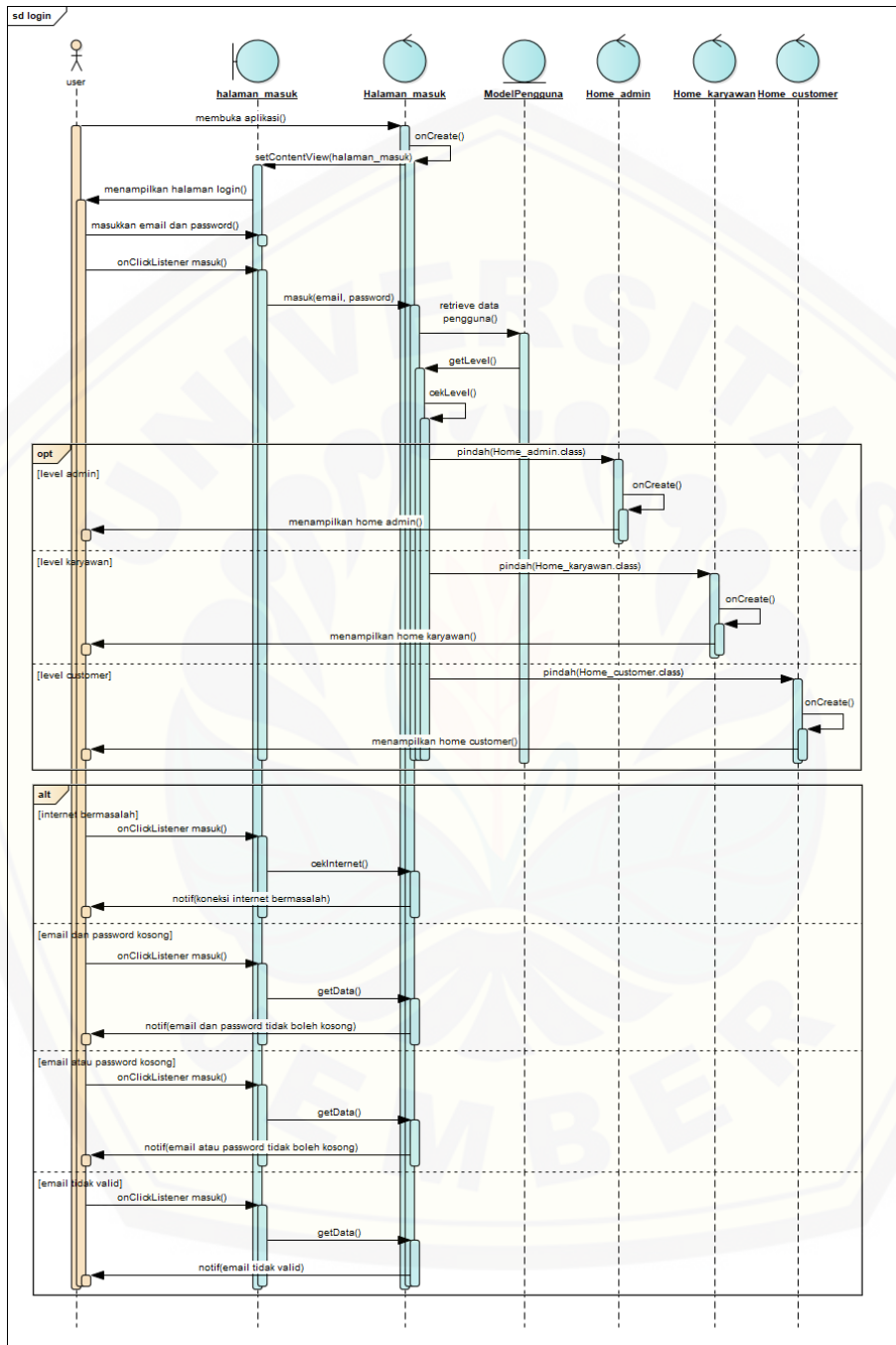
	12. menampilkan pesan “pesanan sudah diambil”
--	---

## A.26. Rasa favorit

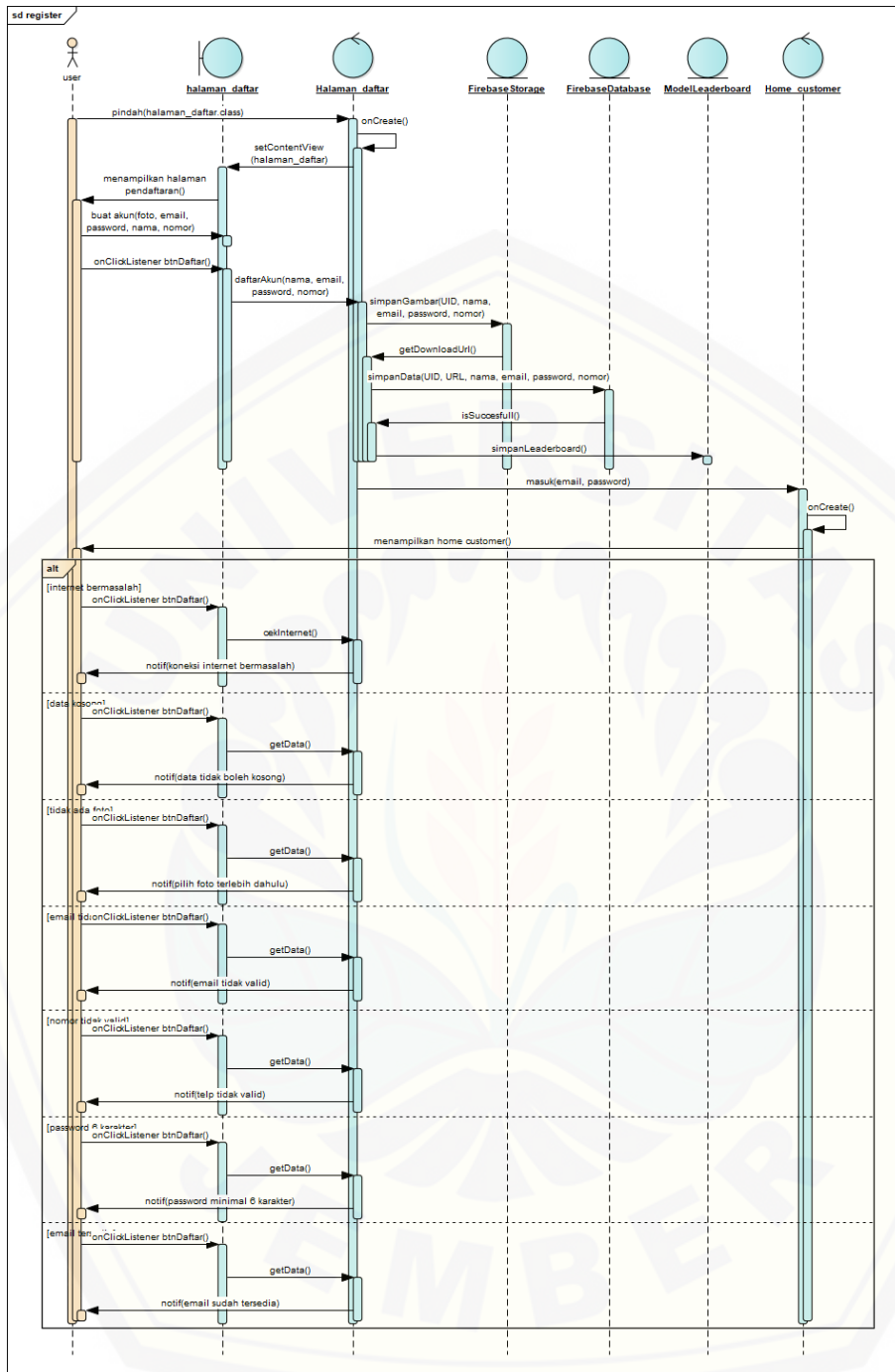
No. <i>Use case</i>	UC26
Nama <i>use case</i>	Melihat rasa favorit
Aktor	<i>Customer</i>
PreKondisi	<i>Customer</i> berada pada <i>home customer</i>
PostKondisi	<i>Customer</i> berhasil melihat daftar rasa favorit
Aliran Normal Melihat daftar rasa favorit	
Aksi Aktor	Reaksi Sistem
1. Klik lihat semua	
	2. Menampilkan halaman daftar rasa favorit

B. Sequence diagram

B.1. Login

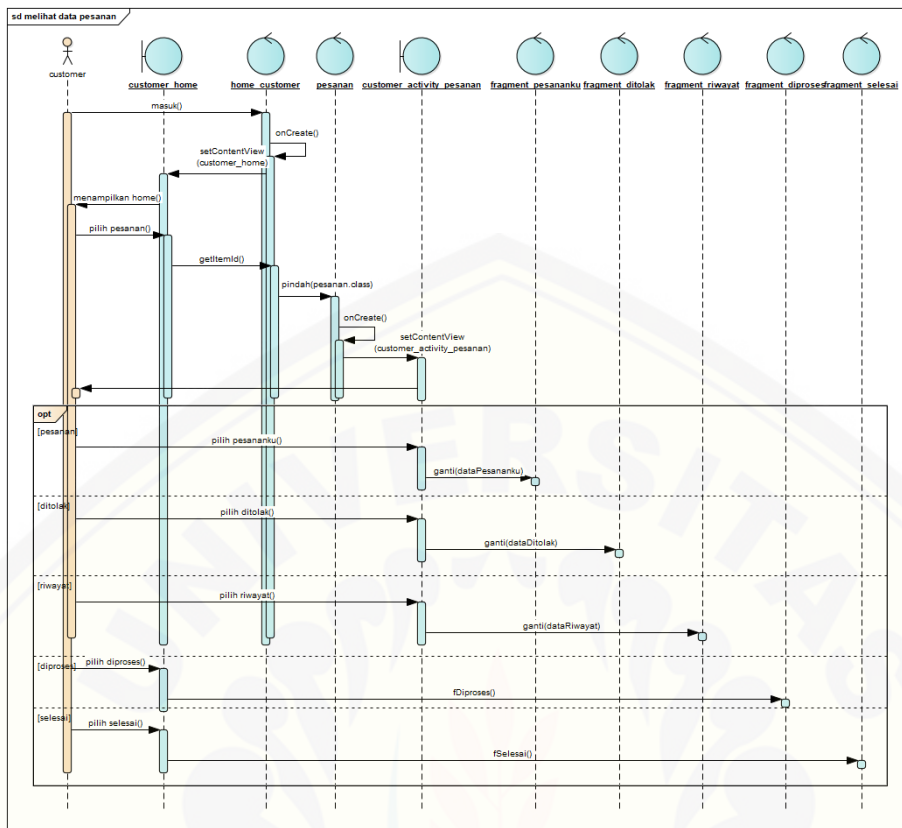


B.2. Registrasi

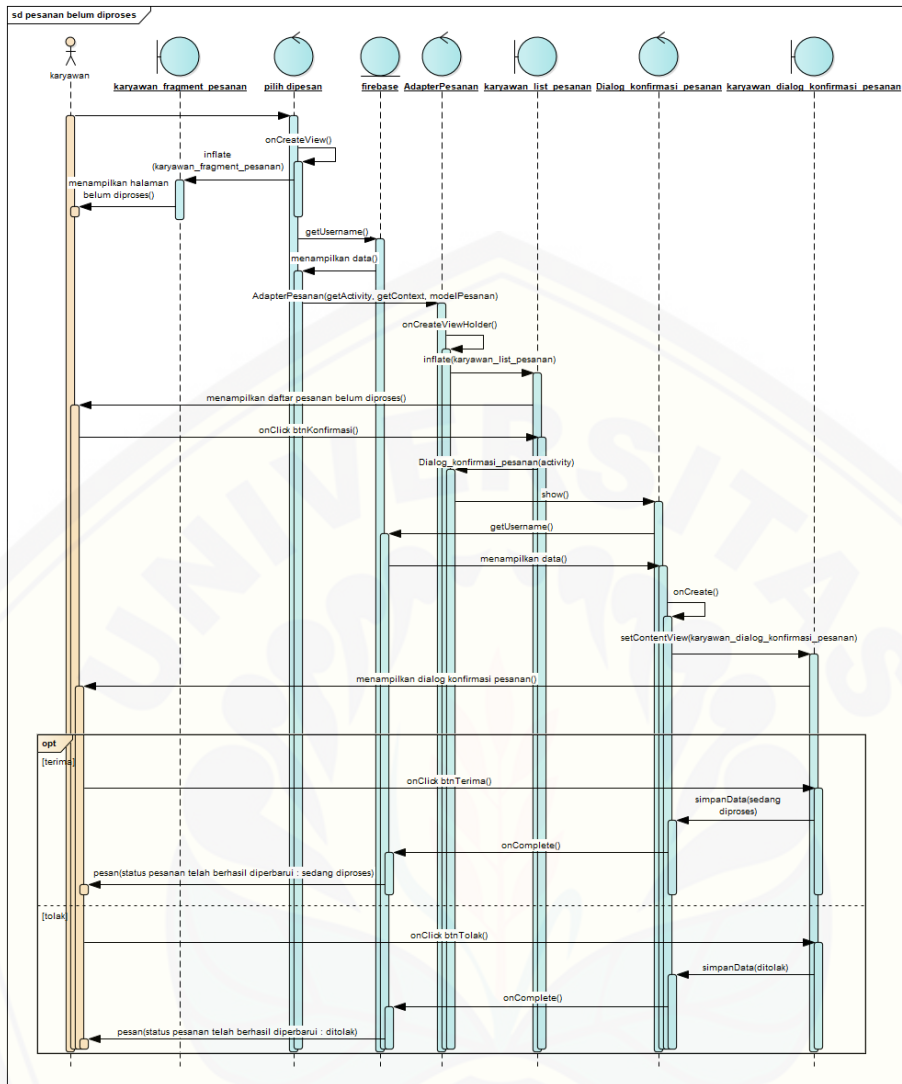


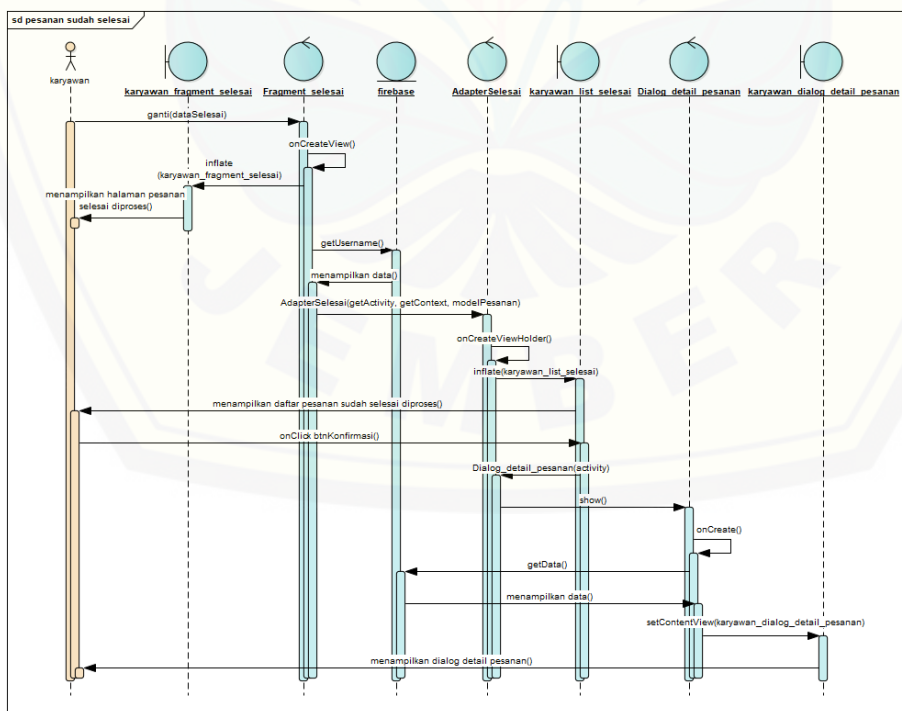
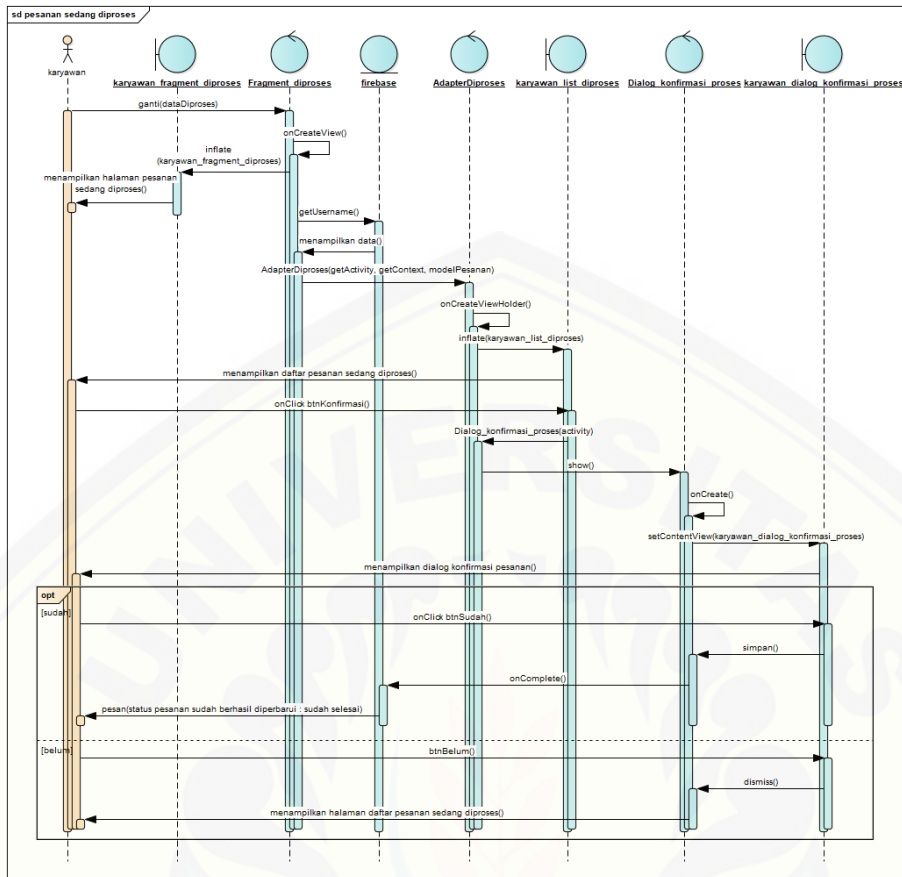
B.3. Melihat data pesanan

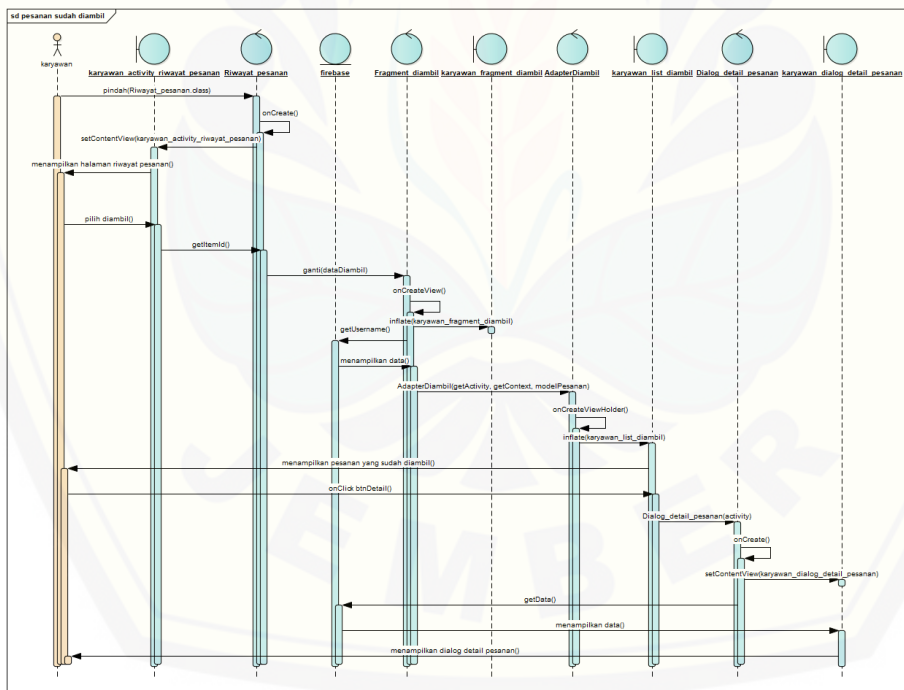
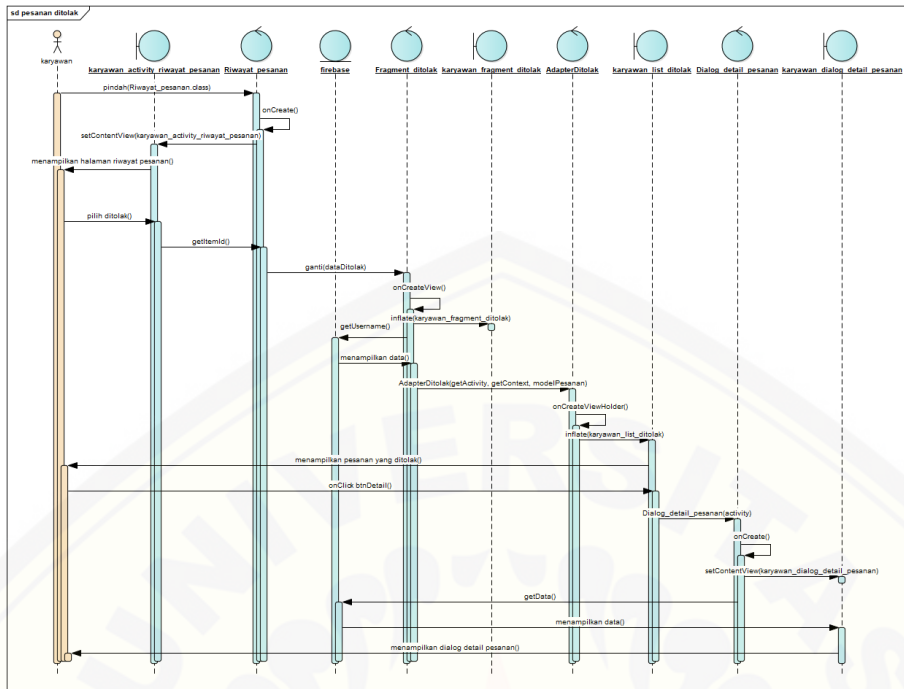




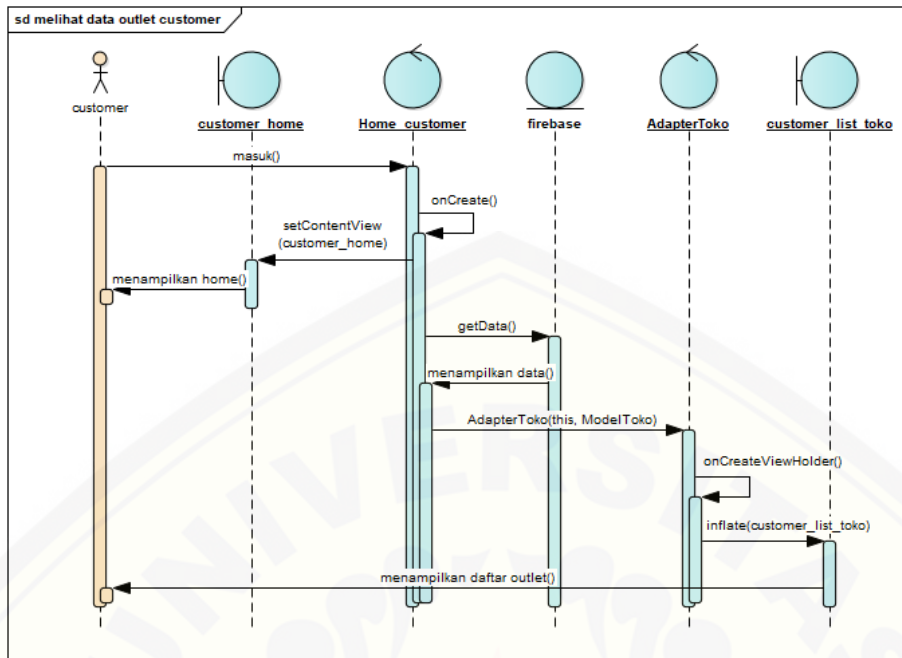
B.4. Mengelola data pesanan



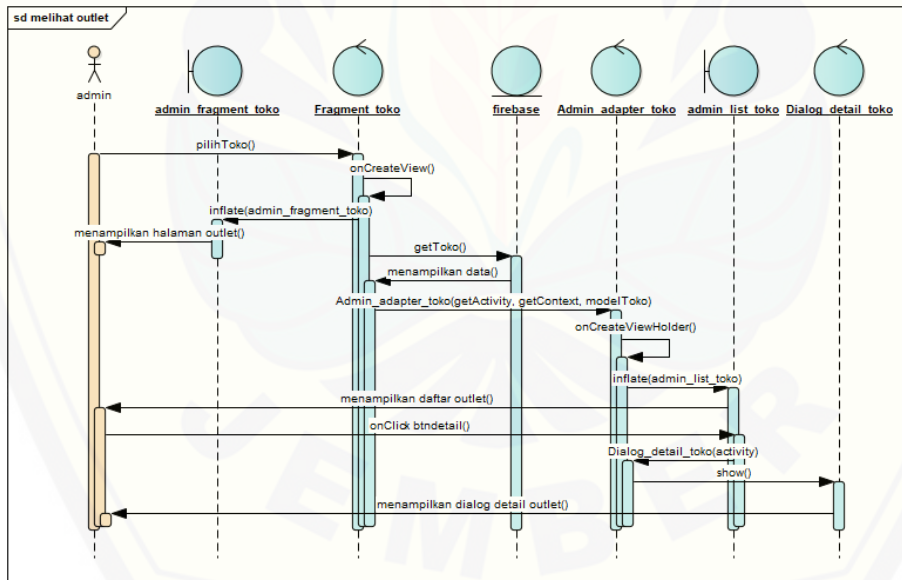


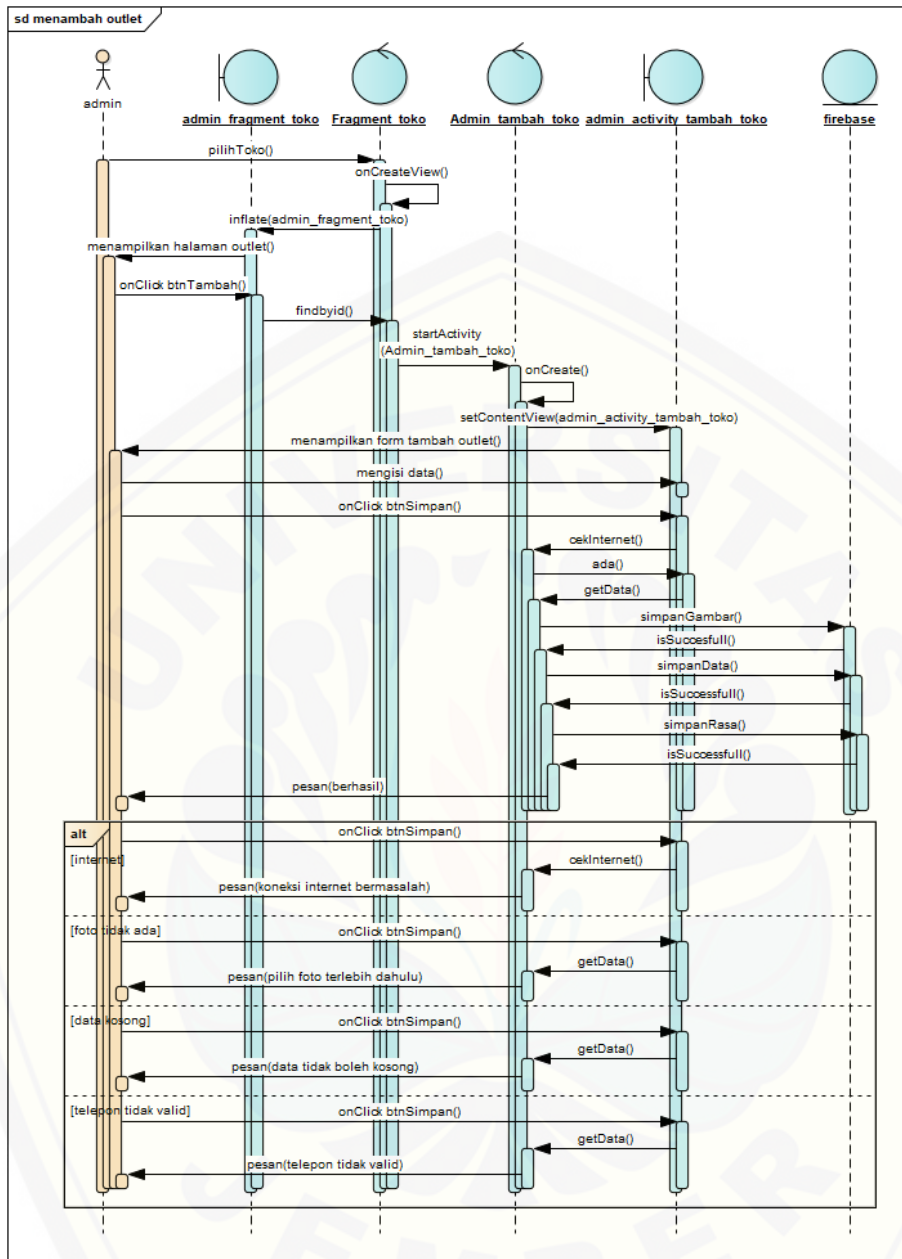


B.5. Melihat data *outlet*

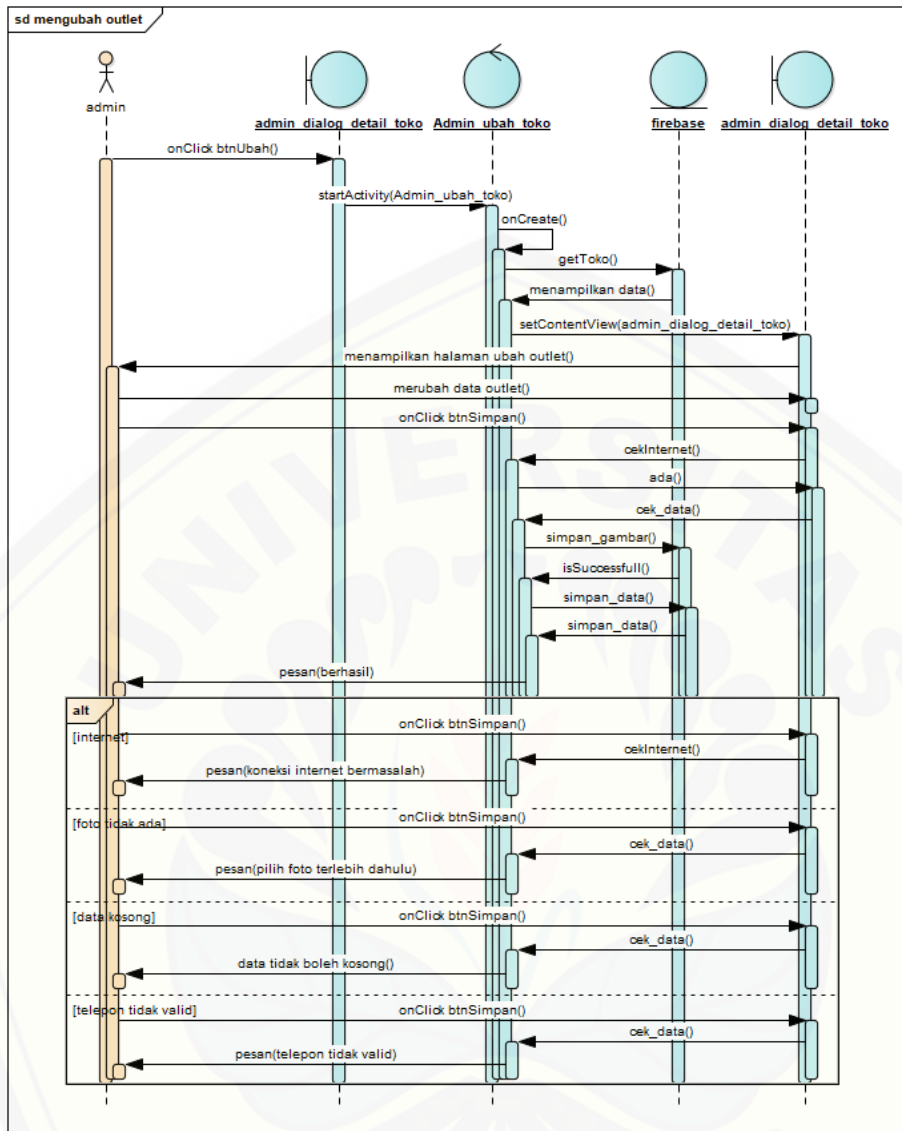


### B.6. Mengelola data outlet

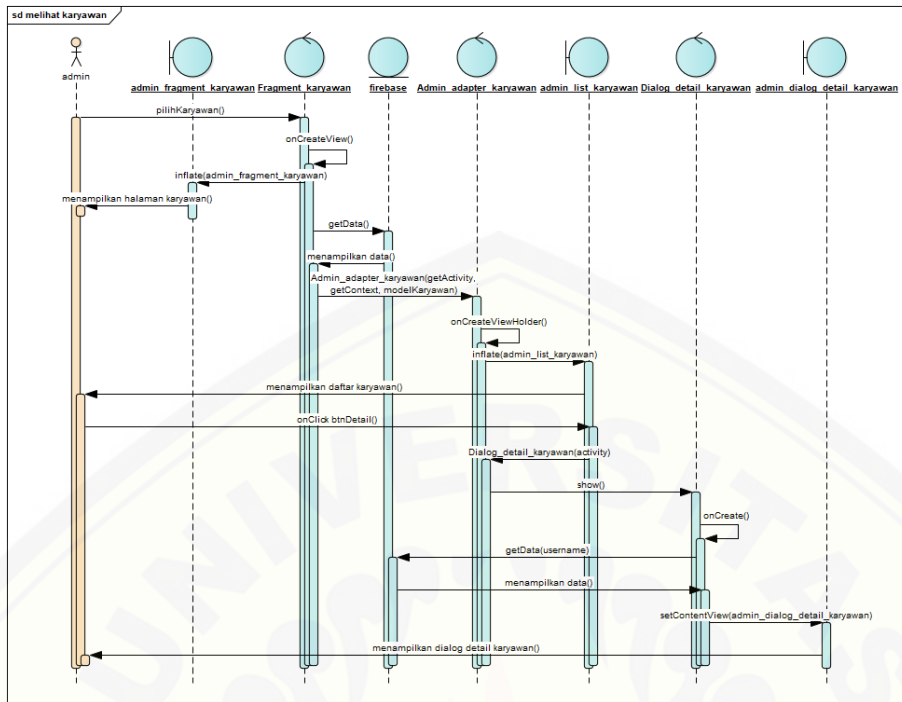


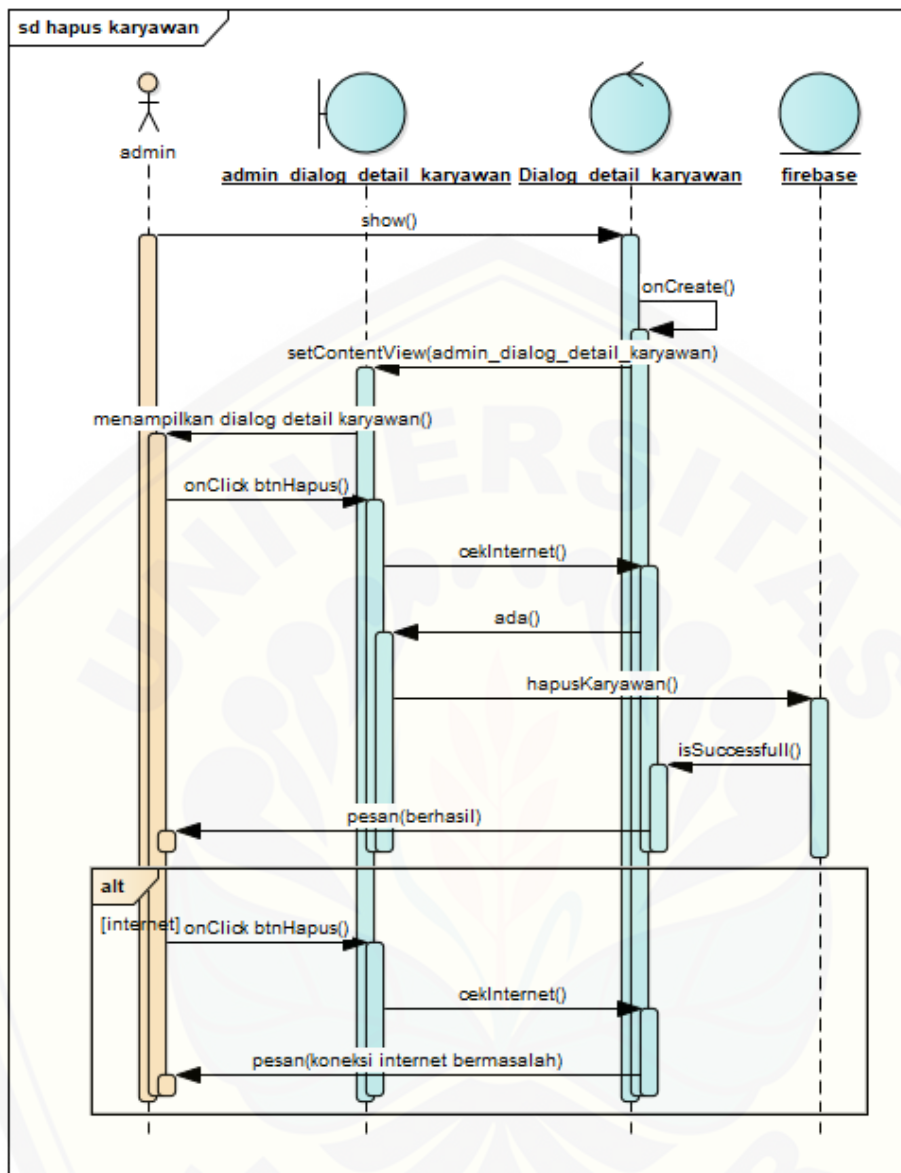


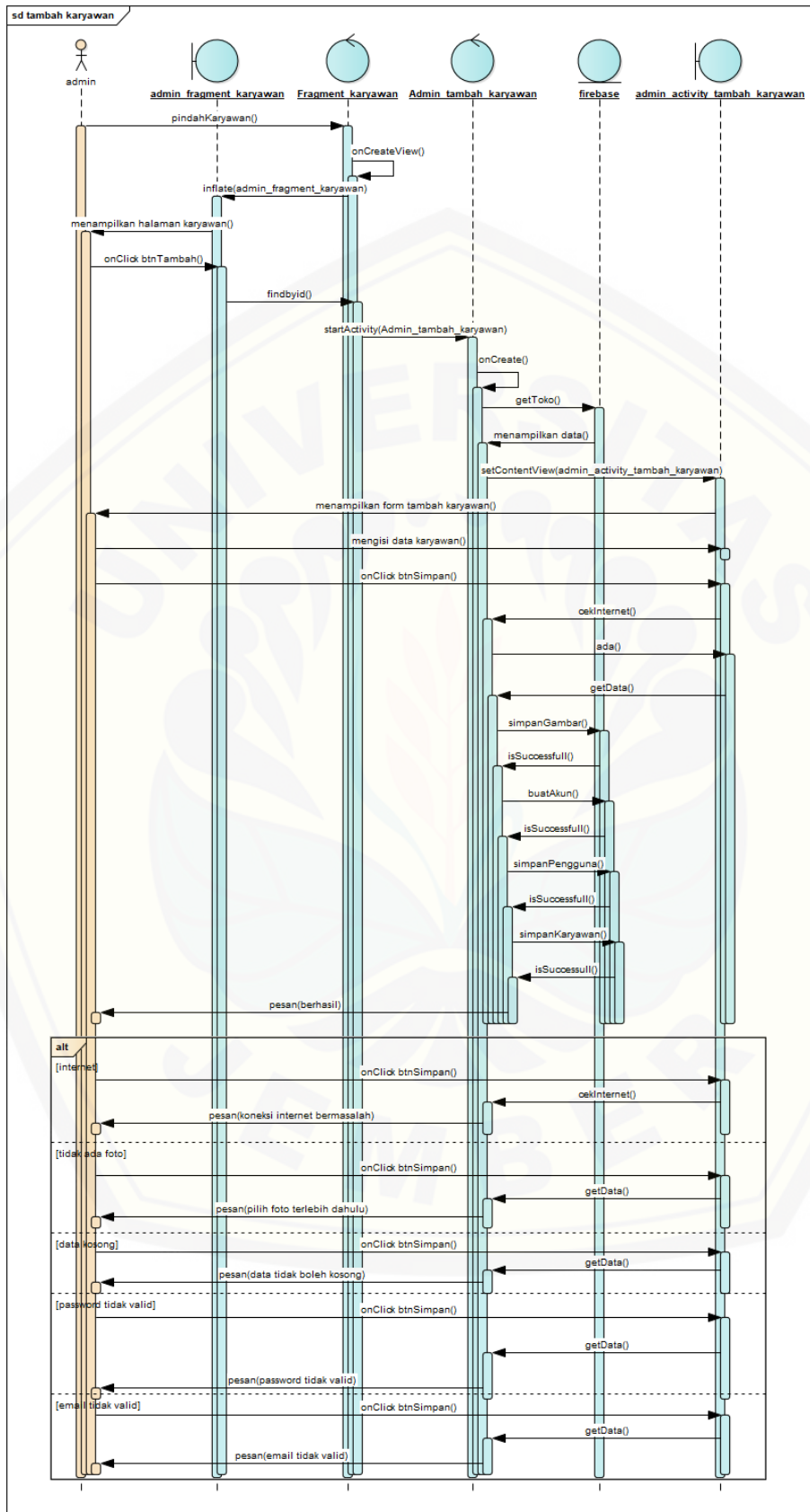


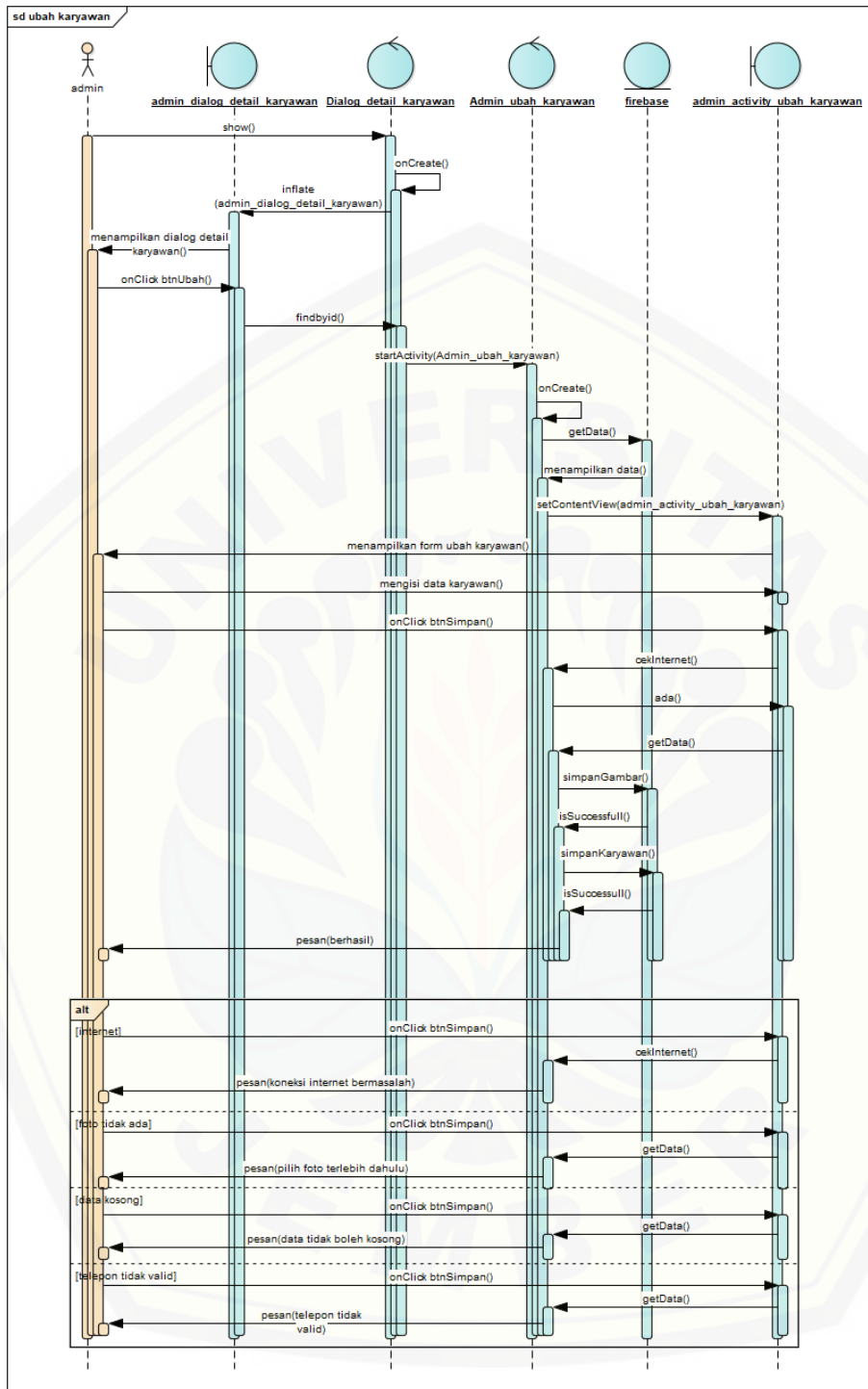


B.7. Mengelola data karyawan

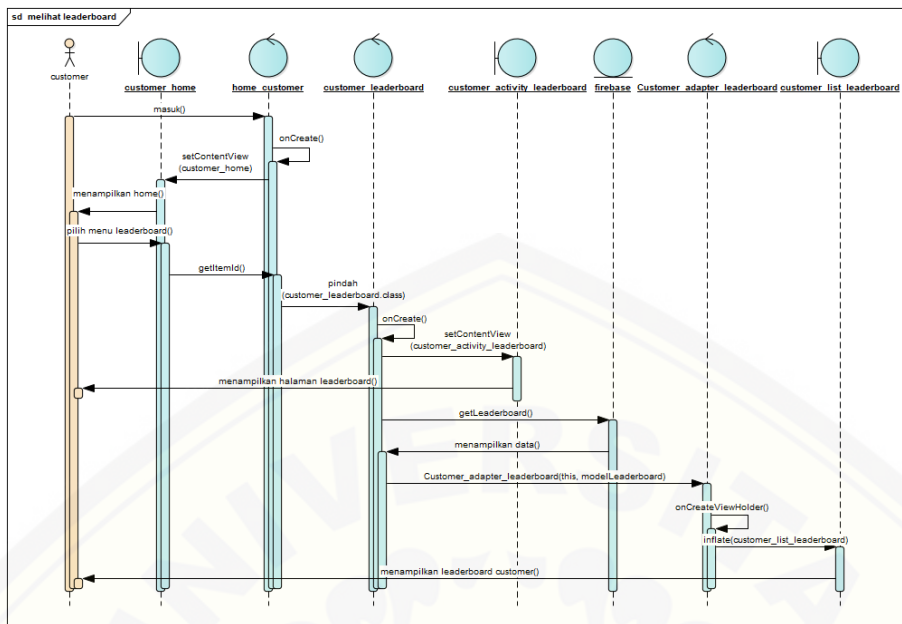






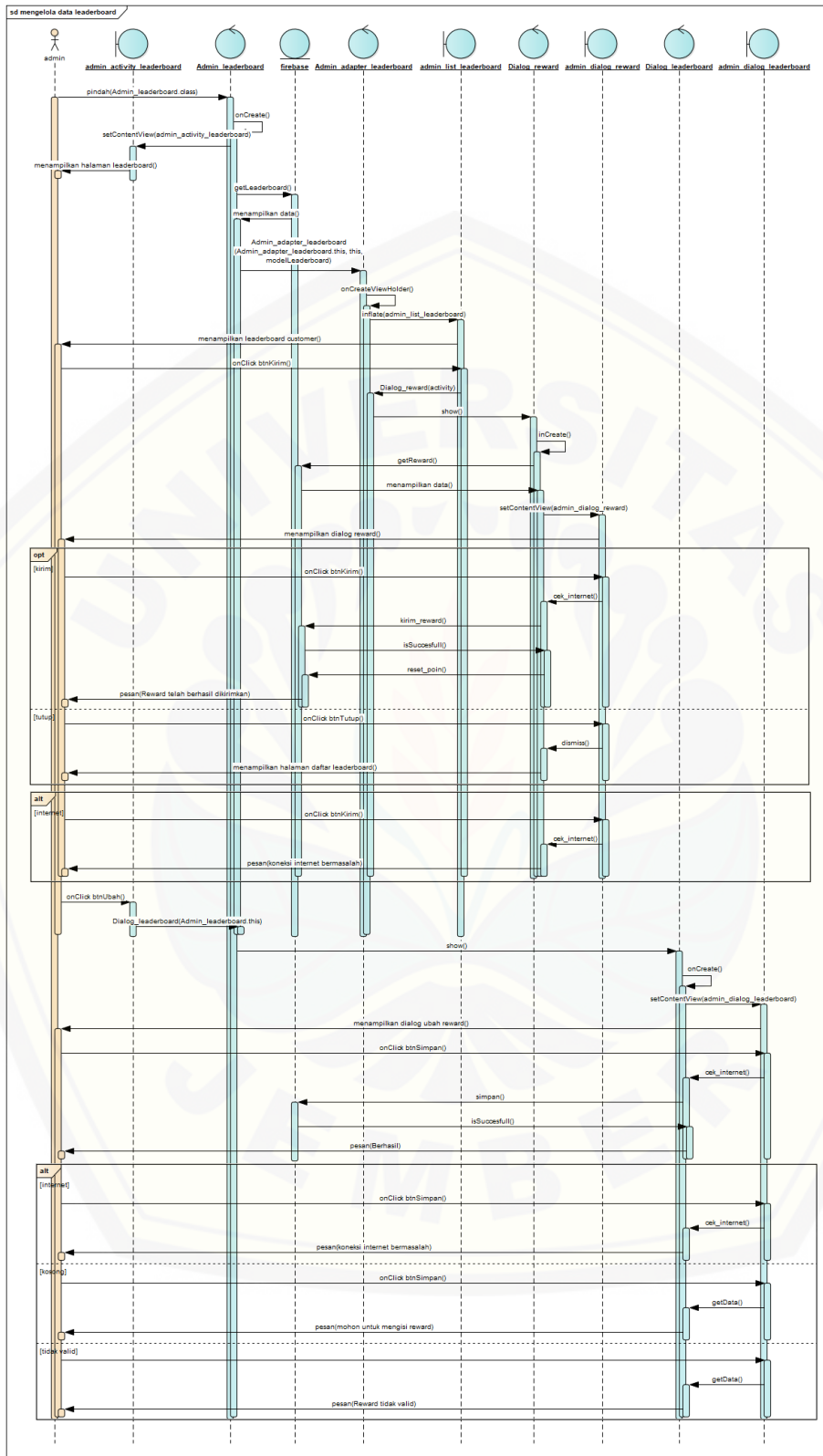


B.8. Melihat *leaderboard*

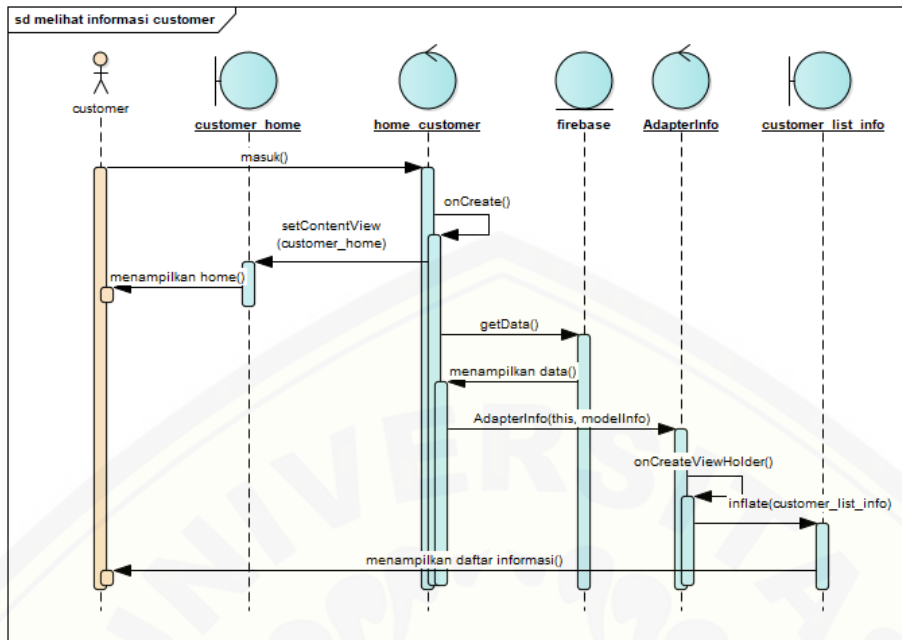


B.9. Mengelola data *leaderboard*

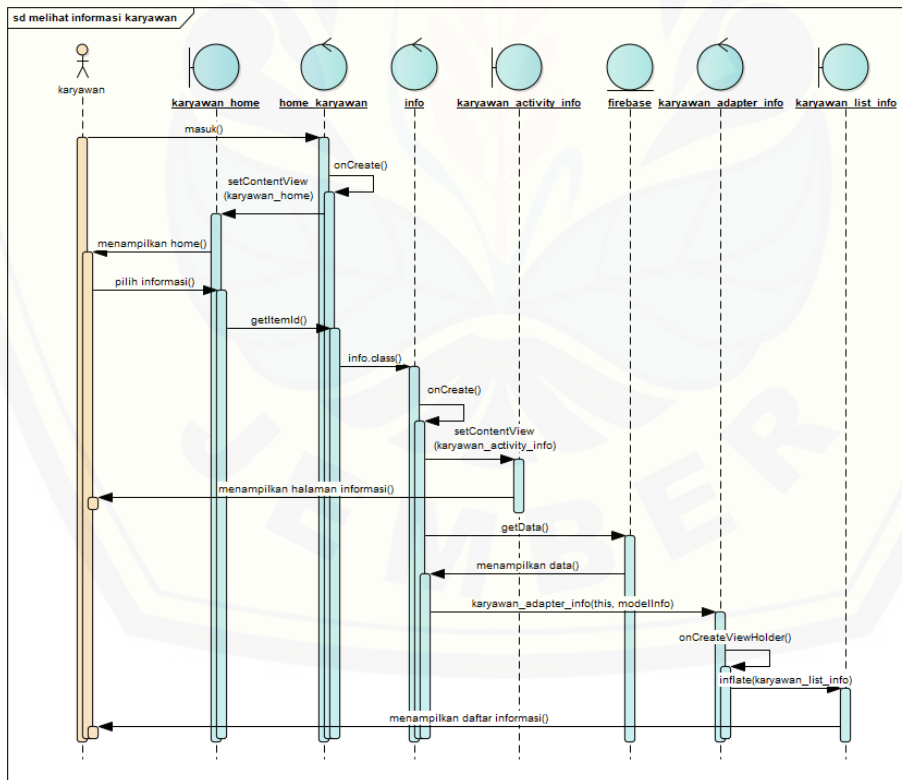




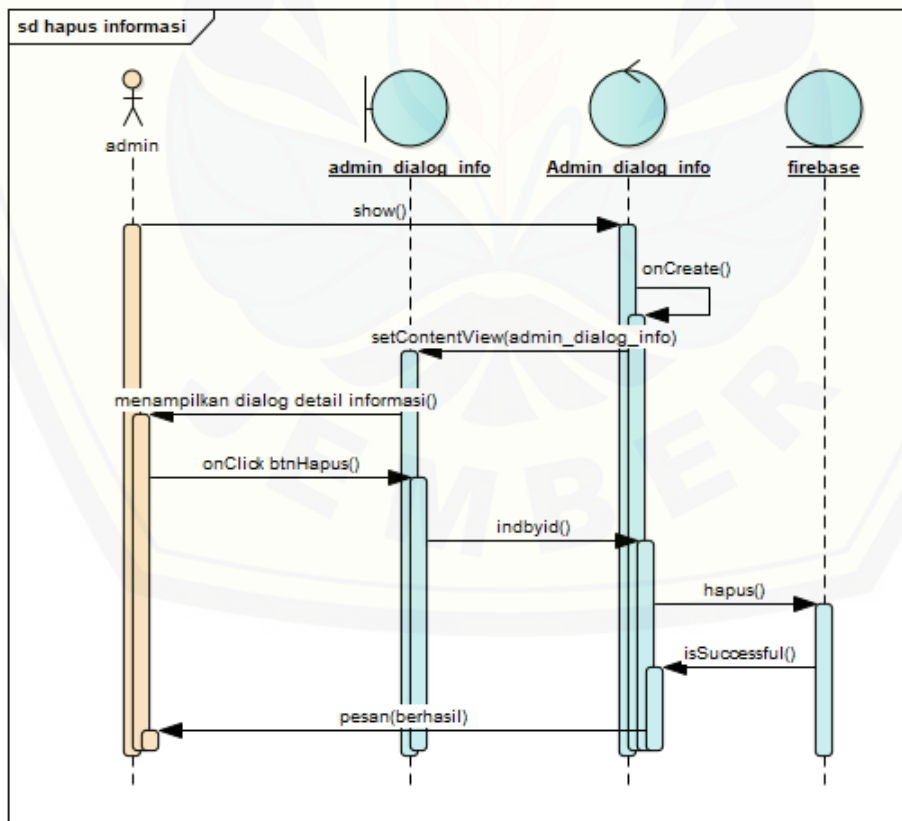
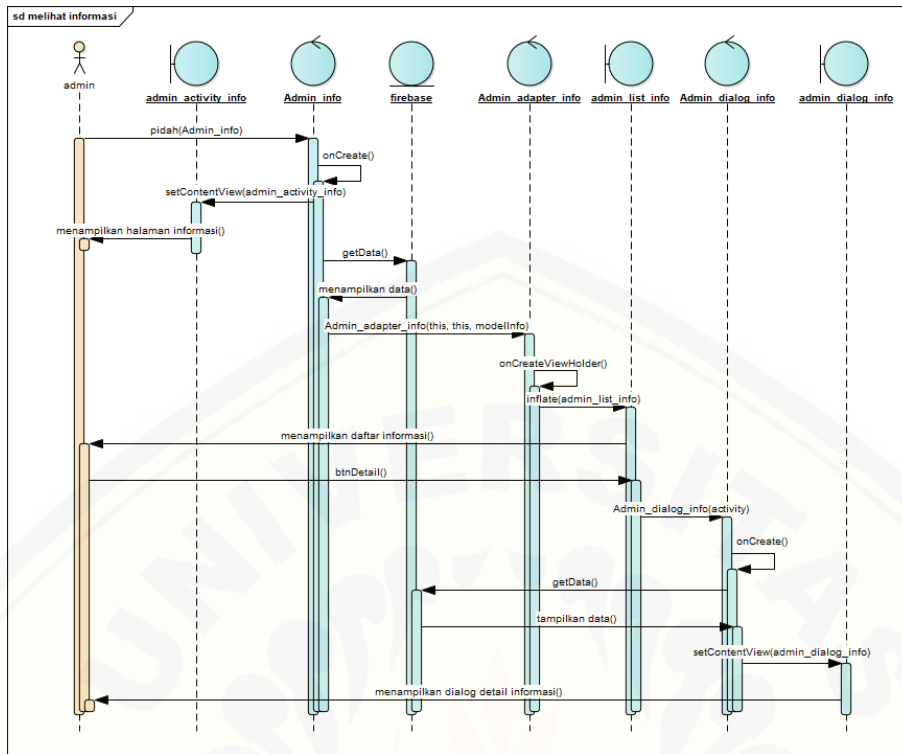
B.10. Melihat informasi *customer*

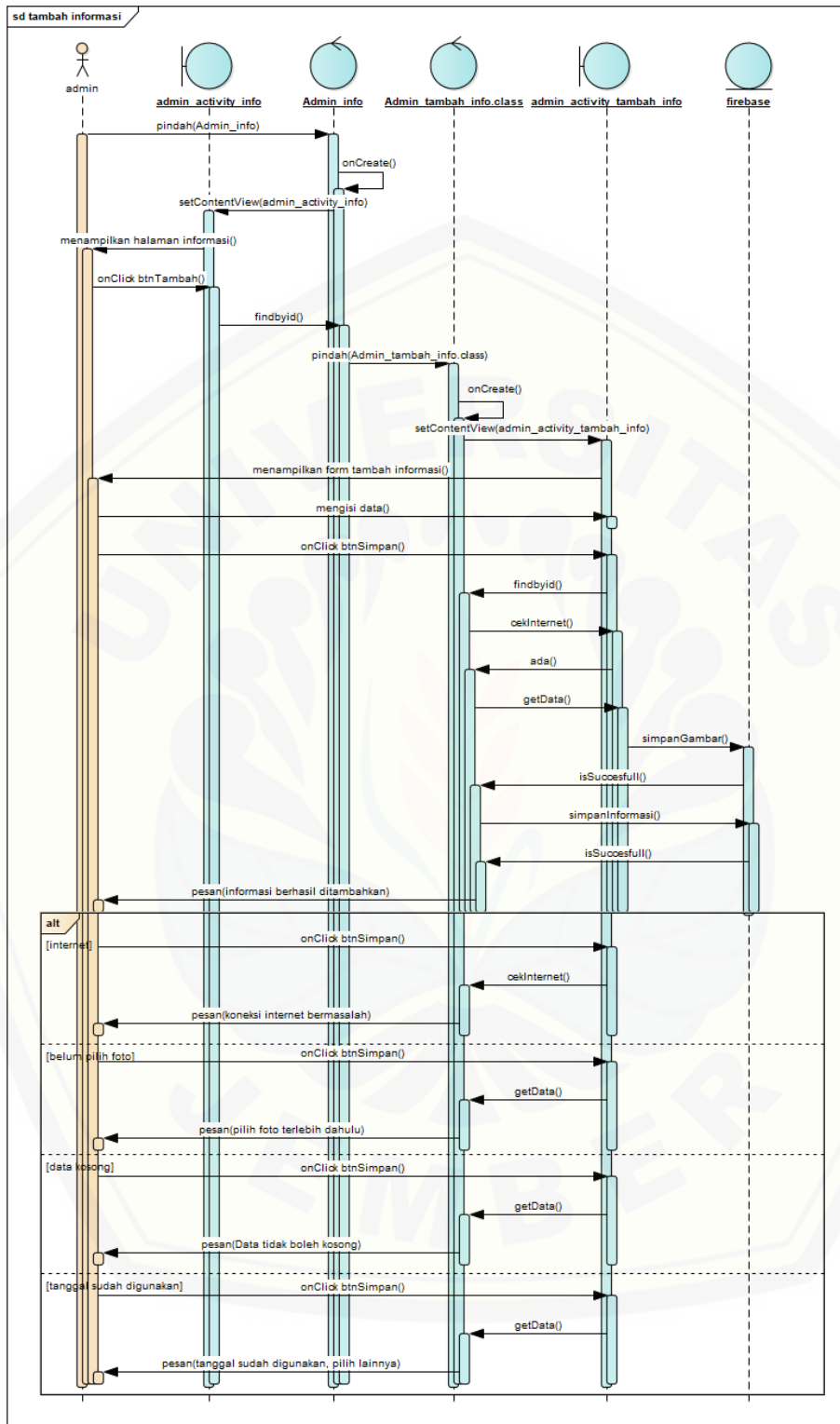


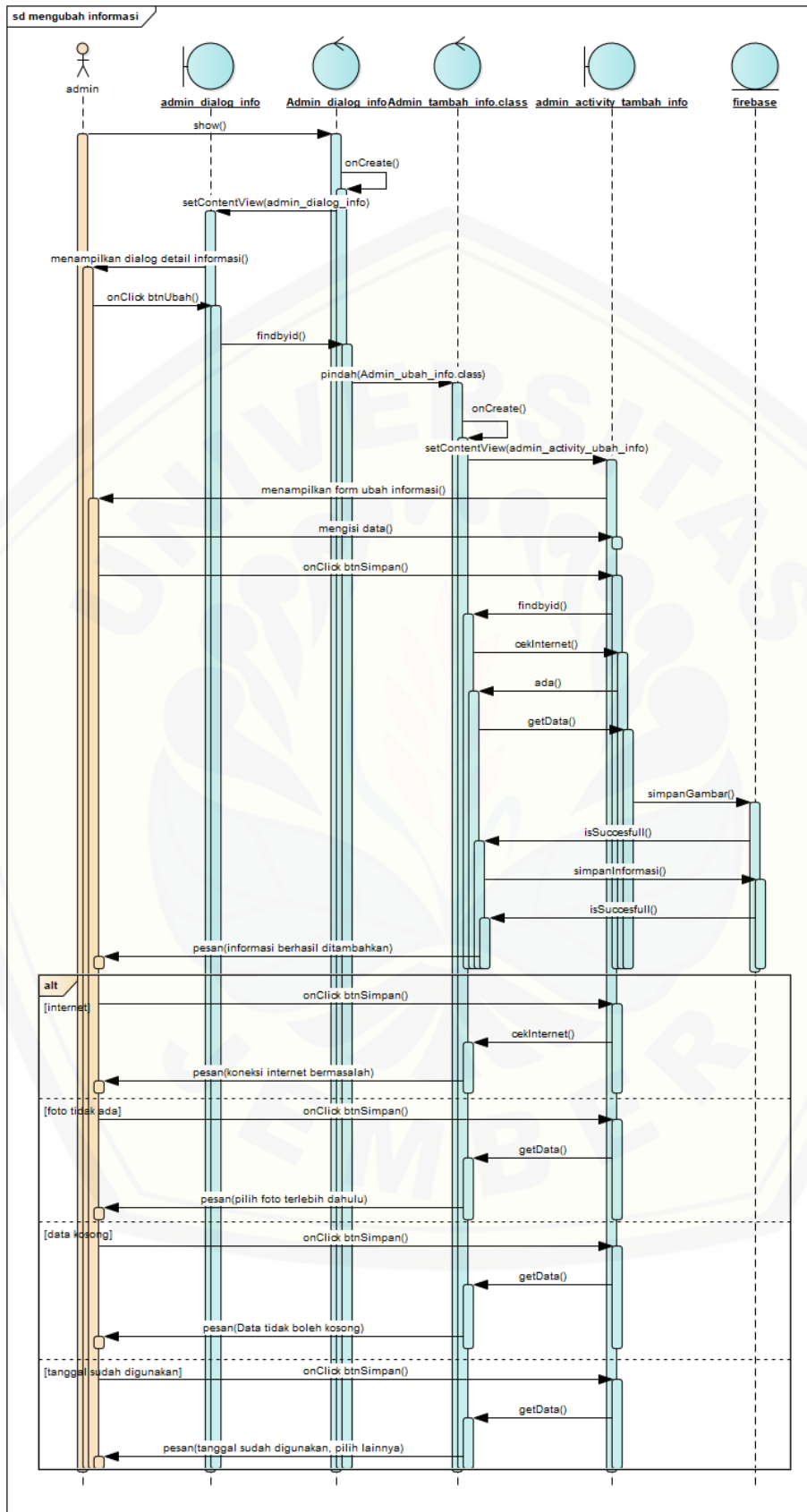
B.11. Melihat informasi karyawan



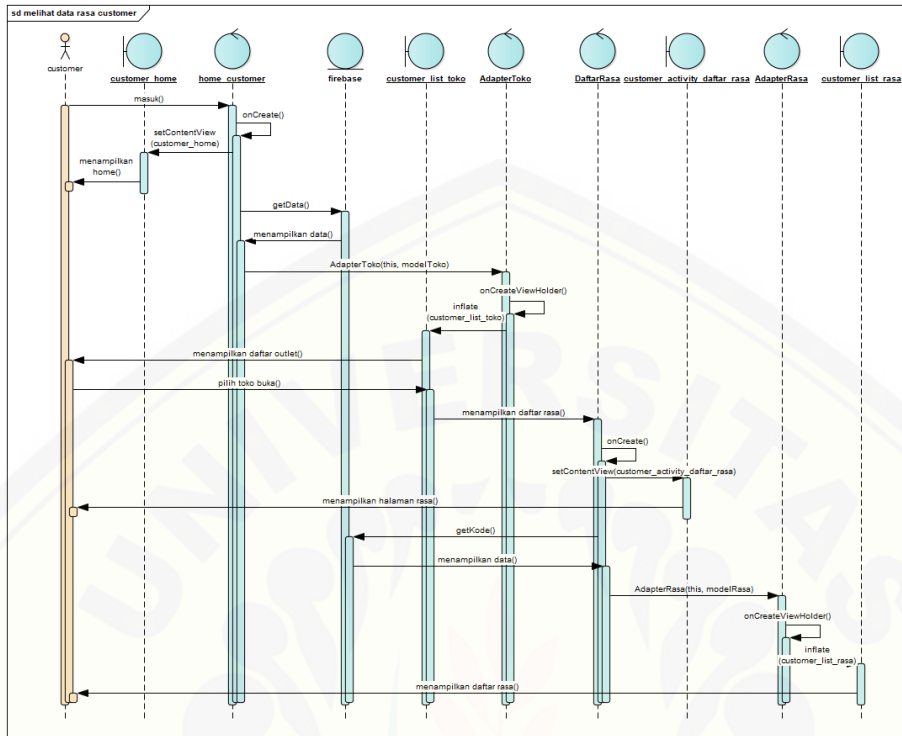
B.12. Mengelola informasi





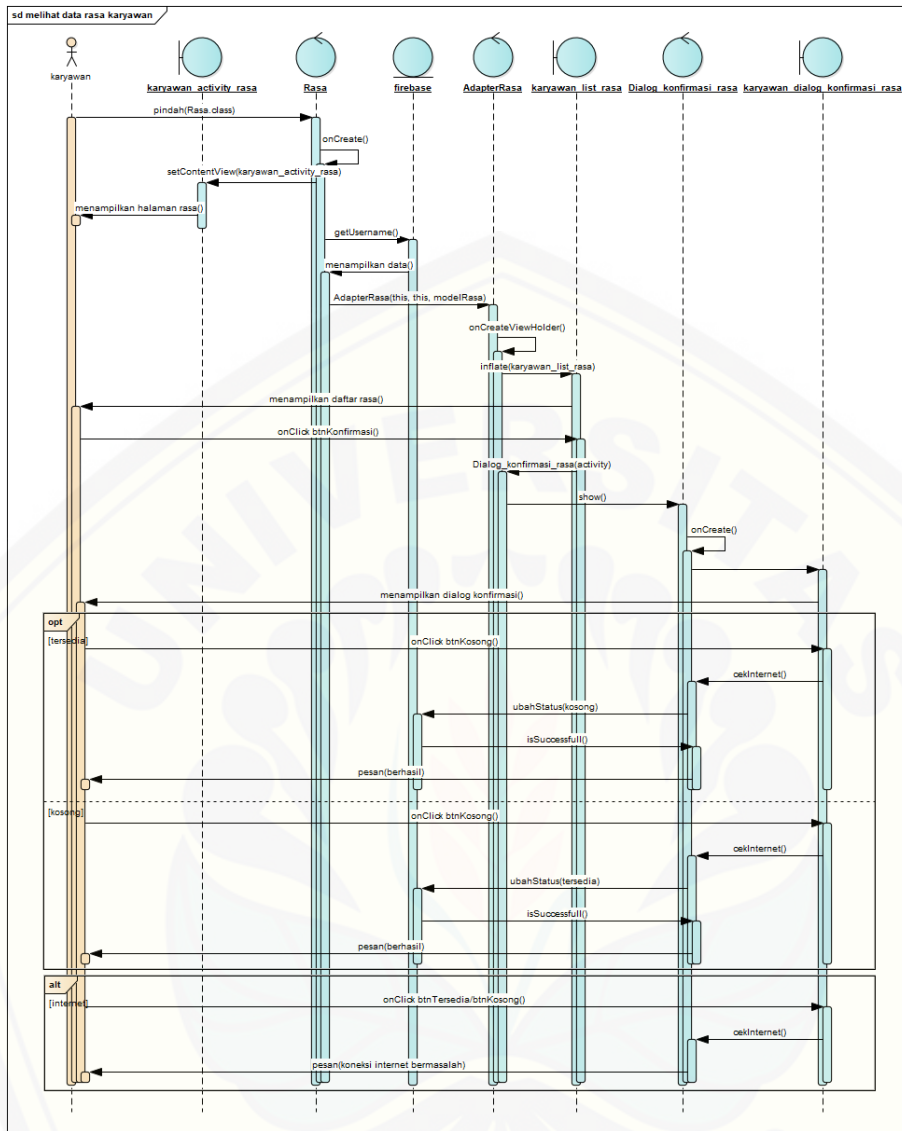


B.13. Melihat data rasa *customer*

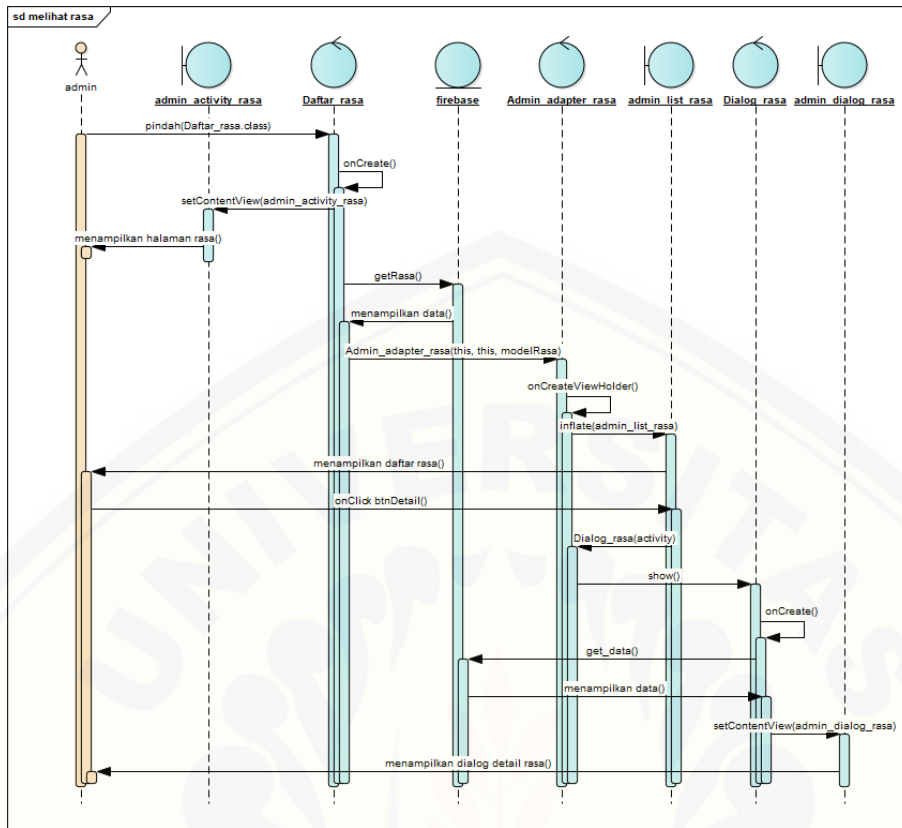


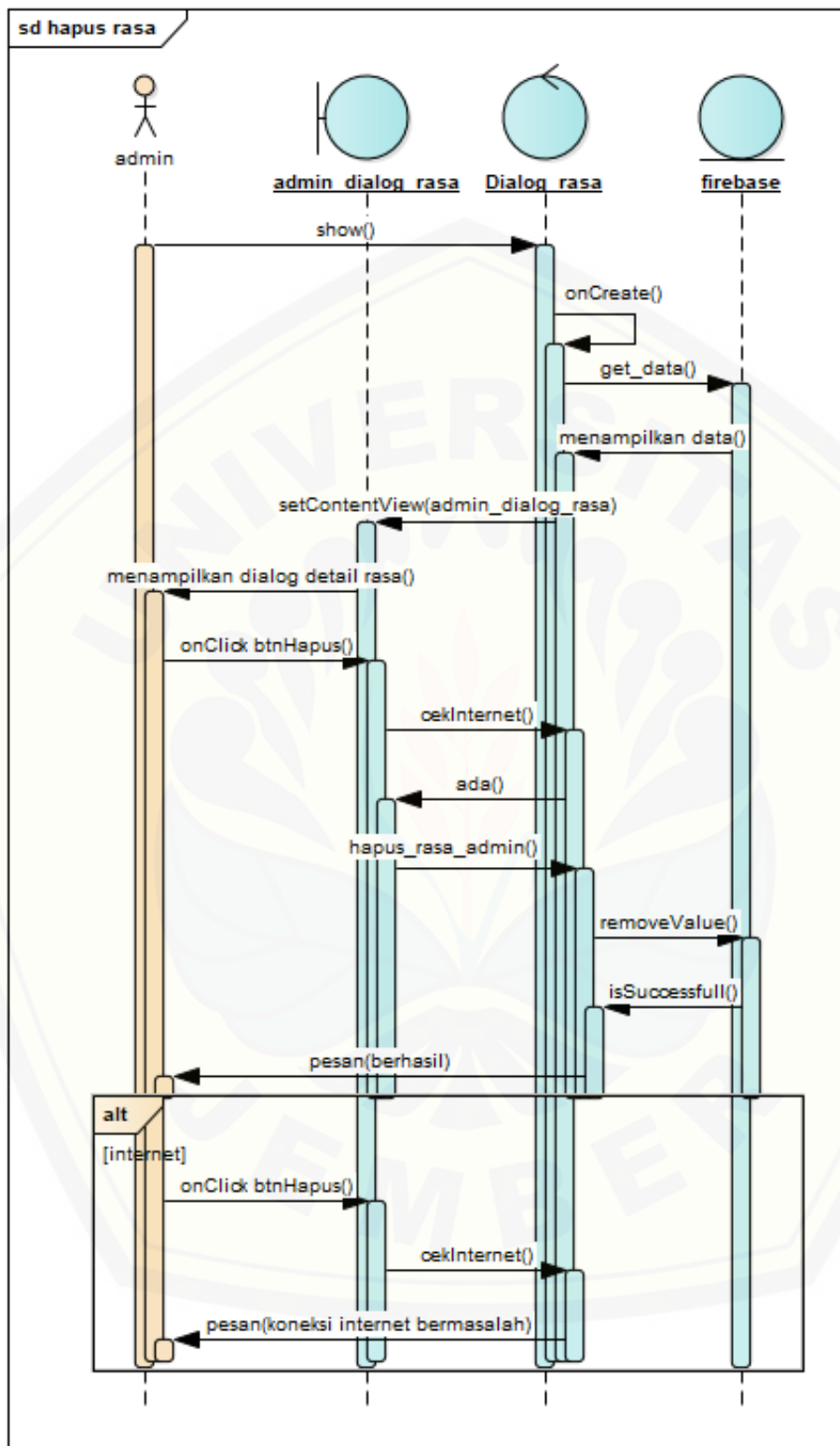
B.14. Melihat data rasa karyawan

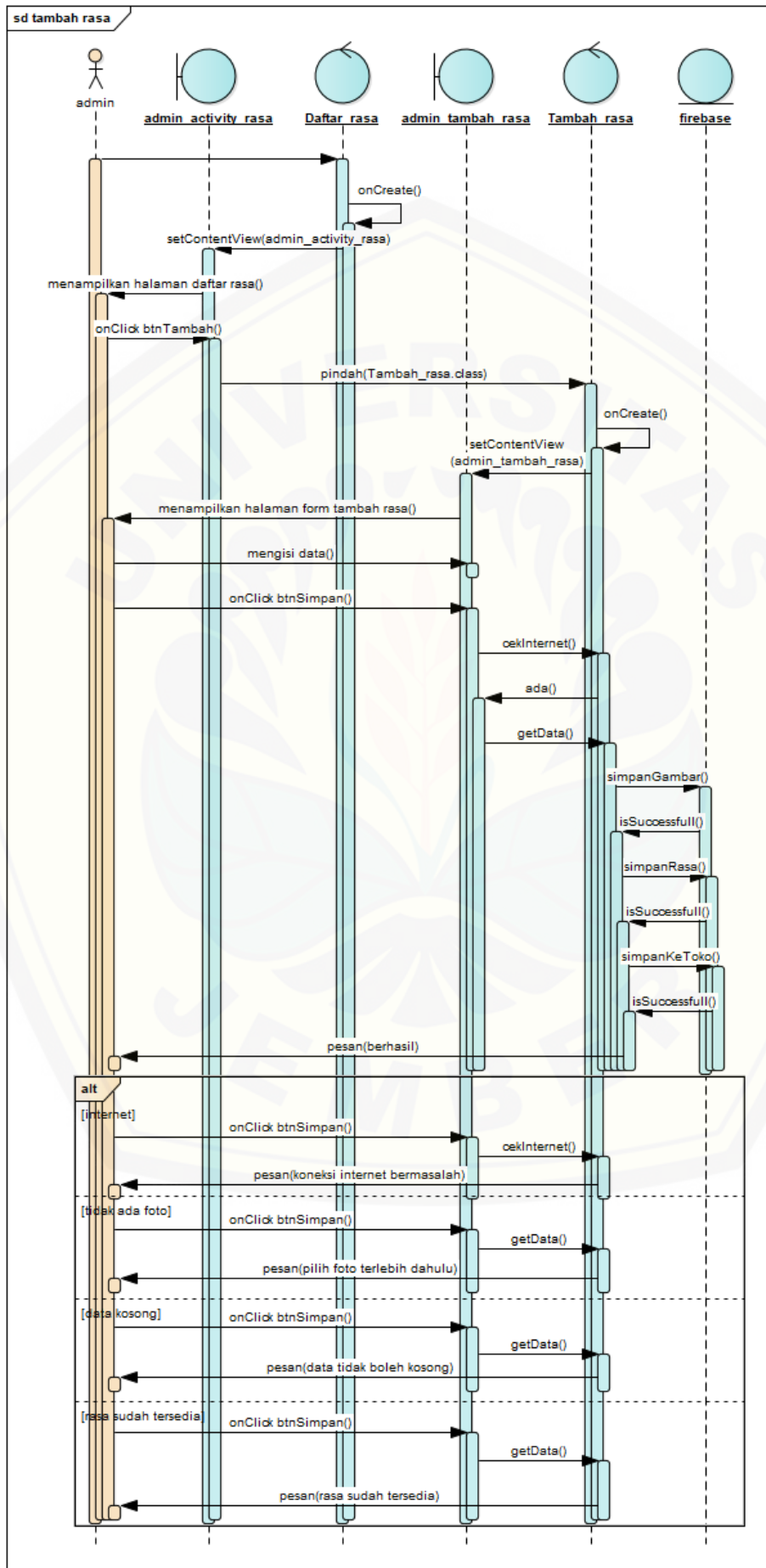


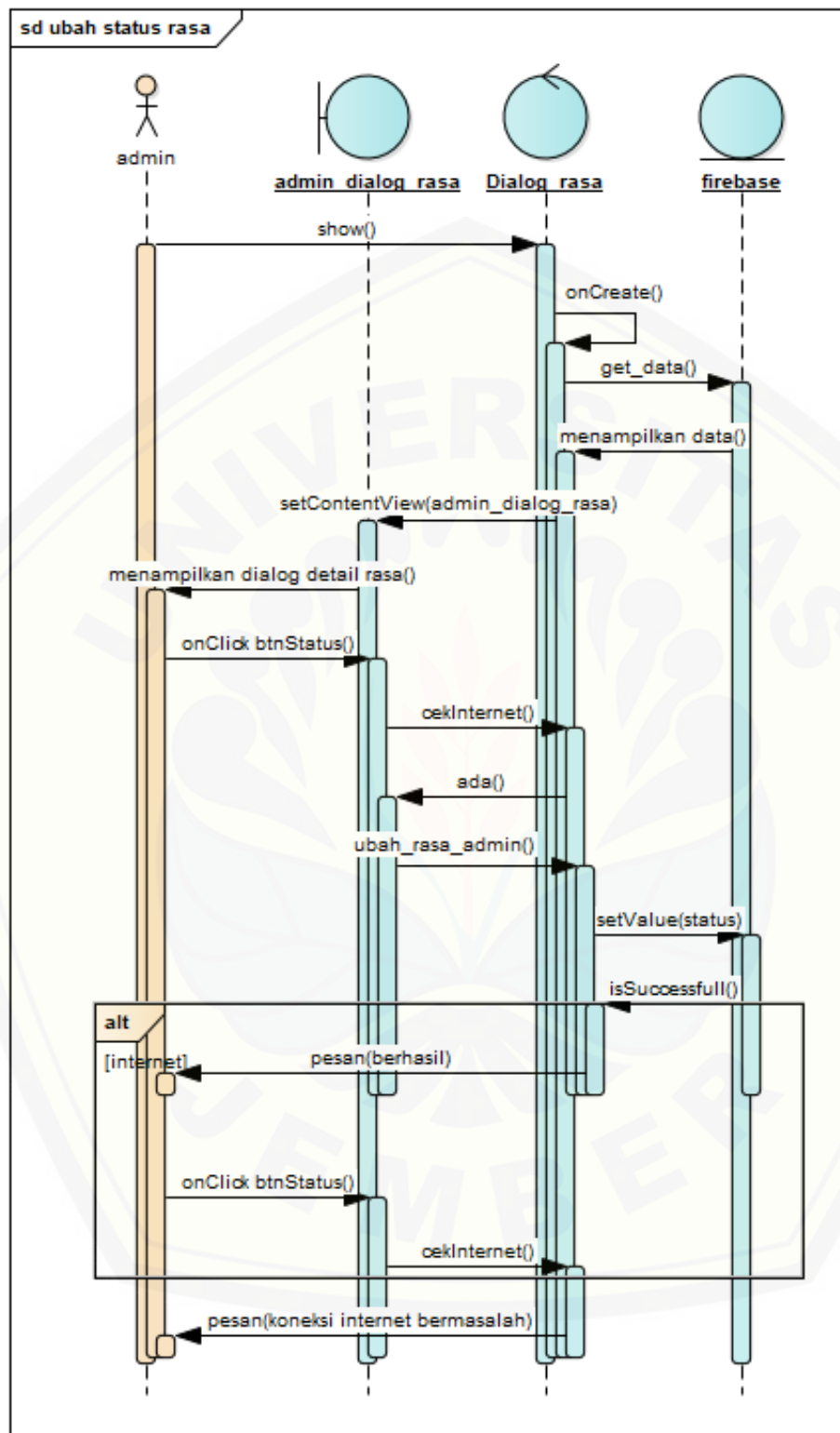


B.15. Mengelola data rasa

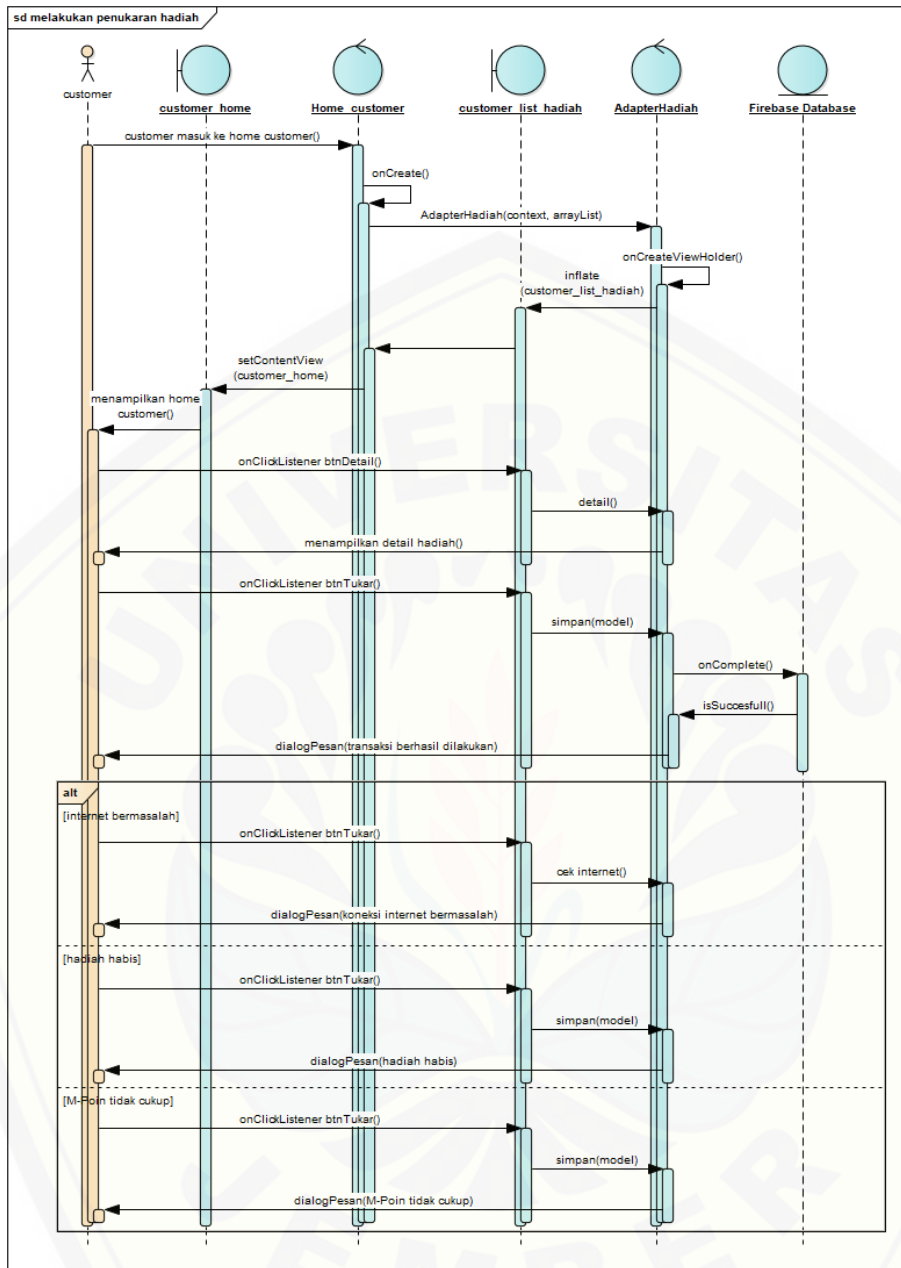






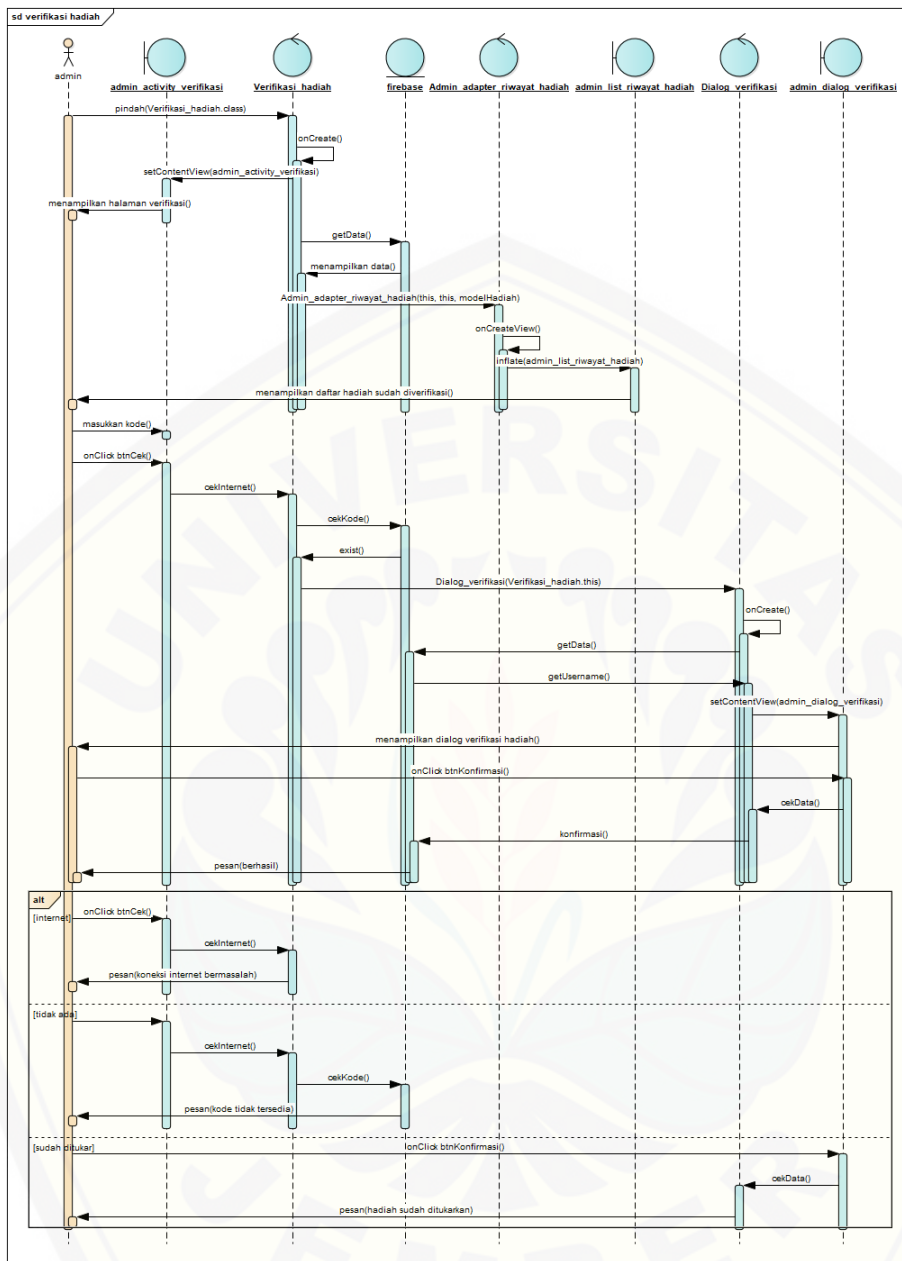


B.16. Melakukan penukaran hadiah

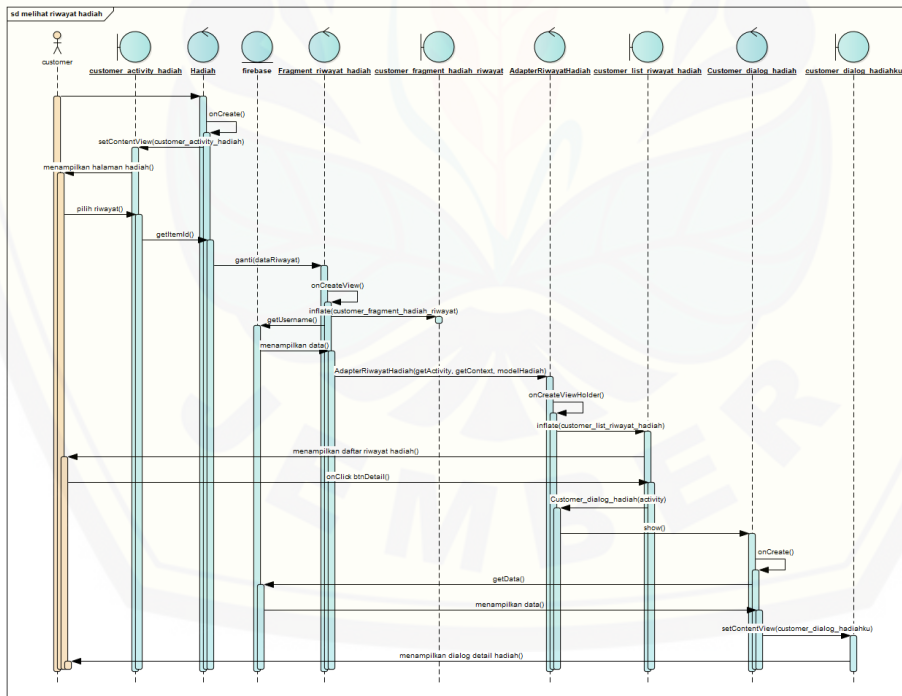
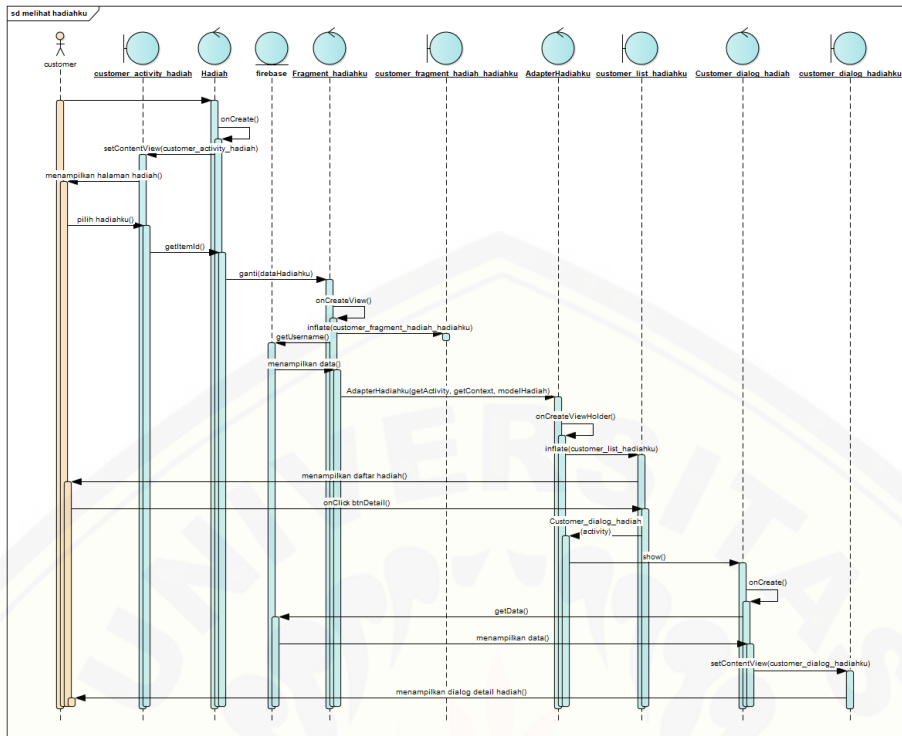


B.17. Verifikasi hadia

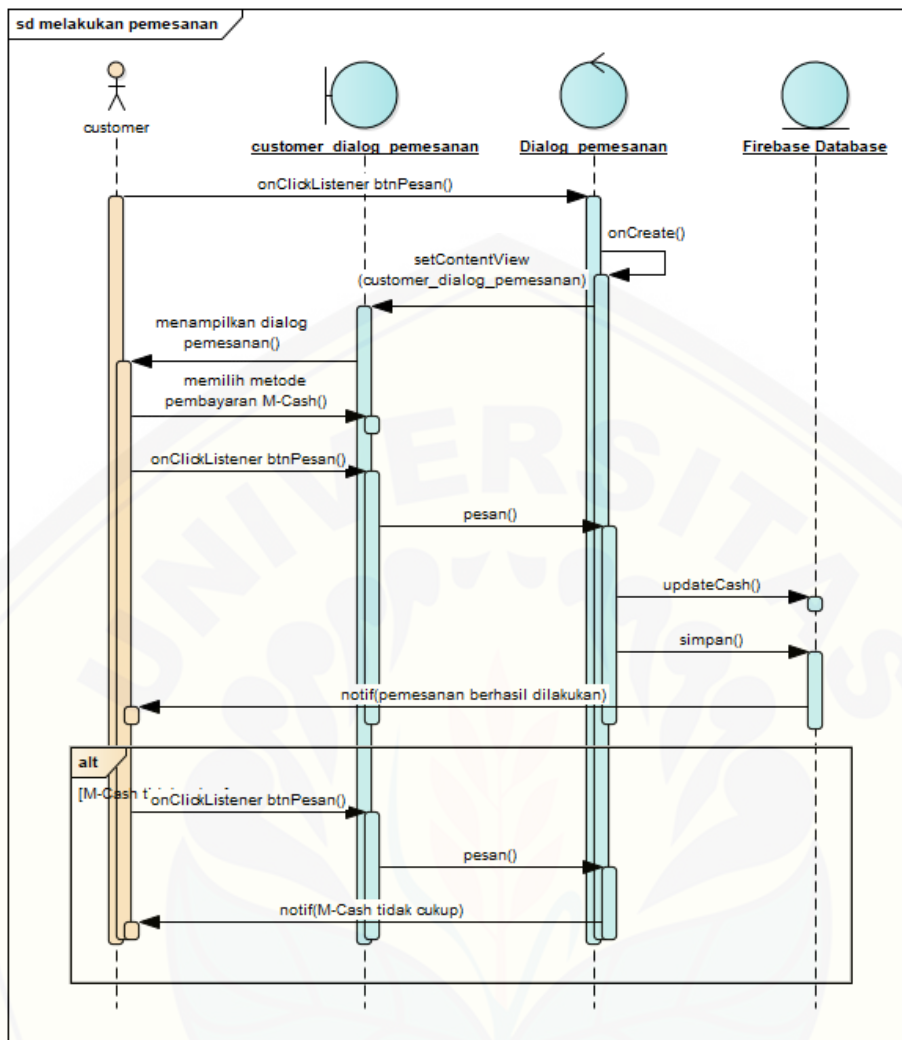




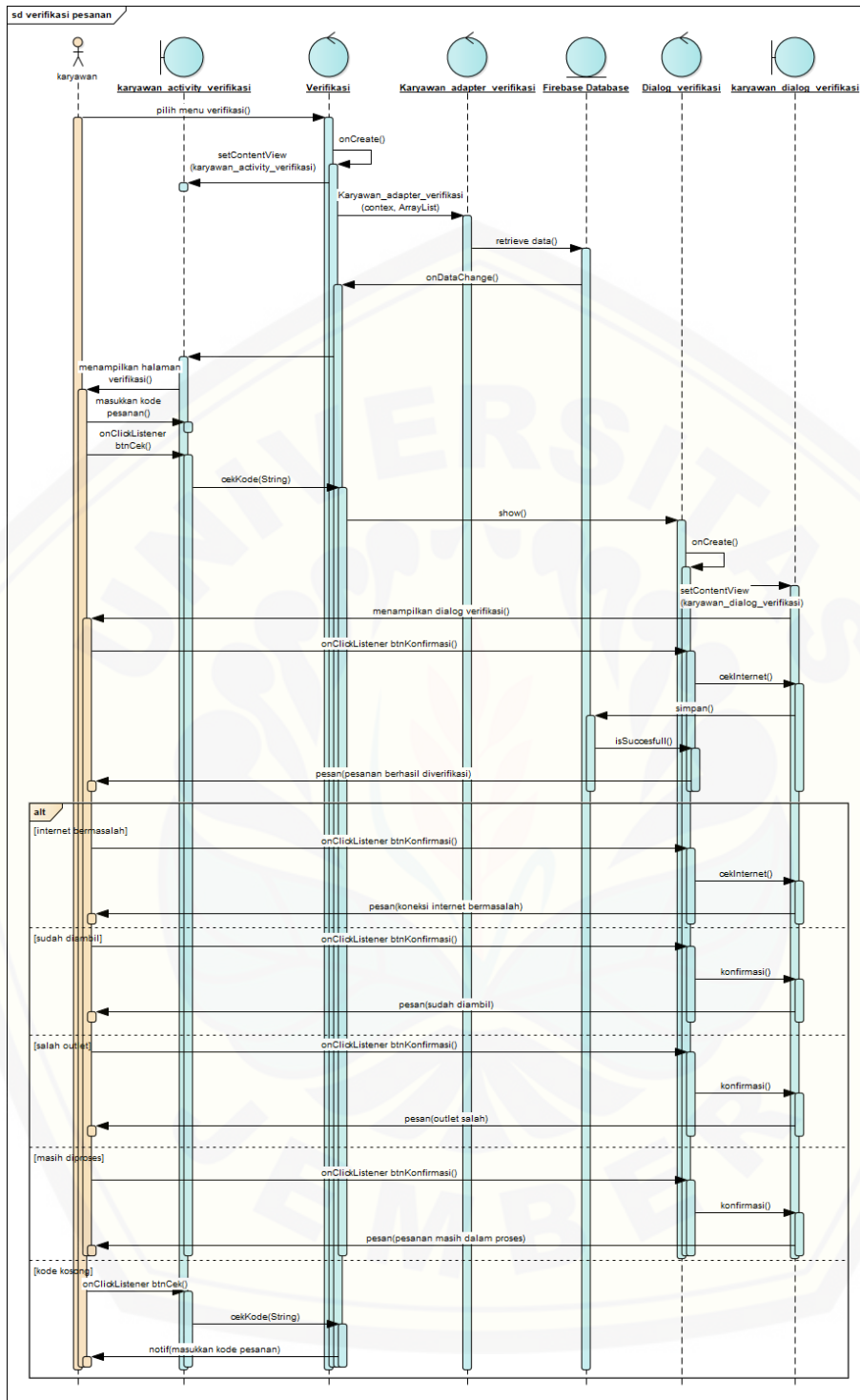
B.18. Melihat data hadiah



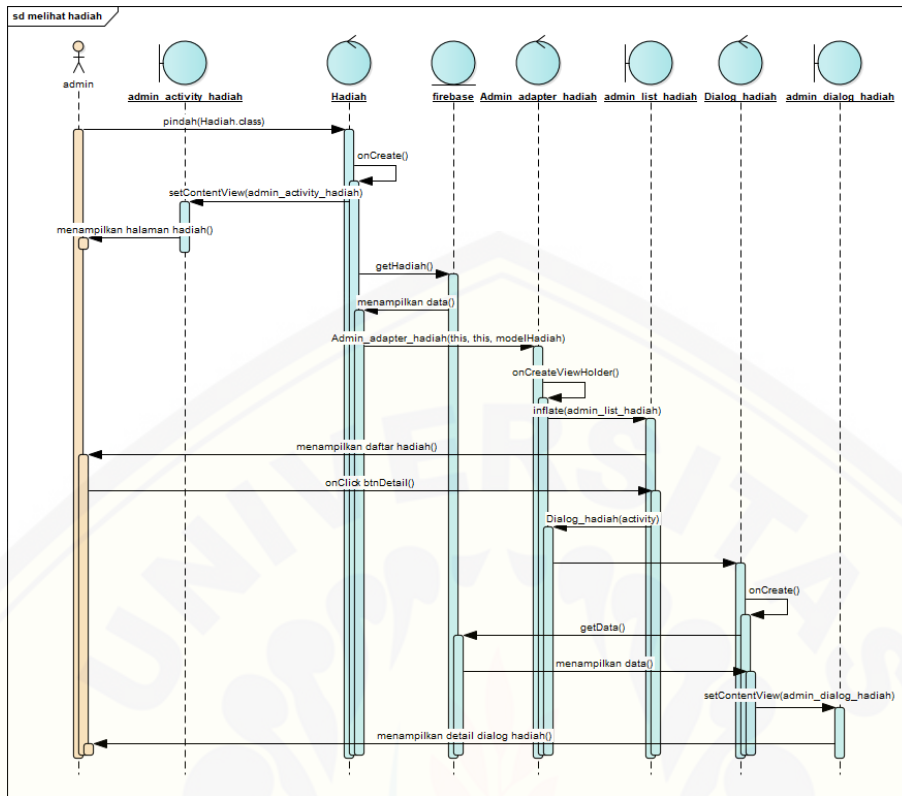
B.19. Melakukan pemesanan

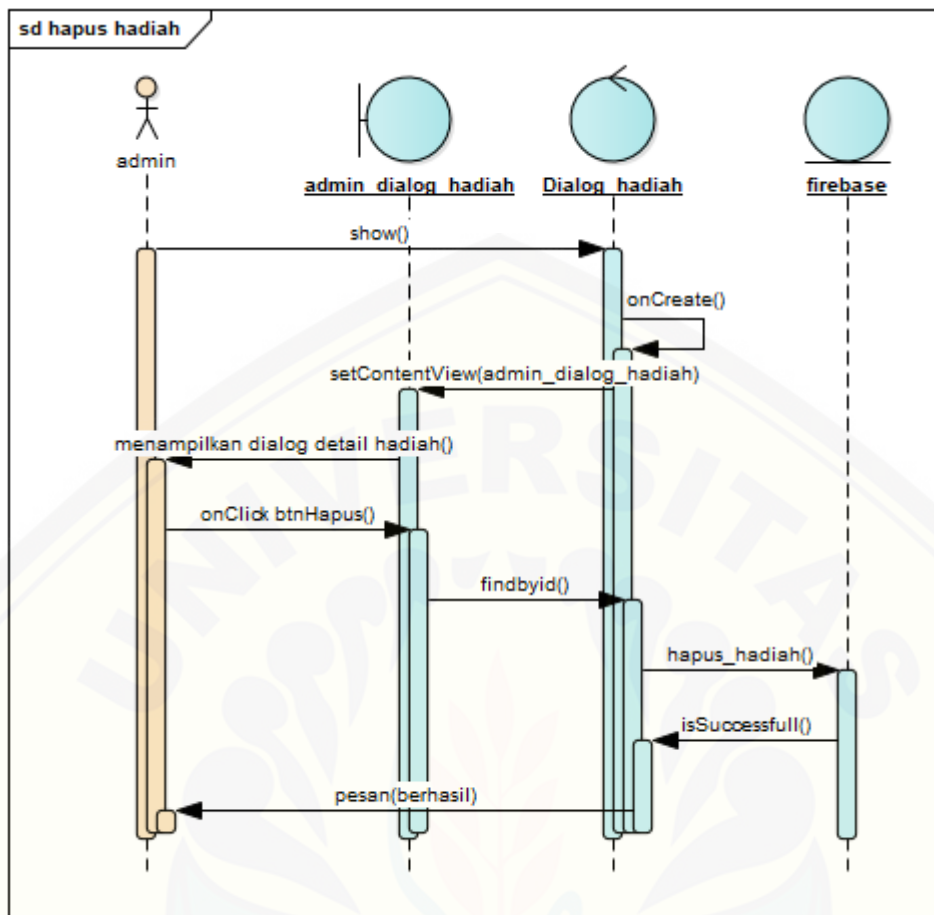


B.20. Verifikasi pesanan

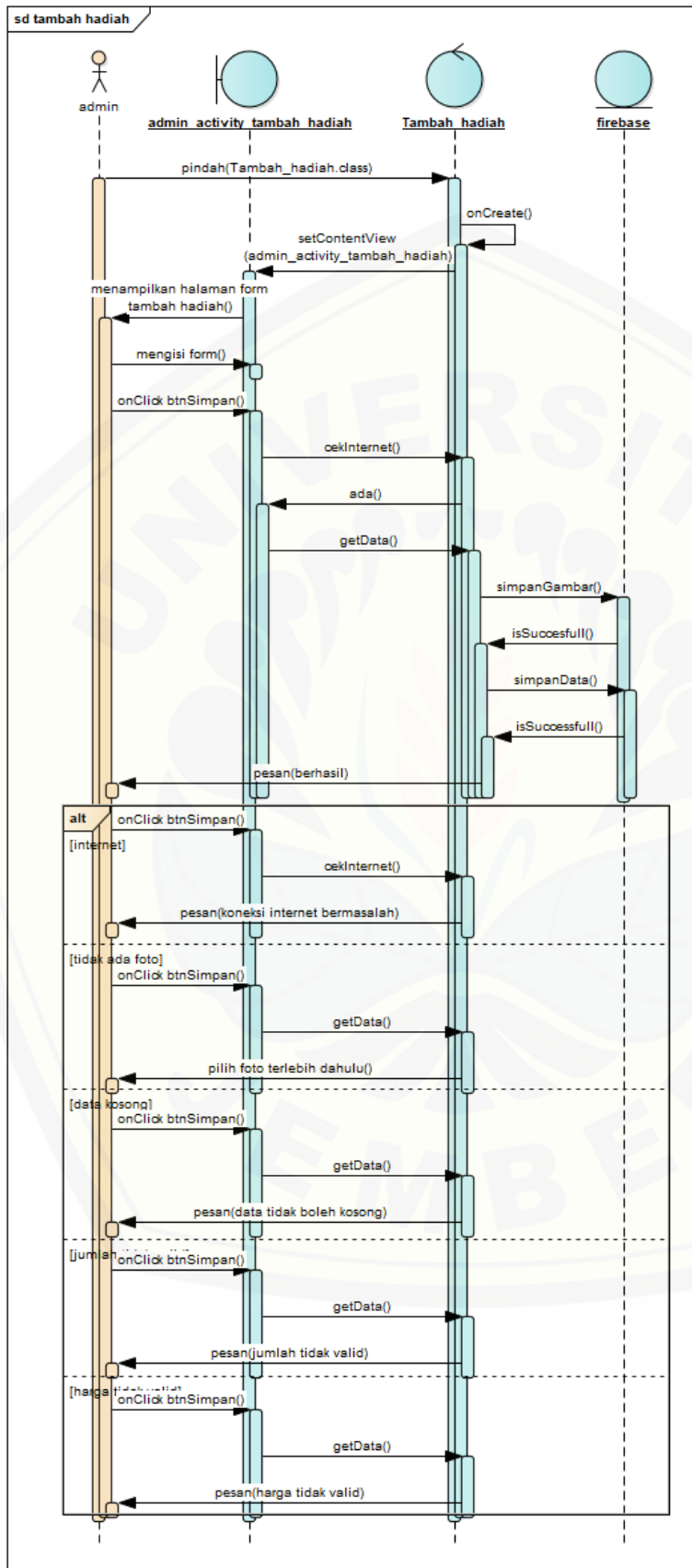


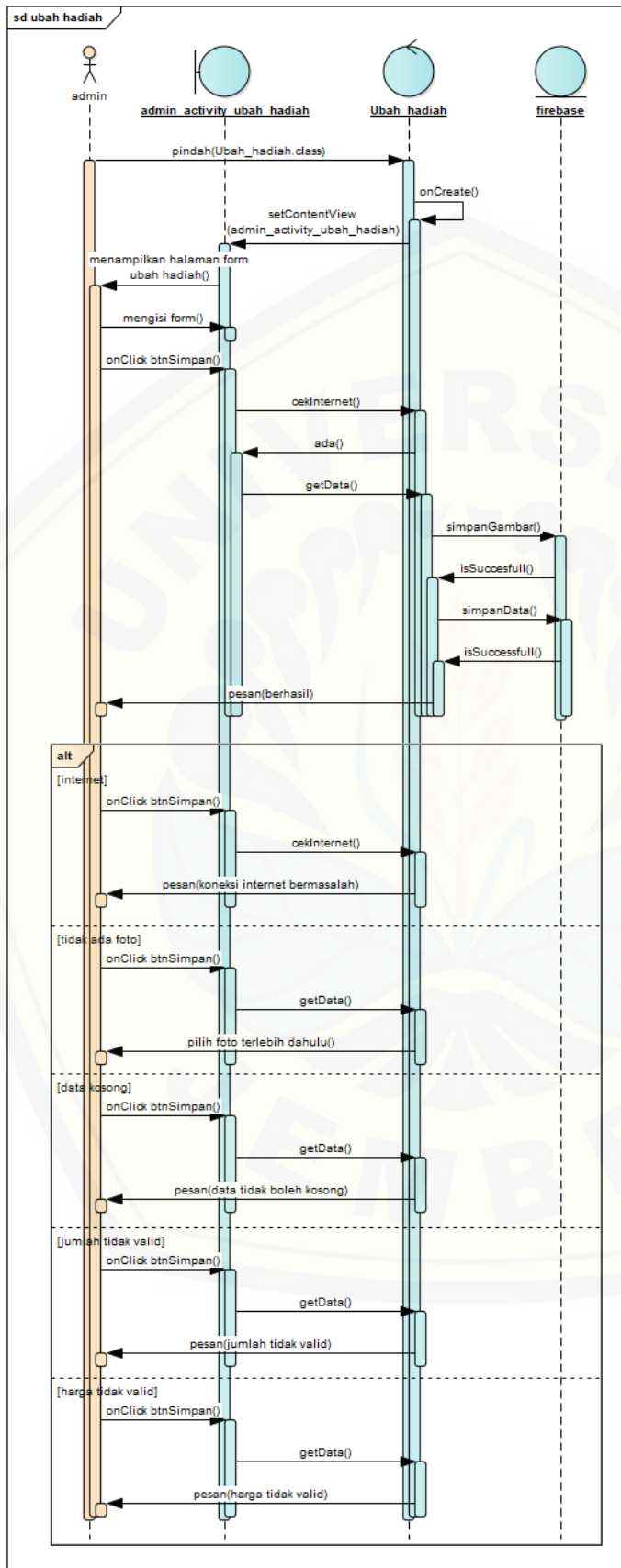
B.21. Mengelola data hadiah

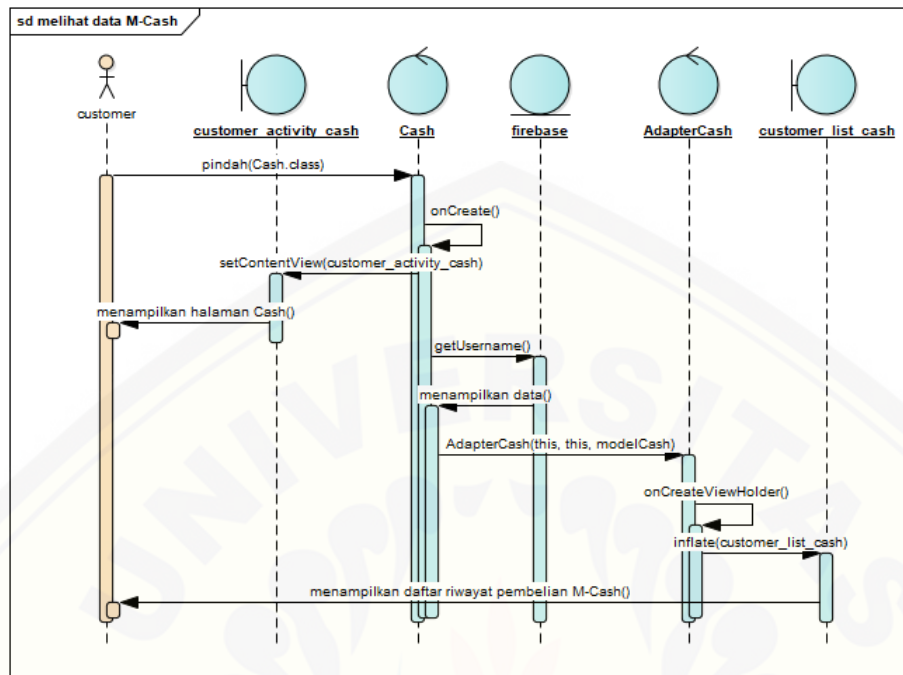


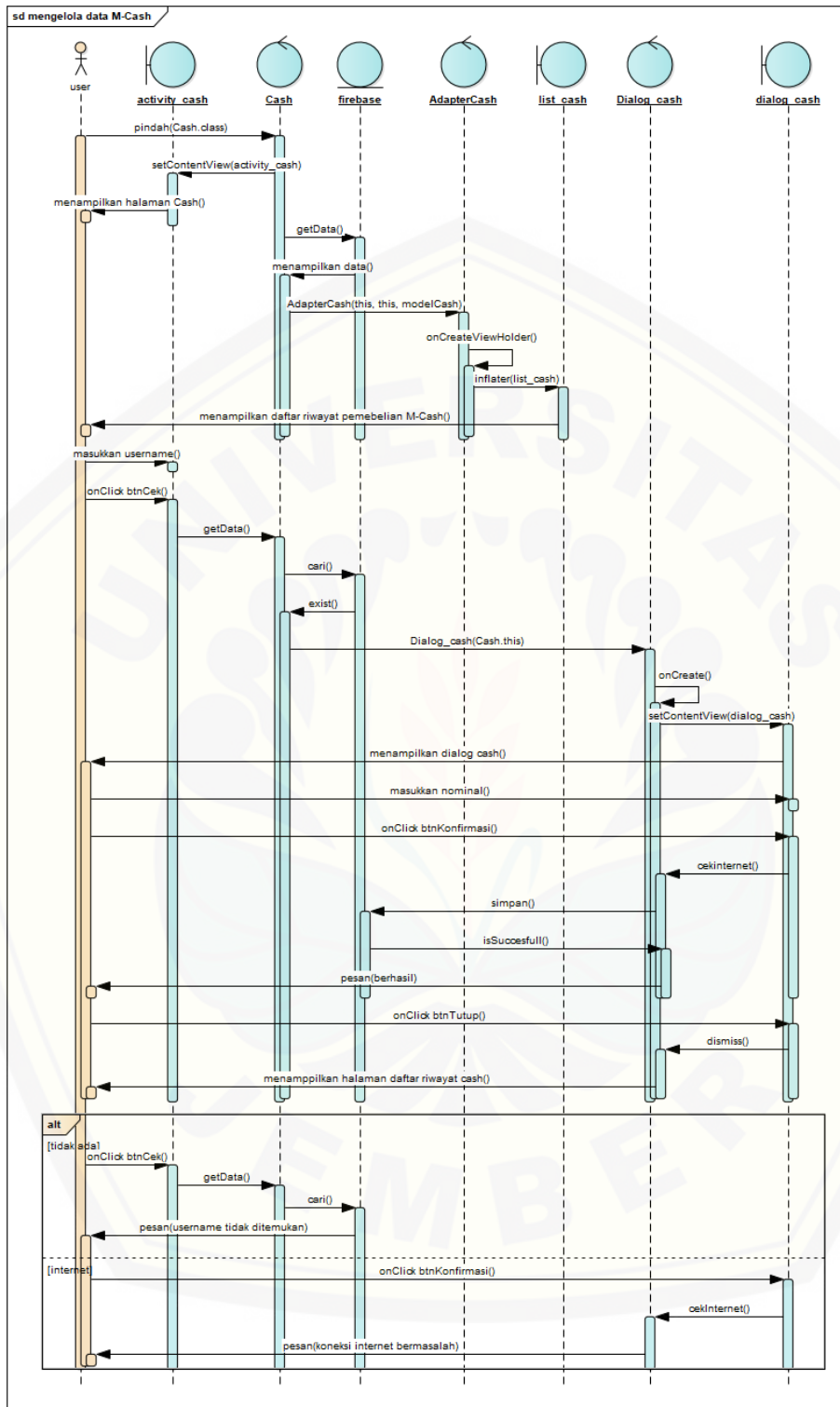




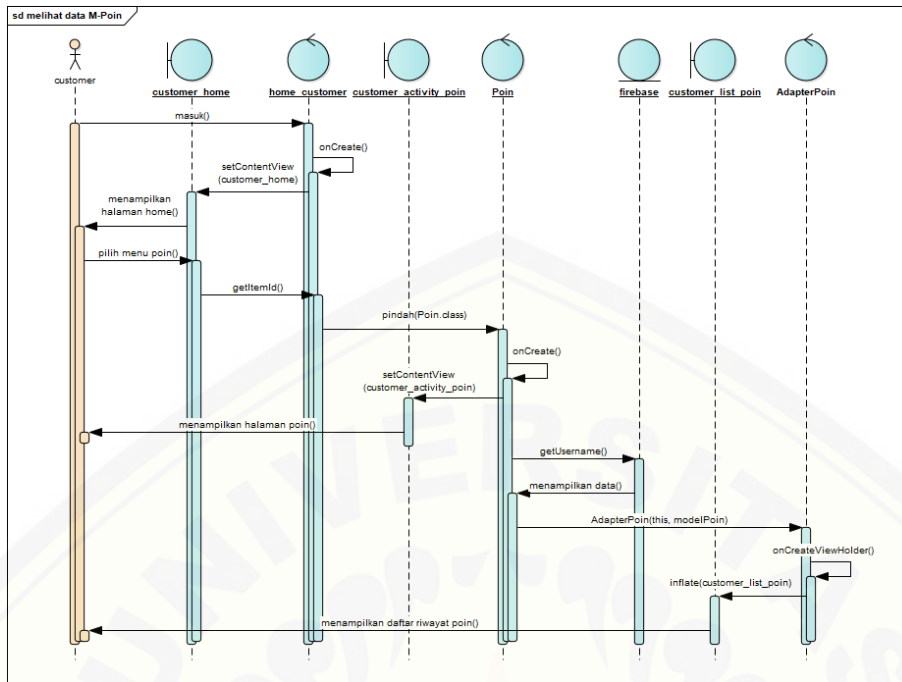




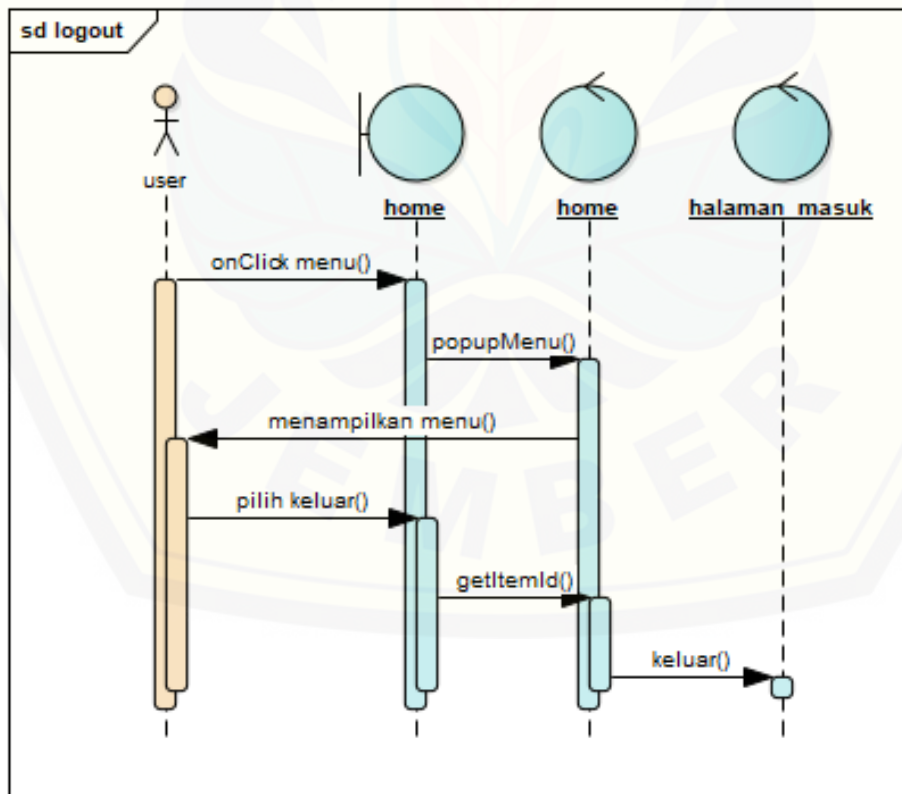
B.22. Melihat data *M-Cash*B.23. Mengelola data *M-Cash*



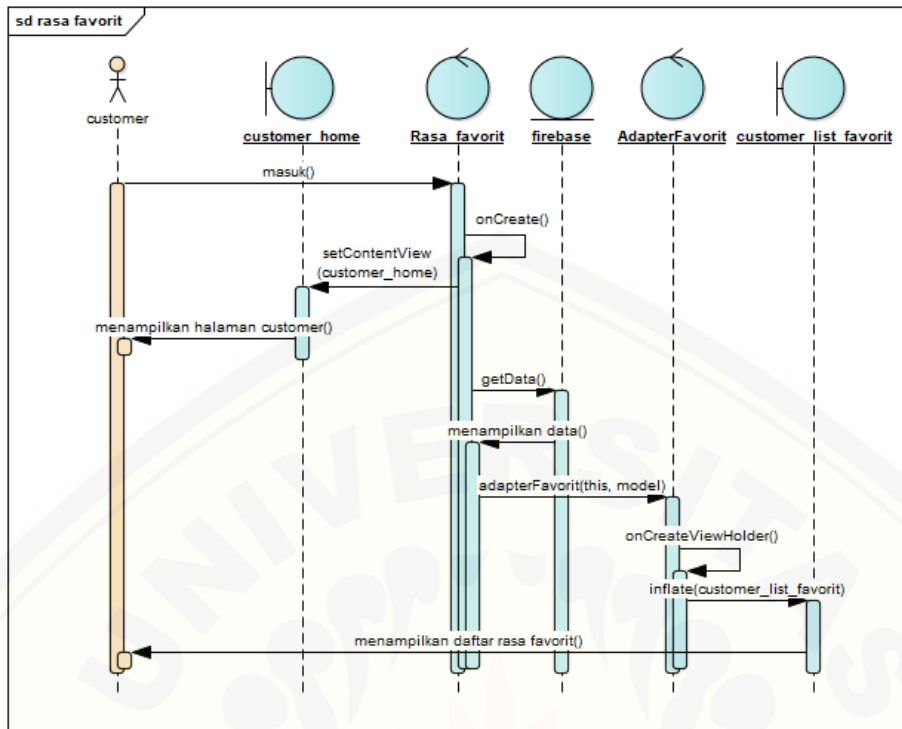
B.24. Melihat data *M-Poin*



B.25. Logout



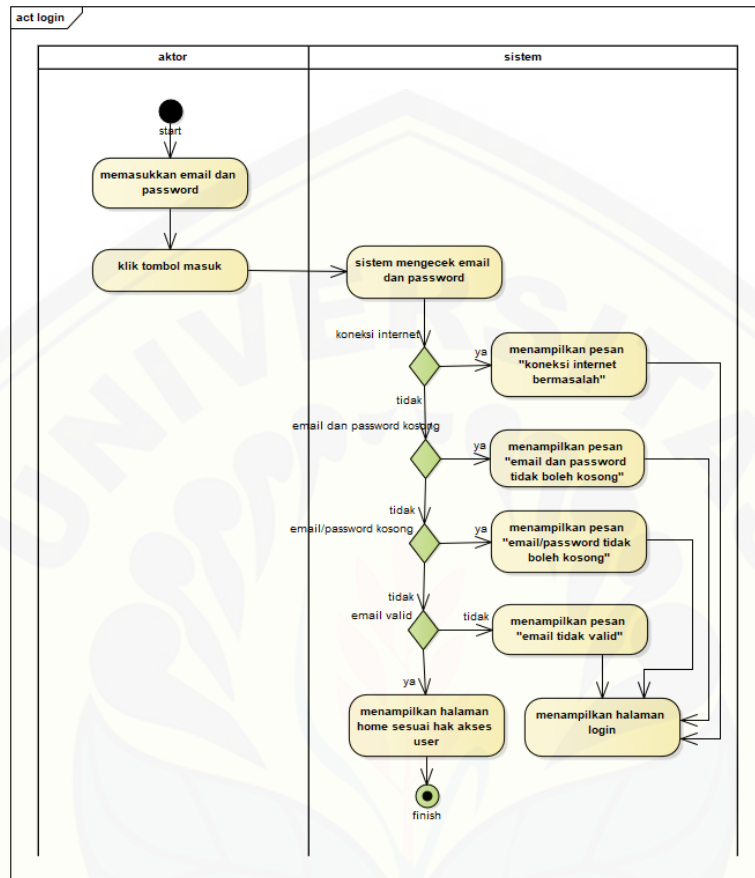
B.26. Rasa favorit



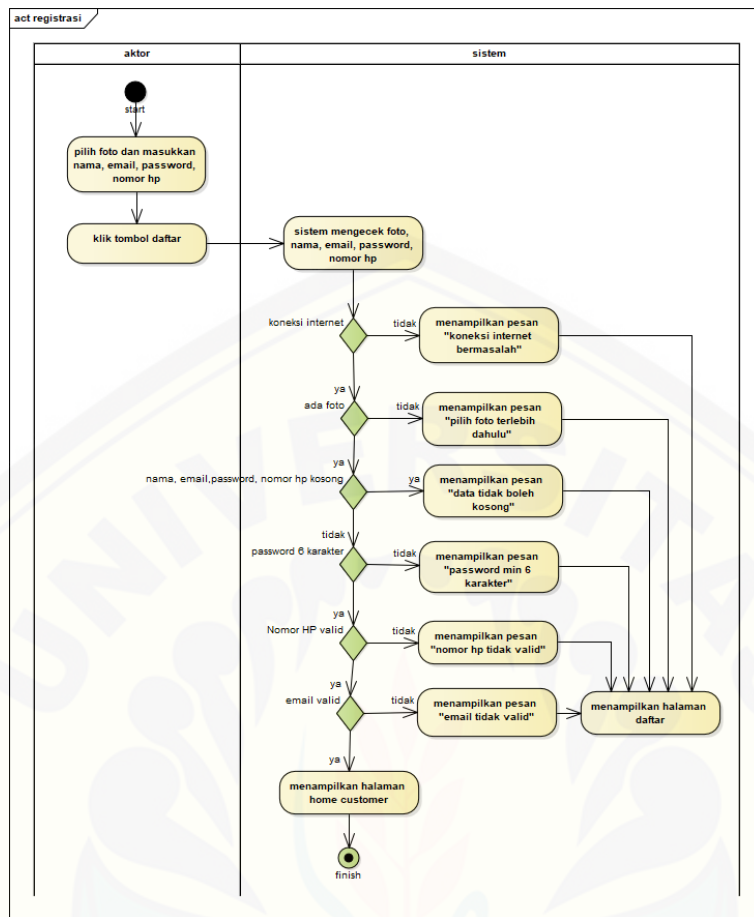


C. Activity diagram

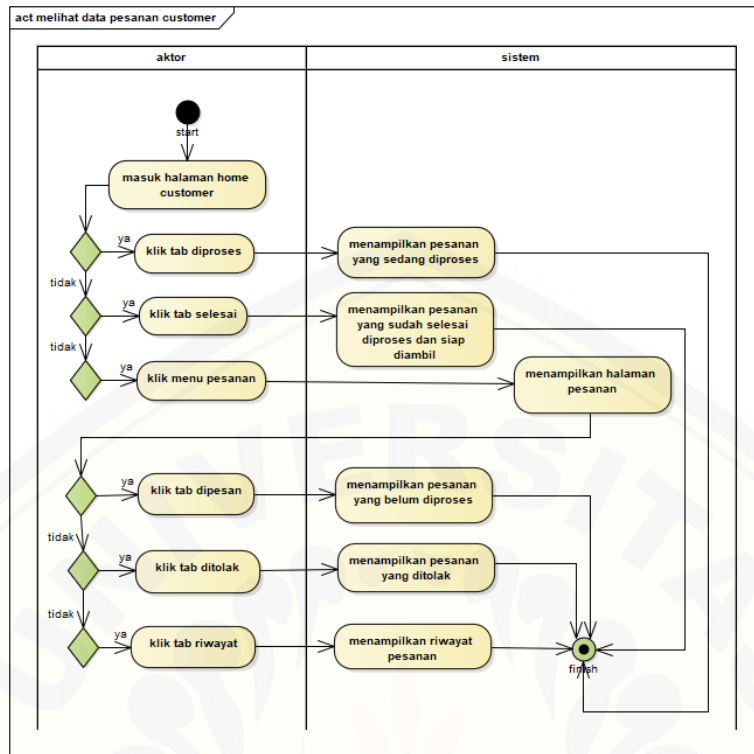
C.1. Login



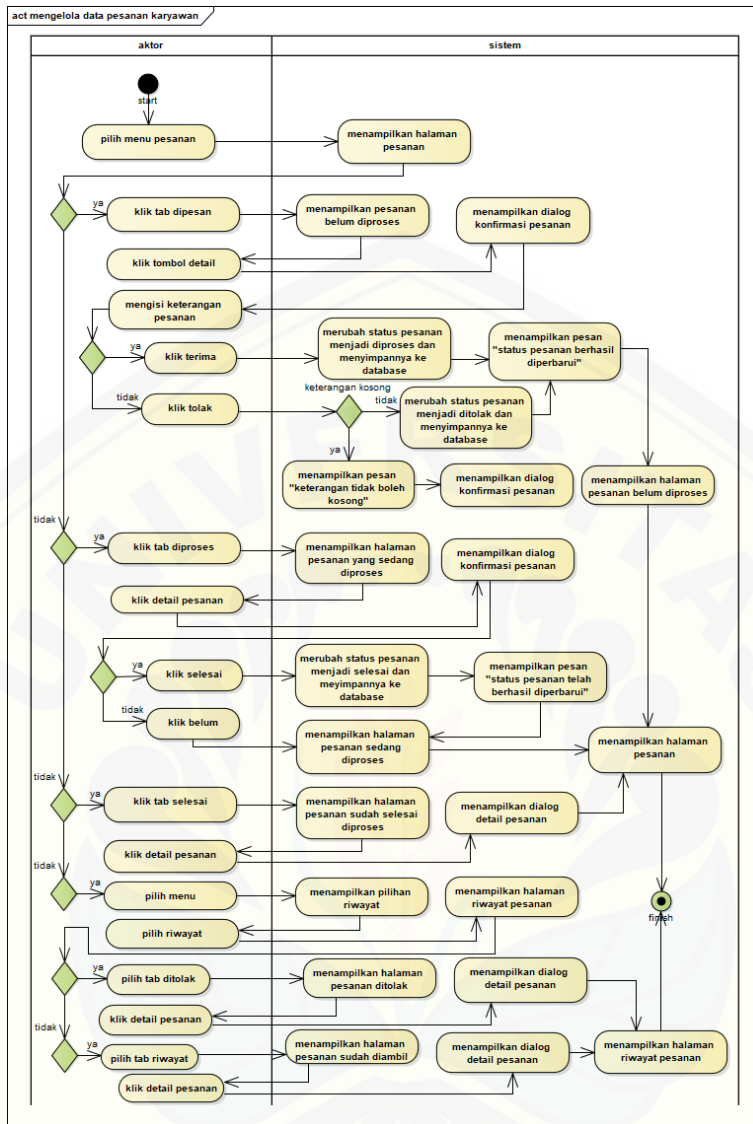
C.2. Registrasi



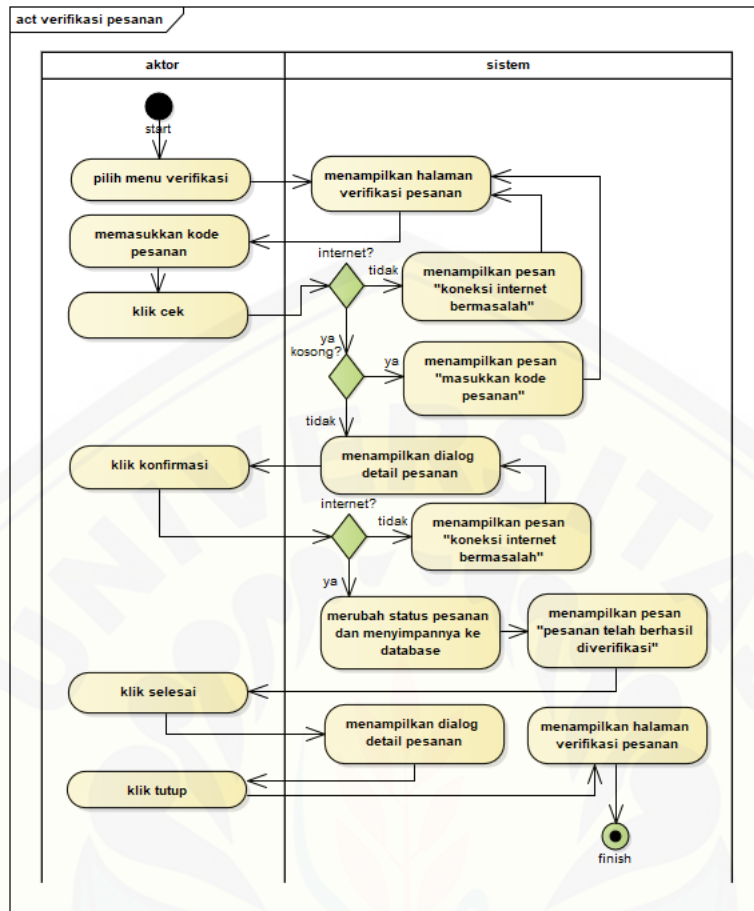
C.3. Melihat data pesanan



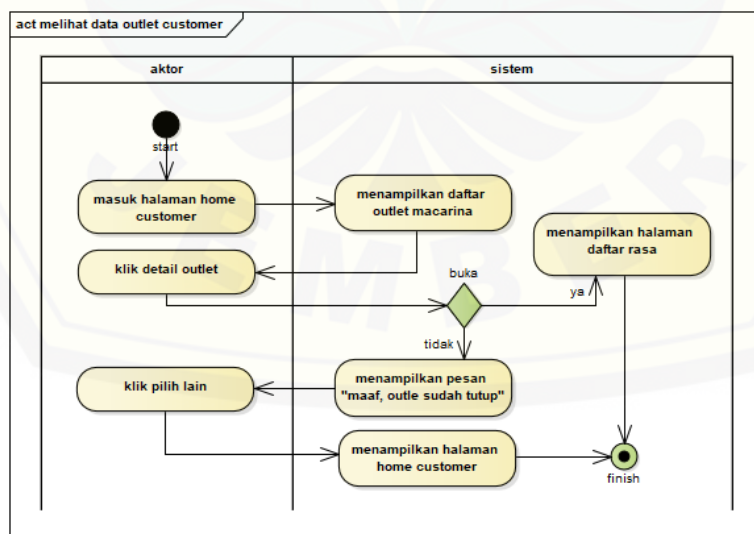
C.4. Mengelola data pesanan



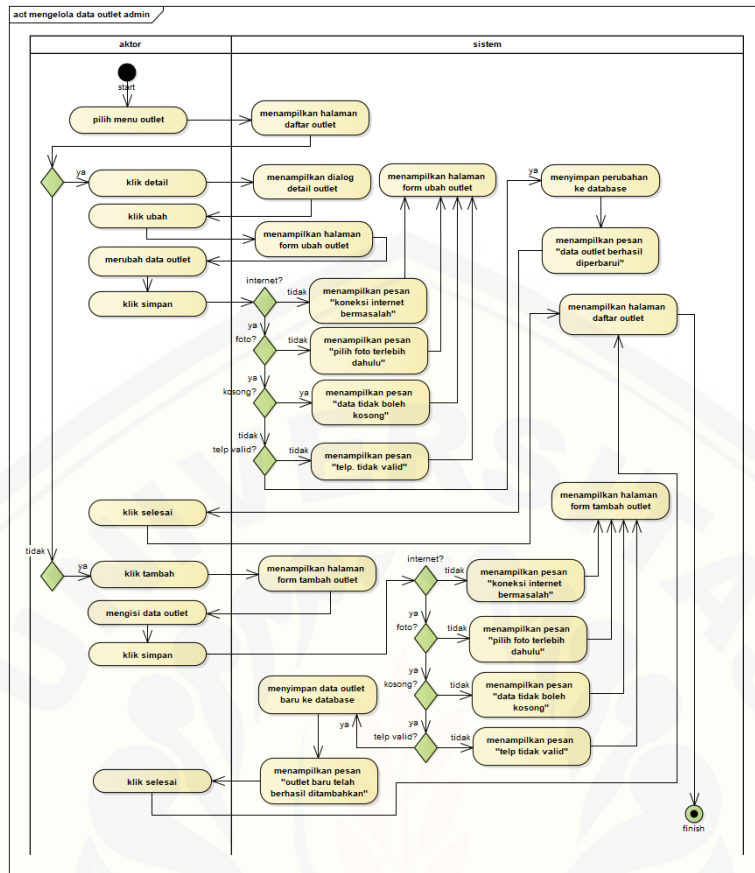
C.5. Verifikasi pesanan



C.6. Melihat data outlet

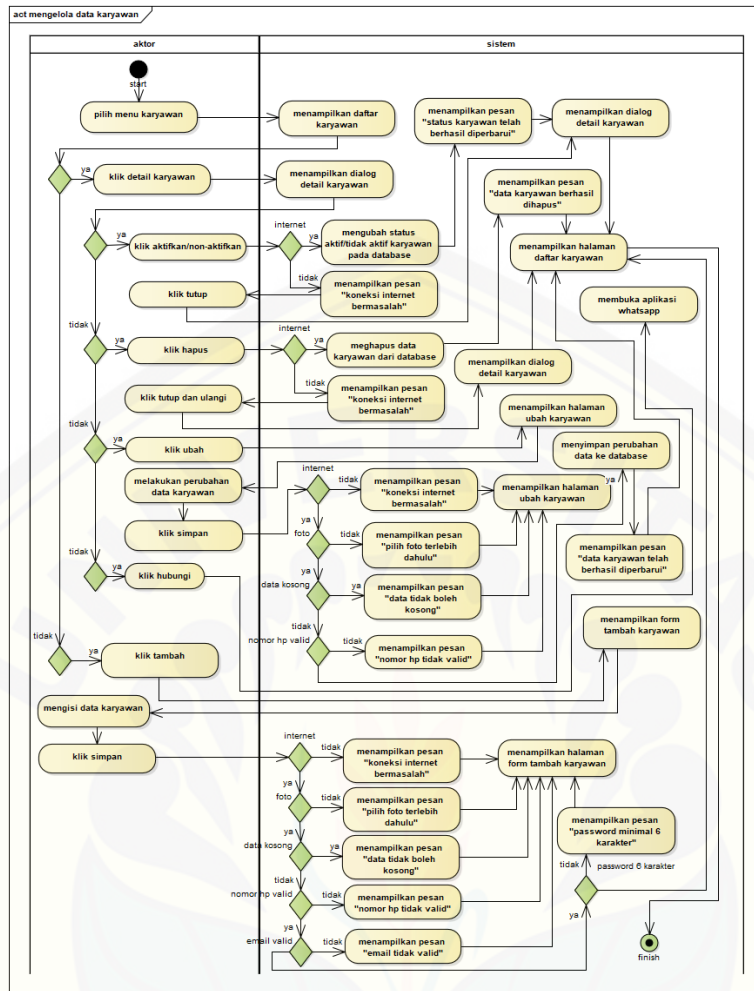


C.7. Mengelola data outlet

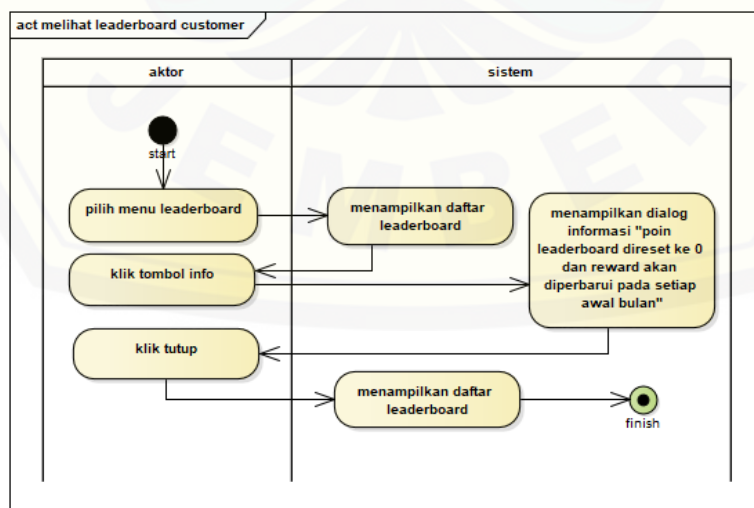


C.8. Mengelola data karyawan

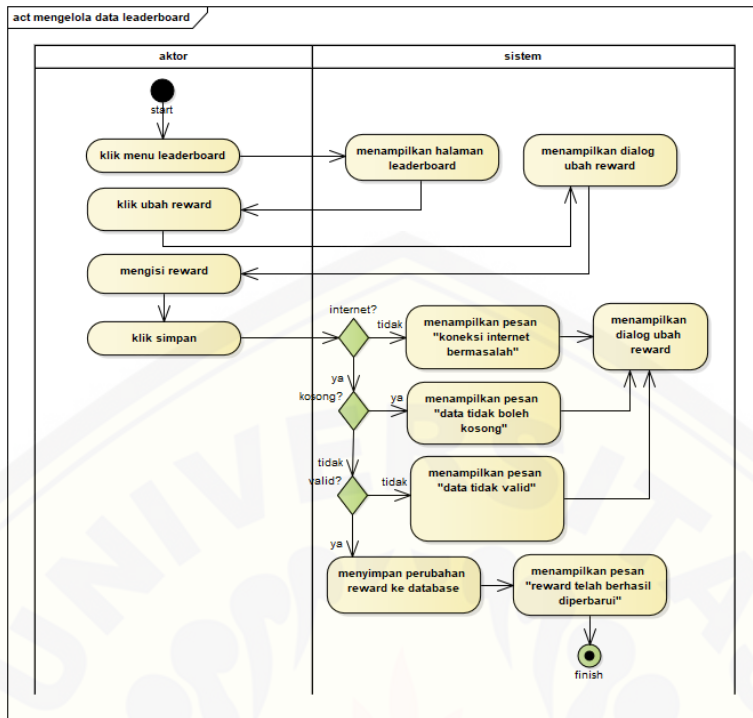




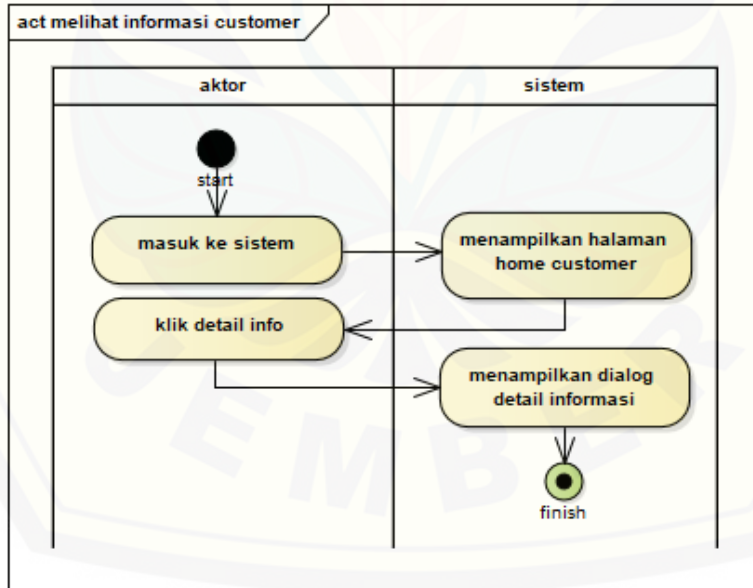
C.9. Melihat *leaderboard*



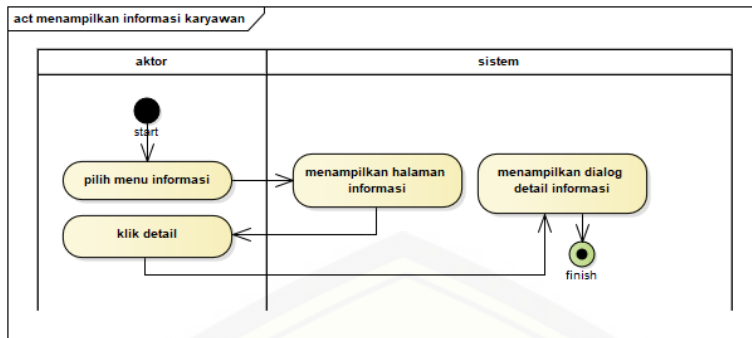
C.10. Mengelola data *leaderboard*



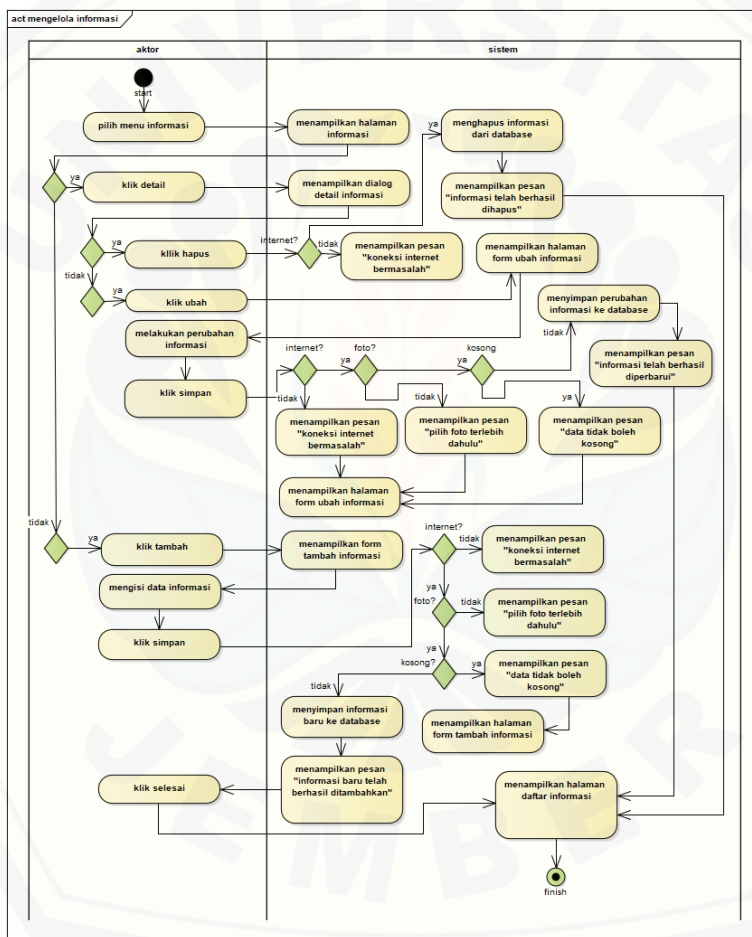
C.11. Melihat informasi *customer*



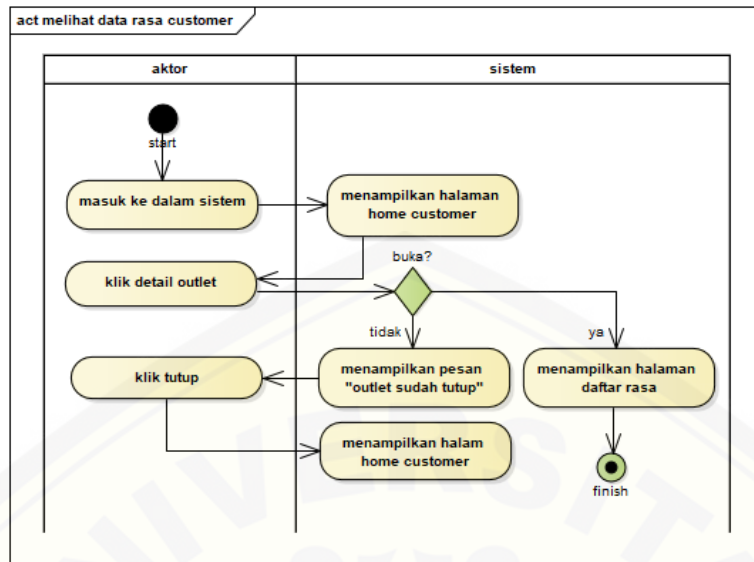
C.12. Melihat informasi karyawan



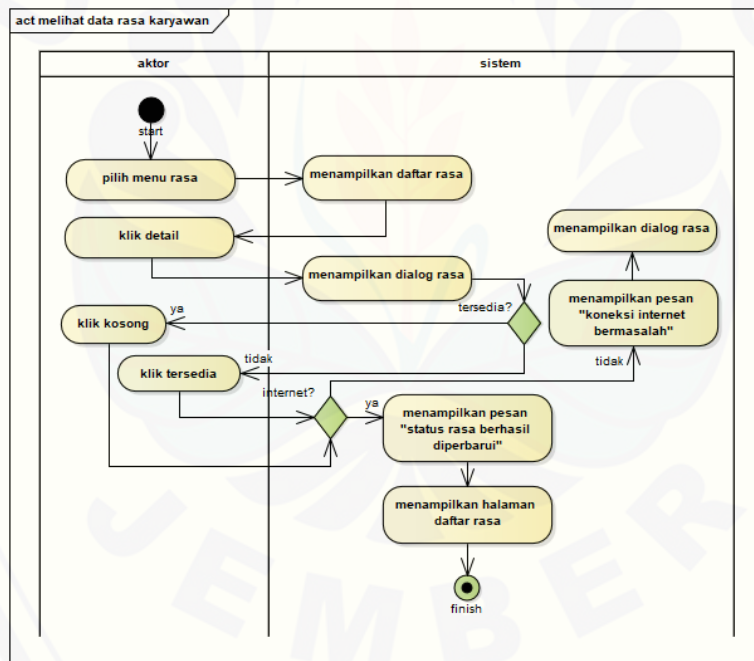
C.13. Mengelola informasi



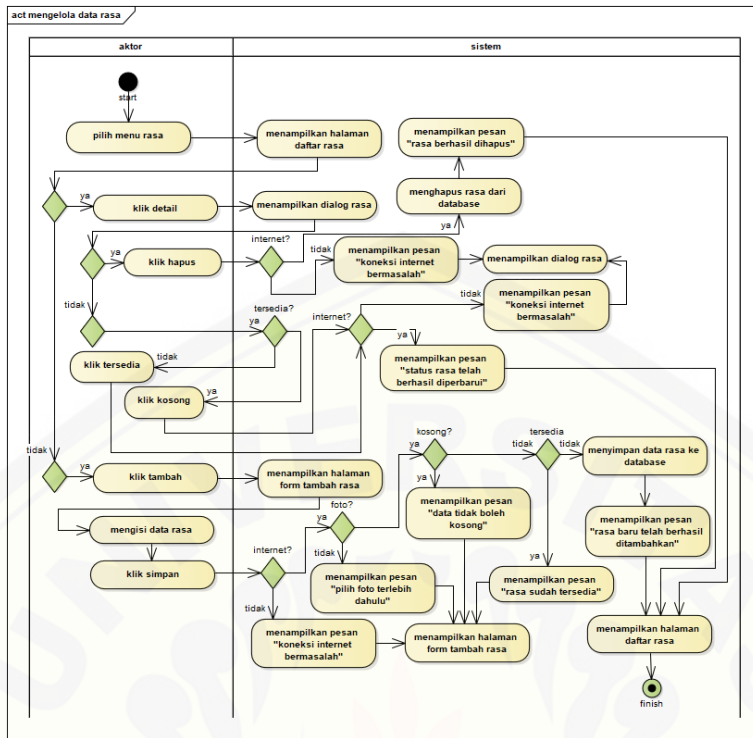
C.14. Melihat data rasa customer



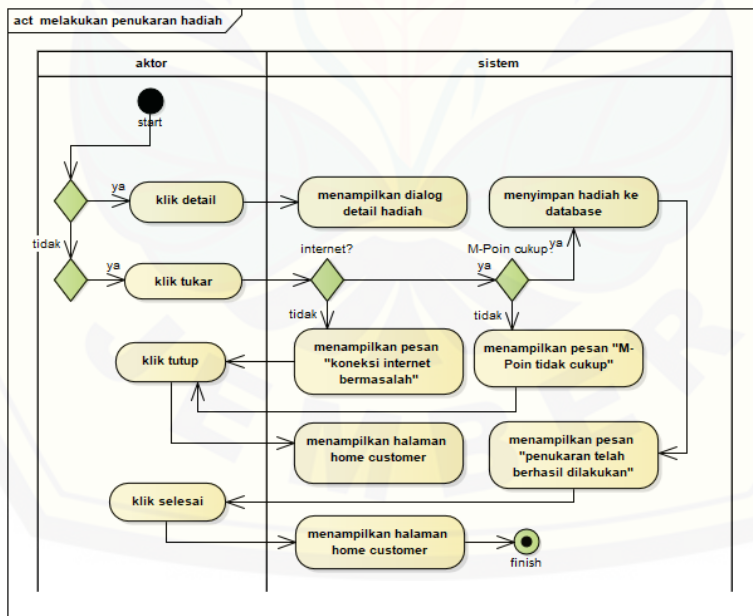
C.15. Melihat data rasa karyawan



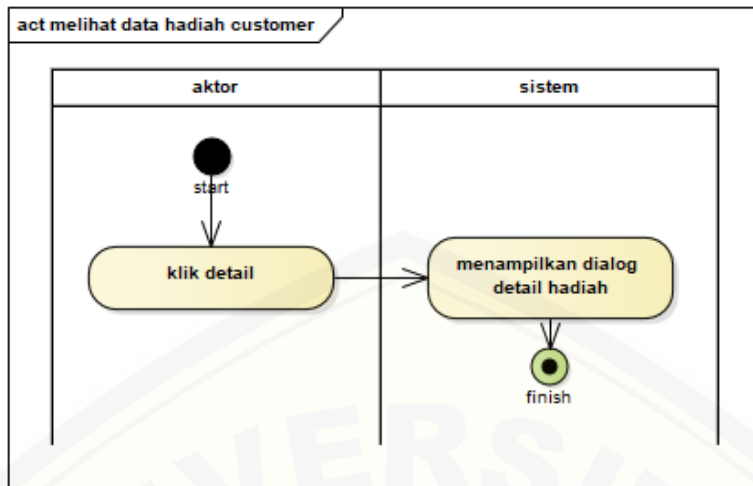
C.16. Mengelola data rasa



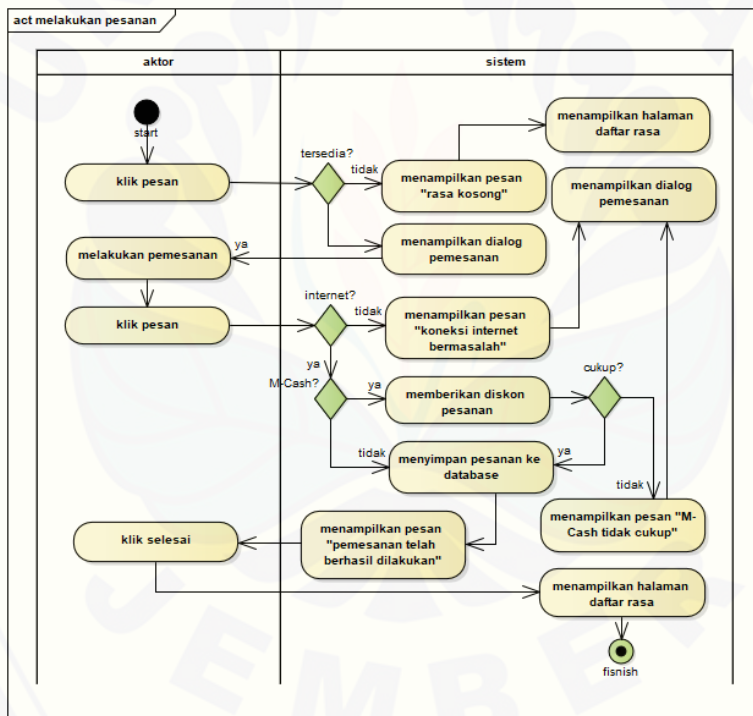
C.17. Melakukan penukaran hadiah



C.18. Melihat data hadiah

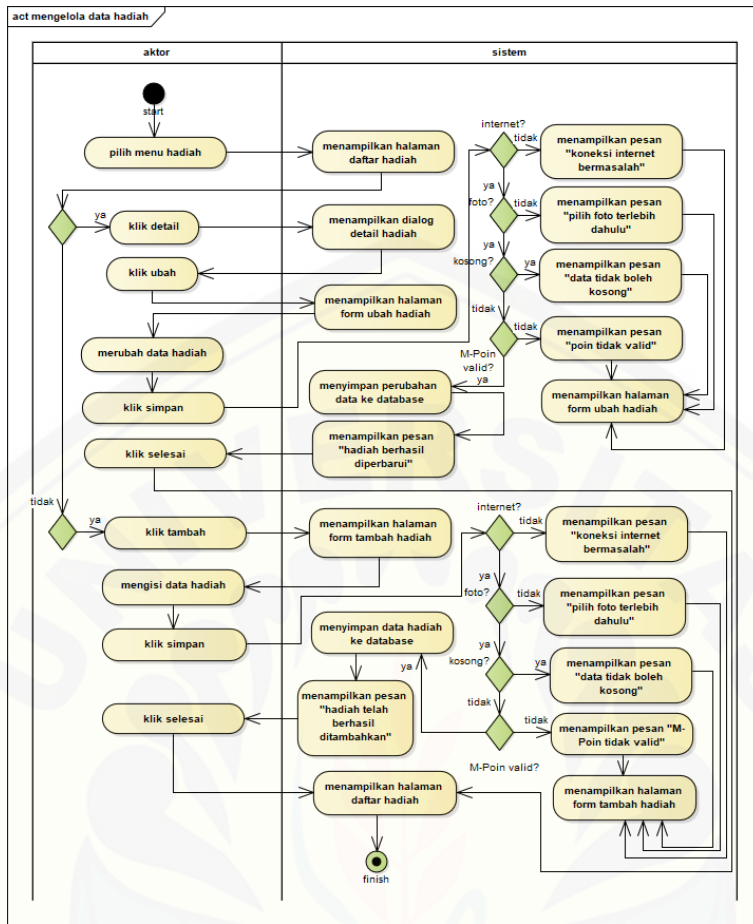


C.19. Melakukan pemesanan

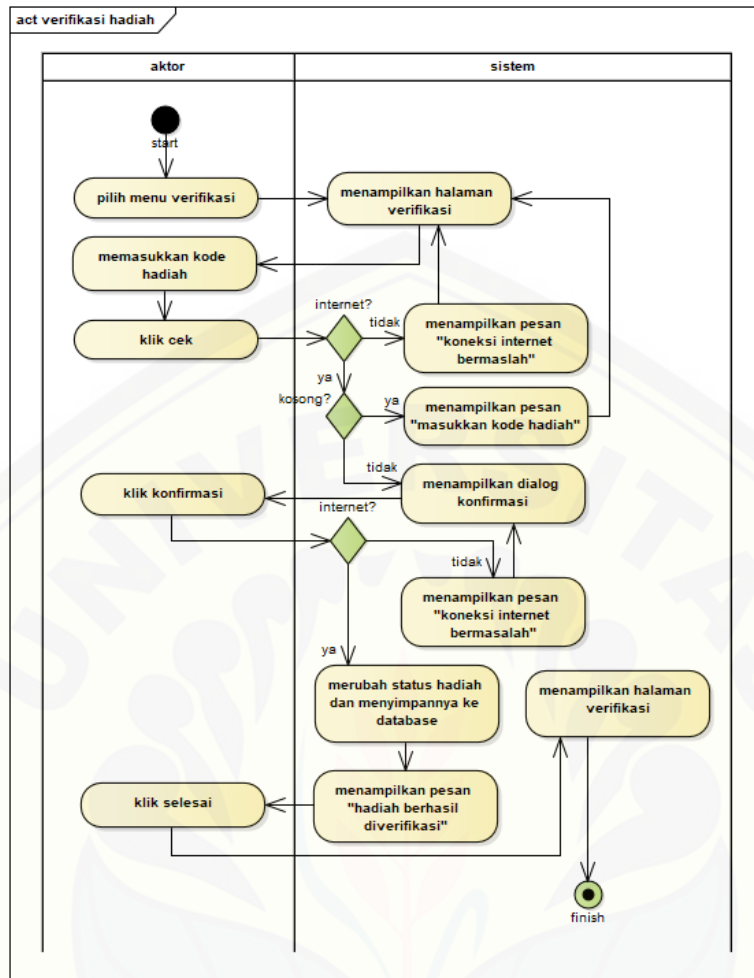


C.20. Mengelola data hadiah

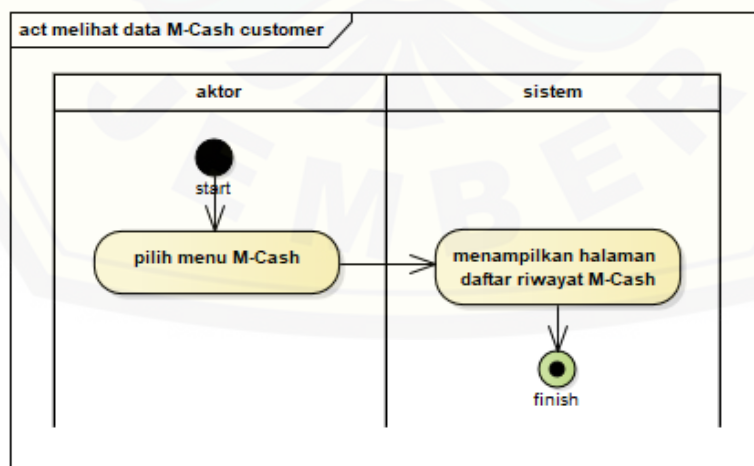




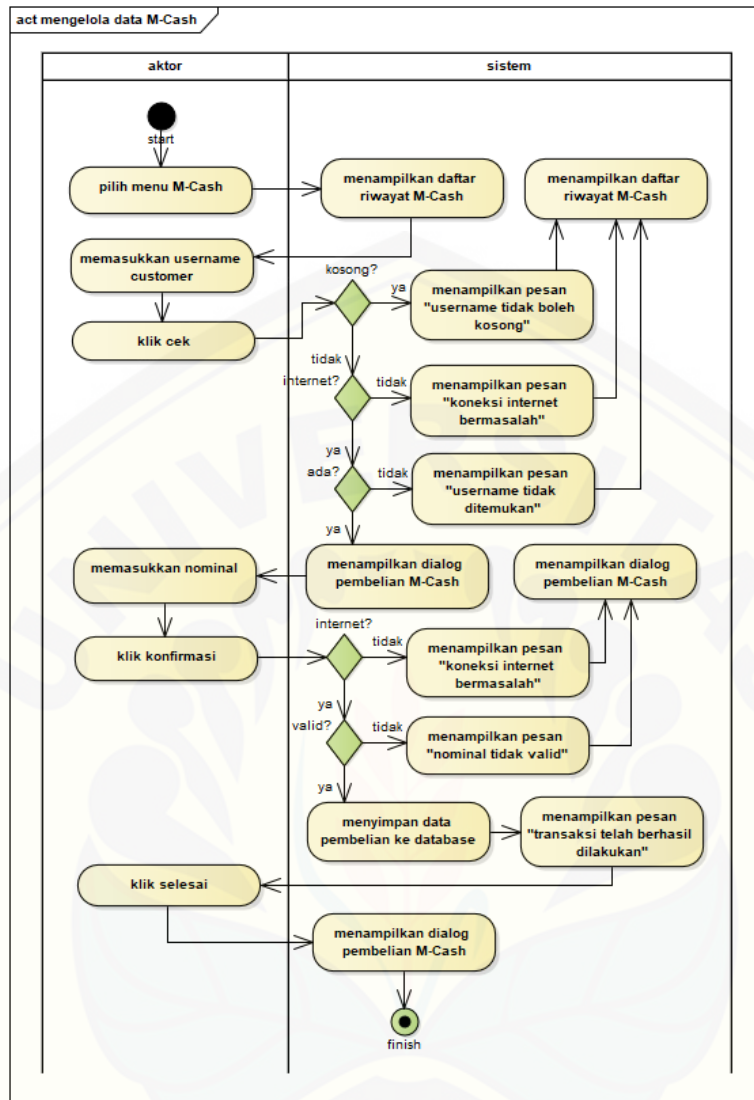
C.21. Verifikasi hadiah



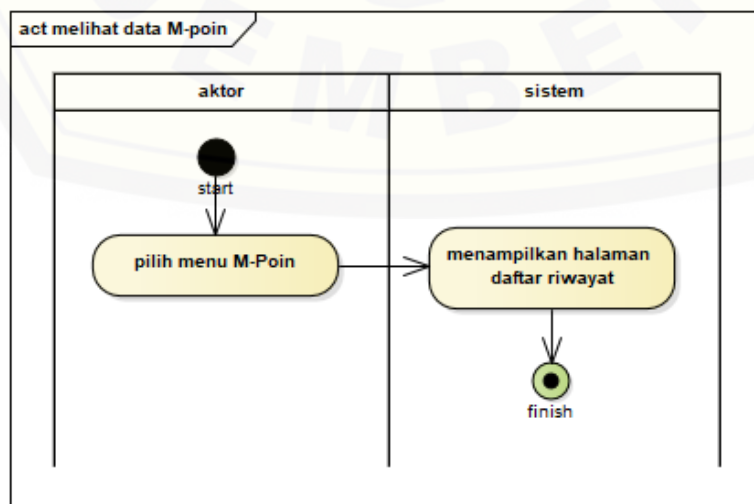
C.22. Melihat data *M-Cash*



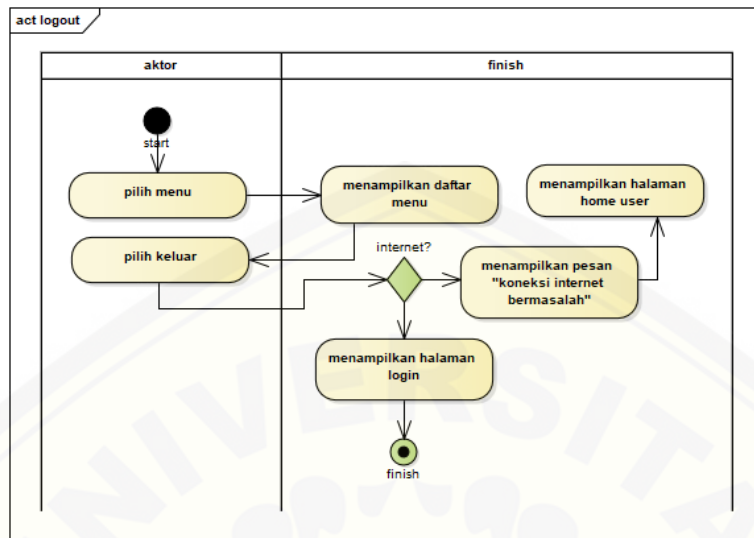
C.23. Mengelola data *M-Cash*



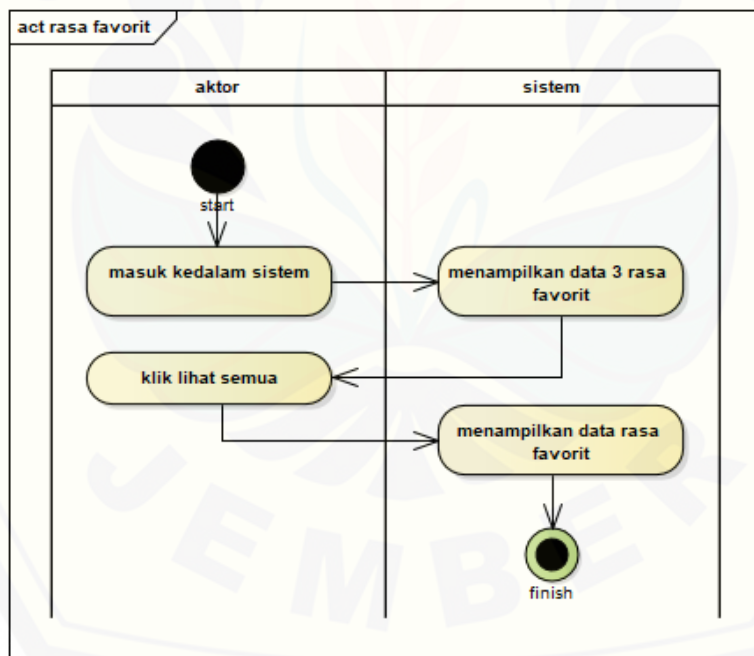
C.24. Melihat data *M-Poin*



C.25. Logout



C.26. Rasa favorit



D. Pengujian *Black Box*D.1. *Login*

No. <i>Use case</i>	UC01		
Nama <i>use case</i>	<i>Login</i>		
Aktor	<i>Customer, admin, karyawan</i>		
PreKondisi	<i>User telah memiliki email dan password dulu agar bisa login</i>		
PostKondisi	<i>User berhasil login sesuai hak akses</i>		
Aliran Normal <i>login</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Masukkan <i>email</i> dan <i>password</i>		√	
2. Klik tombol masuk			
	3. Sistem mengecek <i>email</i> dan <i>password</i>	√	
	4. Menampilkan halaman <i>home</i> sesuai dengan hak akses	√	
Aliran Alternatif Tidak ada koneksi internet			
2. Klik tombol masuk			
	3. Sistem mengecek koneksi internet	√	
	4. Menampilkan pesan “Koneksi internet Anda bermasalah, mohon untuk mengeceknya kembali”	√	
	5. Menampilkan halaman <i>login</i>	√	
Aliran Alternatif <i>email dan password</i> kosong			

2. Klik tombol masuk			
	3. Sistem mengecek <i>email</i> dan <i>password</i>	√	
	4. Menampilkan pesan “ <i>email</i> dan <i>password</i> tidak boleh kosong”	√	
	5. Menampilkan halaman <i>login</i>	√	
Aliran Alternatif <i>email</i> atau <i>password</i> kosong			
2. Klik tombol masuk			
	3. Sistem mengecek <i>email</i> atau <i>password</i>	√	
	4. Menampilkan pesan “ <i>email/password</i> tidak boleh kosong”	√	
	5. Menampilkan halaman <i>login</i>	√	
Aliran Alternatif <i>email</i> tidak valid			
2. Klik tombol masuk			
	3. Sistem mengecek validasi <i>email</i>	√	
	6. Menampilkan pesan “ <i>email</i> tidak valid”	√	
	7. Menampilkan halaman <i>login</i>	√	

## D.2. Registrasi

No. <i>Use case</i>	UC02
Nama <i>use case</i>	Registrasi akun



Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> belum mempunyai akun untuk bisa <i>login</i> ke dalam sistem		
PostKondisi	<i>Customer</i> berhasil membuat akun dan akan diarahkan ke halaman <i>home customer</i>		
Aliran Normal Proses <i>login</i> sistem			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Pilih foto dan masukkan Nama, <i>email</i> , <i>password</i> , nomor hp		√	
2. Klik tombol daftar			
	3. Sistem mengecek foto, nama, <i>email</i> , <i>password</i> dan nomor hp	√	
	4. Menampilkan halaman <i>home customer</i>	√	
Aliran Alternatif Tidak ada koneksi internet			
2. Klik tombol daftar			
	3. Sistem mengecek koneksi internet	√	
	4. Menampilkan pesan “koneksi internet bermasalah”	√	
	5. Menampilkan halaman daftar	√	
Aliran Alternatif Belum memilih foto			
2. Klik tombol daftar			

	3. Sistem mengecek url foto	√	
	4. Menampilkan pesan “Pilih foto terlebih dahulu”	√	
	5. Menampilkan halaman daftar	√	
Aliran Alternatif Nama, <i>email</i> , <i>password</i> atau nomor hp kosong			
2. Klik tombol daftar			
	3. Mengecek nama, <i>email</i> , <i>password</i> atau nomor hp	√	
	4. Menampilkan pesan “Data tidak boleh kosong”	√	
	5. Menampilkan halaman daftar	√	
Aliran Alternatif Panjang karakter <i>password</i>			
2. Klik tombol daftar			
	3. Mengecek panjang karakter <i>password</i>	√	
	4. Menampilkan pesan “ <i>password</i> min 6 karakter”	√	
	5. Menampilkan halaman daftar	√	
Aliran Alternatif Validasi nomor hp			
2. Klik tombol daftar			
	3. Sistem mengecek nomor hp	√	

	4. Menampilkan pesan “nomor hp tidak valid”	√	
	5. Menampilkan halaman daftar	√	
Aliran Alternatif Validasi <i>email</i>			
2. Klik tombol daftar			
	3. Sistem mengecek <i>email</i>	√	
	4. Menampilkan pesan “ <i>email</i> tidak valid”	√	
	5. Menampilkan halaman daftar	√	

D.3. Melihat data pesanan

No. <i>Use case</i>	UC03		
Nama <i>use case</i>	Melihat data pesanan		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> memilih menu pesanan		
PostKondisi	<i>Customer</i> berhasil melihat data pesanan		
Aliran Normal Melihat pesanan belum diproses			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	1. Menampilkan halaman <i>home customer</i>	√	
2. Klik menu pesanan			
3. Pilih tab pesanan belum diproses			
	4. Menampilkan daftar pesanan yang belum diproses	√	
5. Klik detail pesanan			

	6. Menampilkan dialog detail pesanan	√	
Aliran Normal Melihat pesanan diproses			
3. Pilih tab diproses pada <i>home customer</i>			
	4. Menampilkan daftar pesanan sedang diproses	√	
5. Pilih detail pesanan			
	6. Menampilkan dialog detail pesanan sedang diproses	√	
Aliran Normal Melihat pesanan selesai diproses			
3. Pilih tab selesai pada <i>home customer</i>			
	4. Menampilkan daftar pesanan selesai diproses	√	
5. Klik detail pesanan			
	6. Menampilkan dialog detail pesanan selesai diproses	√	
Aliran Normal Melihat pesanan ditolak			
3. Pilih ditolak			
	4. Melihat daftar pesanan yang tertolak	√	
5. Klik detail pesanan			
	6. Menampilkan dialog detail pesanan	√	
Aliran Normal Melihat riwayat pesanan			

7. Pilih riwayat			
	8. Menampilkan daftar riwayat pesanan	√	
9. Klik detail pesanan			
	10. Menampilkan dialog detail riwayat pesanan	√	

## D.4. Mengelola data pesanan

No. <i>Use case</i>	UC04		
Nama <i>use case</i>	Mengelola data pesanan		
Aktor	karyawan		
PreKondisi	Karyawan memilih menu pesanan		
PostKondisi	Karyawan berhasil melihat daftar pesanan pada <i>outletnya</i>		
Aliran Normal Melihat data pesanan belum diproses			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	1. Menampilkan halaman <i>home</i> karyawan	√	
2. Pilih menu pesanan			
3. Pilih pesanan			
	4. Menampilkan daftar pesanan belum diproses	√	
5. klik detail pesanan			
	6. Menampilkan dialog konfirmasi pesanan dengan pilihan “terima” atau “tolak”	√	
7. klik terima			

	8. Sistem mengubah status pesanan menjadi “sedang diproses”	√	
	9. Menampilkan halaman pesanan belum diproses	√	
Aliran Normal Melihat data pesanan diproses			
3. Pilih diproses			
	4. Menampilkan daftar pesanan sedang diproses	√	
5. klik detail pesanan			
	6. Menampilkan dialog konfirmasi pesanan dengan pilihan “sudah” atau “belum”	√	
7. Klik sudah			
	8. Sistem merubah status pesanan menjadi “sudah selesai”	√	
	9. Menampilkan halaman pesanan diproses	√	
Aliran Normal Meihat data pesanan selesai diproses			
3. Pilih selesai			
	4. Menampilkan daftar pesanan sudah selesai diproses	√	
5. klik detail pesanan			
	6. Menampilkan dialog detail pesanan	√	
Aliran Normal			



Meihat data pesanan ditolak			
3. Pilih menu			
4. Klik riwayat			
5. Pilih ditolak			
	6. Menampilkan daftar pesanan ditolak	√	
7. klik detail pesanan			
	8. Menampilkan dialog detail pesanan ditolak	√	
Aliran Normal			
Melihat data riwayat pesanan			
5. Pilih diambil			
	6. Menampilkan daftar pesanan sudah diambil	√	
7. klik detail pesanan			
	8. Menampilkan dialog detail pesanan selesai	√	
Aliran Normal			
Menolak pesanan			
10. Pilih tolak			
	11. Sistem mengubah status pesanan dari “dipesan” menjadi “ditolak”	√	
	12. Menutup dialog konfirmasi	√	
Aliran Normal			
Pesanan belum selesai diproses			
7. Pilih belum			
	8. Menutup dialog konfirmasi	√	
Aliran Alternatif			
Keterangan kosong saat melakuka penolakan pesanan			

7. Pilih tolak			
	8. Sistem mengecek keterangan	√	
	7. Menampilkan pesan “Keterangan tidak boleh kosong”	√	

D.5. Melihat data *outlet*

No. <i>Use case</i>	UC05		
Nama <i>use case</i>	Melihat data <i>outlet</i>		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> melihat <i>outlet</i> pada <i>home customer</i>		
PostKondisi	<i>Customer</i> mengetahui status <i>outlet</i> yang “buka” atau “tutup”		
Aliran Normal Melihat data <i>outlet</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	1. Menampilkan halaman <i>home customer</i>	√	
2. Melihat daftar <i>outlet</i>			
3. Klik tombol detail <i>outlet</i>			
	4. Menampilkan halaman rasa	√	
Aliran Alternatif Status <i>outlet</i> tutup			
3. Melihat daftar <i>outlet</i>			
4. Klik tombol detail <i>outlet</i>			
	5. Sistem mengecek status <i>outlet</i>	√	

	6. Menampilkan pesan “ <i>outlet</i> sudah tutup”	√	
	7. Menampilkan halaman <i>home customer</i>	√	

D.6. Mengelola data *outlet*

No. <i>Use case</i>	UC06		
Nama <i>use case</i>	Mengelola data <i>outlet</i>		
Aktor	<i>Admin</i>		
PreKondisi	<i>Admin</i> melihat daftar <i>outlet</i> di menu <i>outlet</i>		
PostKondisi	<i>Admin</i> berhasil melihat daftar <i>outlet</i>		
Aliran Normal Melihat data <i>outlet</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	1. Menampilkan daftar <i>outlet</i>	√	
2. Klik detail <i>outlet</i>			
	3. Menampilkan dialog detail <i>outlet</i>	√	
Aliran Normal Mengubah data <i>outlet</i>			
2. Klik detail <i>outlet</i>			
	3. Menampilkan dialog detail <i>outlet</i>	√	
4. Klik ubah			
	5. Menampilkan halaman form ubah <i>outlet</i>	√	
6. Klik simpan			

	7. Sistem akan merubah data <i>outlet</i> pada database	√	
	8. Menampilkan pesan “data <i>outlet</i> telah berhasil diperbarui”	√	
Aliran Normal Menambah <i>outlet</i> baru			
2. Klik tambah			
	3. Menampilkan halaman form tambah <i>outlet</i>	√	
4. Klik simpan			
	5. Sistem akan menambahkan data baru <i>outlet</i> ke database	√	
	6. Menampilkan pesan “data <i>outlet</i> baru telah berhasil ditambahkan”	√	
Aliran Alternatif Koneksi internet bermasalah			
4. Klik simpan			
	5. Sistem melakukan pengecekan koneksi internet	√	
	6. Menampilkan pesan “Koneksi internet bermasalah”	√	
	7. Menampilkan halaman tambah/ubah <i>outlet</i>	√	
Aliran Alternatif Belum milih foto <i>outlet</i>			

4. Klik simpan			
	5. Sistem melakukan pengecekan foto	√	
	6. Menampilkan pesan “pilih foto terlebih dahulu”	√	
	7. Menampilkan halaman form tambah/ubah <i>outlet</i>	√	
Aliran Alternatif Nomor telepon atau alamat kosong			
4. Klik simpan			
	5. Sistem melakukan pengecekan terhadap nomor telepon atau alamat	√	
	6. Menampilkan pesan “data tidak boleh kosong”	√	
	7. Menampilkan halaman form tambah/ubah <i>outlet</i>	√	
Aliran Alternatif Validasi nomor telepon			
4. Klik simpan			
	5. Sistem melakukan validasi terhadap nomor telepon	√	
	8. Menampilkan pesan “nomor telepon tidak valid”	√	
	9. Menampilkan halaman form tambah/ubah <i>outlet</i>	√	

## D.7. Mengelola data karyawan

No. Use case	UC07		
Nama use case	Mengelola data karyawan		
Aktor	Admin		
PreKondisi	Admin memilih menu karyawan		
PostKondisi	Admin berhasil melihat, mengaktifkan atau menon-aktifkan, mengubah, menambah dan menghapus data karyawan		
Aliran Normal Melihat data karyawan			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik detail karyawan			
	2. Menampilkan dialog detail karyawan	√	
Aliran Normal Mengaktifkan karyawan			
1. Klik detail karyawan			
	2. Menampilkan dialog detail karyawan	√	
3. Klik aktifkan/non-aktifkan			
	4. Mengubah status karyawan yang sebelumnya “aktif” menjadi “tidak aktif” dan sebaliknya.	√	
	5. Menampilkan pesan “status karyawan telah berhasil diperbarui”	√	
	6. Menampilkan dialog detail karyawan	√	
Aliran Normal Mengubah data karyawan			

1. Klik detail karyawan			
	2. Menampilkan dialog detail karyawan	√	
3. Klik ubah			
	4. Menampilkan halaman form ubah karyawan	√	
5. Mengubah data-data karyawan			
6. Klik simpan			
	7. Sistem melakukan penyimpanan perubahan data karyawan ke dalam database	√	
	8. Menampilkan pesan "data karyawan berhasil diperbarui"	√	
	9. Menampilkan halaman daftar karyawan	√	
Aliran Alternatif Koneksi internet bermasalah			
6. Klik simpan			
	7. Sistem mengecek koneksi internet	√	
	8. Menampilkan pesan "koneksi internet bermasalah"	√	
	9. Menampilkan halaman form ubah data karyawan	√	
Aliran Alternatif Nama, nomor hp atau alamat kosong			
6. Klik simpan			



	7. Sistem mengecek nama, nomor hp dan alamat	√	
	8. Menampilkan pesan “data tidak boleh kosong”	√	
	9. Menampilkan halaman form ubah karyawan	√	
Aliran Alternatif Nomor hp tidak valid			
6. Klik simpan			
	7. Sistem mengecek nomor hp	√	
	8. Menampilkan pesan “nomor hp tidak valid”	√	
	9. Menampilkan halaman form ubah karyawan	√	
Aliran Normal Menghapus data karyawan			
2. Klik detail karyawan			
	3. Menampilkan dialog detail karyawan	√	
4. Klik hapus			
	5. Sistem menghapus data karyawan dari database	√	
	6. Menampilkan pesan “data karyawan berhasil dihapus”	√	
Aliran Alternatif Koneksi internet bermasalah			
4. Klik hapus			
	5. Sistem mengecek koneksi internet	√	

	6. Menampilkan pesan “koneksi internet bermasalah”	√	
	7. Menampilkan dialog detail karyawan	√	
Aliran Normal Tambah data karyawan			
5. Klik simpan			
	6. Menampilkan halaman form tambah karyawan	√	
7. Memasukkan data-data karyawan			
8. Klik simpan			
	9. Sistem menyimpan data karyawan baru ke database	√	
	10. Menampilkan pesan “data karyawan baru telah berhasil ditambahkan”	√	
	11. Menampilkan halaman daftar karyawan	√	
Aliran Alternatif Nama, nomor hp atau alamat kosong			
5. Klik simpan			
	6. Sistem mengecek nama, nomor dan alamat	√	
	7. Menampilkan pesan “data tidak boleh kosong”	√	
	8. Menampilkan halaman form tambah karyawan	√	
Aliran Alternatif Koneksi internet bermasalah			

5. Klik simpan			
	6. Sistem mengecek koneksi internet	√	
	7. Menampilkan pesan “koneksi internet bermasalah”	√	
	8. Menampilkan halaman form tambah karyawan	√	
Aliran Alternatif Validasi nomor hp			
5. Klik simpan			
	6. Sistem mengecek nomor hp	√	
	7. Menampilkan pesan “nomor hp tidak valid”	√	
	8. Menampilkan halaman form tambah karyawan	√	

D.8. Melihat *leaderboard*

No. <i>Use case</i>	UC08		
Nama <i>use case</i>	Melihat <i>leaderboard</i>		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> memilih menu <i>leaderboard</i>		
PostKondisi	<i>Customer</i> berhasil melihat daftar <i>leaderboard</i> dan melihat <i>reward</i> yang didapat		
Aliran Normal Melihat daftar <i>leaderboard</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	4. Menampilkan <i>reward</i> dan daftar <i>leaderboard</i>	√	

D.9. Mengelola *leaderboard*

No. <i>Use case</i>	UC09		
Nama <i>use case</i>	Mengelola <i>leaderboard</i>		
Aktor	<i>Admin</i>		
PreKondisi	<i>Admin</i> memilih menu <i>leaderboard</i>		
PostKondisi	<i>Admin</i> berhasil melihat daftar <i>leaderboard</i> , melihat <i>reward</i> dan merubah <i>reward</i>		
Aliran Normal Melihat daftar <i>leaderboard</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	2. Menampilkan <i>reward</i> dan daftar <i>leaderboard</i>	√	
Aliran Normal Mengubah <i>reward</i>			
2. klik ubah <i>reward</i>			
	3. menampilkan dialog form ubah <i>reward</i>	√	
4. memasukkan jumlah <i>reward</i>			
5. klik simpan			
	6. sistem menyimpan perubahan <i>reward</i> ke dalam database	√	
	7. menampilkan pesan “ <i>reward</i> berhasil diperbarui”	√	
	8. menampilkan daftar <i>leaderboard</i>	√	
Aliran Alternatif Koneksi internet bermasalah			
5. klik simpan			

	6. sistem mengecek koneksi internet	√	
	7. menampilkan pesan “koneksi internet bermasalah”	√	
	8. menampilkan dialog form <i>reward</i>	√	
Aliran Alternatif <i>Reward</i> tidak valid			
5. klik simpan			
	6. sistem mengecek <i>reward</i>	√	
	7. menampilkan pesan “ <i>reward</i> tidak valid”	√	
	8. menampilkan dialog form <i>reward</i>	√	

D.10. Melihat informasi *customer*

No. <i>Use case</i>	UC10		
Nama <i>use case</i>	Melihat informasi		
Aktor	<i>Customer</i>		
PreKondisi	Masuk halaman <i>home customer</i>		
PostKondisi	<i>Customer</i> melihat daftar informasi		
Aliran Normal Melihat informasi			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik tombol detail			
	2. Menampilkan dialog detail informasi	√	

## D.11. Melihat informasi karyawan

No. <i>Use case</i>	UC11		
Nama <i>use case</i>	Melihat informasi		
Aktor	Karyawan		
PreKondisi	Karyawan memilih menu informasi		
PostKondisi	Karyawan berhasil melihat daftar informasi		
Aliran Normal Melihat informasi			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik tombol detail			
	2. Menampilkan dialog detail informasi	√	

## D.12. Mengelola informasi

No. <i>Use case</i>	UC12		
Nama <i>use case</i>	Mengelola informasi		
Aktor	<i>Admin</i>		
PreKondisi	<i>Admin</i> memilih menu informasi		
PostKondisi	<i>Admin</i> berhasil melihat, mengubah, menghapus dan menambah informasi		
Aliran Normal Melihat informasi			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik tombol detail			
	2. Menampilkan dialog detail informasi	√	
Aliran Normal Mengubah informasi			
1. klik tombol detail			
	2. Menampilkan dialog detail informasi	√	
3. klik ubah			

	4. menampilkan halaman form ubah informasi	√	
5. klik simpan			
	6. sistem menyimpan perubahan informasi ke dalam database	√	
	7. menampilkan pesan “informasi telah berhasil diperbarui”	√	
	8. menampilkan halaman daftar informasi	√	
Aliran Alternatif Koneksi internet bermasalah			
5. klik simpan			
	6. sistem mengecek koneksi internet	√	
	7. menampilkan pesan “koneksi internet bermasalah”	√	
	8. menampilkan halaman form ubah informasi	√	
Aliran Alternatif Judul atau deskripsi kosong			
5. klik simpan			
	6. sistem melakukan pengecekan terhadap judul dan deskripsi	√	
	7. menampilkan pesan “data tidak boleh kosong”	√	
	8. menampilkan halaman form ubah informasi	√	
Aliran Alternatif Foto kosong			



5. klik simpan			
	6. sistem mengecek foto	√	
	7. menampilkan pesan “pilih foto terlebih dahulu”	√	
	8. menampilkan halaman form ubah informasi	√	
Aliran Normal Menghapus informasi			
1. klik tombol detail			
	2. Menampilkan dialog detail informasi yang berisi: <ul style="list-style-type: none"> <li>- Judul</li> <li>- Tanggal dibuat</li> <li>- Tanggal mulai</li> <li>- Tanggal berakhir</li> <li>- deskripsi</li> </ul>	√	
3. klik hapus			
	4. sistem menghapus data informasi dari database	√	
	5. menampilkan pesan “informasi berhasil dihapus”	√	
	6. menampilkan halaman daftar informasi	√	
Aliran Alternatif Koneksi internet bermasalah			
3. klik hapus			
	4. sistem mengecek koneksi internet	√	
	5. menampilkan pesan “koneksi internet bermasalah”	√	

	6. menampilkan dialog detail informasi	√	
--	--	---	--

D.13. Melihat data rasa *customer*

No. <i>Use case</i>	UC13		
Nama <i>use case</i>	Melihat data rasa		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> memilih <i>outlet</i> yang buka		
PostKondisi	<i>Customer</i> berhasil melihat daftar rasa pada <i>outlet</i> tersebut		
Aliran Normal Melihat daftar rasa			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik tombol detail <i>outlet</i>			
	2. Menampilkan daftar rasa	√	
Aliran Alternatif <i>Outlet</i> tutup			
1. Klik tombol detail <i>outlet</i>			
	2. Sistem mengecek status <i>outlet</i>	√	
	3. Menampilkan pesan “ <i>outlet</i> sudah tutup, silahkan pilih <i>outlet</i> yang lainnya”	√	
	4. Menampilkan halaman - <i>home customer</i>	√	

## D.14. Melihat data rasa karyawan

No. <i>Use case</i>	UC14		
Nama <i>use case</i>	Melihat data rasa		
Aktor	Karyawan		
PreKondisi	karyawan memilih menu rasa		
PostKondisi	Karyawan berhasil melihat daftar rasa dan mengubah status persedia		
Aliran Normal Melihat daftar rasa			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
3. Klik tombol detail			
	4. Menampilkan dialog ubah status	√	
5. Klik tombol tersedia/kosong			
	6. Sistem menyimpan perubahan status persediaan rasa ke database	√	
	7. Menampilkan pesan “status persedian telah berhasil diperbarui”	√	
	8. Menampilkan halaman daftar rasa	√	
Aliran Alternatif Koneksi internet bermasalah			
5. Klik tombol tersedia/kosong			
	6. Sistem mengecek status <i>outlet</i>	√	
	7. Menampilkan pesan “ <i>outlet</i> sudah tutup, silahkan pilih <i>outlet</i> yang lainnya”	√	

	8. Menampilkan halaman <i>home customer</i>	√	
--	---	---	--

## D.15. Mengelola data rasa

No. Use case	UC15		
Nama use case	Mengelola data rasa		
Aktor	Admin		
PreKondisi	Admin memilih menu rasa		
PostKondisi	Admin berhasil melihat, mengubah status persediaan, menghapus dan menambah rasa baru		
Aliran Normal			
Melihat daftar rasa			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	1. Menampilkan daftar rasa	√	
Aliran Normal			
Mengubah status persediaan			
	1. Menampilkan daftar rasa	√	
2. Klik tombol detail			
	3. Menampilkan dialog detail rasa	√	
4. Klik tersedia/kosong			
	5. Sistem menyimpan perubahan status persedia ke database	√	
	6. Menampilkan pesan “status persediaan berhasil diperbarui”	√	
	7. Menampilkan halaman daftar rasa	√	
Aliran Alternatif			

Koneksi internet bermasalah			
4. Klik tersedia/kosong			
	5. Sistem mengecek koneksi internet	√	
	6. Menampilkan pesan “koneksi internet bermasalah”	√	
	7. Menampilkan dialog detail rasa	√	
Aliran Normal Meghapus rasa			
4. Klik hapus			
	5. Sistem menghapus rasa pada database	√	
	6. Menampilkan pesan “data rasa berhasil dihapus”	√	
	7. Menampilkan halaman daftar rasa	√	
Aliran Alternatif Koneksi internet bermasalah			
4. Klik hapus			
	5. Sistem mengecek koneksi internet	√	
	6. Menampilkan pesan “koneksi internet bermasalah”	√	
	7. Menampilkan dialog detail rasa	√	
Aliran Normal Menambah rasa			
2. Klik tombol tambah			

	3. Menampilkan halaman form tambah rasa	√	
4. Klik simpan			
	5. Sistem menyimpan rasa ke database	√	
	6. Menampilkan pesan “rasa baru telah berhasil ditambahkan”	√	
	7. Menampilkan daftar rasa	√	
Aliran Alternatif Koneksi internet bermasalah			
4. Klik simpan			
	5. Sistem mengecek koneksi internet	√	
	6. Menampilkan pesan “koneksi internet bermasalah”	√	
	7. Menampilkan halaman form tambah rasa	√	
Aliran Alternatif Belum pilih foto			
4. Klik simpan			
	5. Sistem mengecek foto	√	
	6. Menampilkan pesan “pilih foto terlebih dahulu”	√	
	7. Menampilkan halaman form tambah rasa	√	
Aliran Alternatif Rasa kosong			

4. Klik simpan			
	5. Sistem mengecek rasa	√	
	6. Menampilkan pesan “data tidak boleh kosong”	√	
	7. Menampilkan halaman form tambah rasa	√	
Aliran Alternatif Rasa sudah tersedia			
4. Klik simpan			
	5. Sistem mengecek rasa ke database	√	
	6. Menampilkan pesan “rasa sudah tersedia”	√	
	7. Menampilkan halaman form tambah rasa	√	

## D.16. Melakukan penukaran hadiah

No. <i>Use case</i>	UC16		
Nama <i>use case</i>	Melakukan penukaran data hadiah		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> berada pada halaman <i>home customer</i>		
PostKondisi	<i>Customer</i> berhasil melakukan penukaran hadiah menggunakan <i>M-Poin</i>		
Aliran Normal Menukar hadiah menggunakan <i>M-Poin</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik tombol tukar			
	2. Sistem menyimpan hadiah ke database	√	



	3. Menampilkan pesan “penukaran hadiah telah berhasil dilakukan”	√	
	4. Menampilkan halaman <i>home customer</i>	√	
Aliran Alternatif <i>M-Poin</i> tidak cukup			
1. Klik tombol tukar			
	2. Sistem mengecek <i>M-Poin customer</i>	√	
	3. Menampilkan pesan “ <i>M-Poin</i> tidak cukup”	√	
	4. Menampilkan halaman <i>home customer</i>	√	

## D.17. Melihat data hadiah

No. <i>Use case</i>	UC17		
Nama <i>use case</i>	Melihat data hadiah		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> berada pada halaman <i>home customer</i>		
PostKondisi	<i>Customer</i> berhasil melihat daftar hadiah		
Aliran Normal Melihat daftar hadiah			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik tombol detail			
	2. Menampilkan dialog detail hadiah	√	

## D.18. Melakukan pemesanan

No. <i>Use case</i>	UC18
---------------------	------

Nama <i>use case</i>	Melakukan pemesanan		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> melihat persediaan rasa pada <i>outlet</i>		
PostKondisi	<i>Customer</i> berhasil melakukan pemesanan		
Aliran Normal Pemesanan produk			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik tombol pesan			
	2. Menampilkan dialog pemesanan	√	
3. Menentukan pembayaran, jumlah, kemasan dan level			
4. Klik pesan			
	5. Sistem menyimpan data pesanan ke database	√	
	6. Menampilkan pesan “pemesanan telah berhasil dilakukan”	√	
	7. Menampilkan halaman daftar rasa	√	
Aliran Alternatif Status rasa kosong			
4. Klik tombol pesan			
	5. Sistem mengecek status persediaan	√	
	6. Menampilkan pesan “rasa kosong”	√	
	7. Menampilkan halaman daftar rasa	√	
Aliran Alternatif <i>M-Cash</i> tidak cukup			
3. Memilih metode pembayaran <i>M-Cash</i>			

4. Klik pesan			
	5. Sistem mengecek <i>M-Cash customer</i>	√	
	6. Menampilkan pesan “ <i>M-Cash</i> tidak cukup”	√	
	7. Menampilkan dialog pemesanan	√	
Aliran Alternatif Koneksi internet bermasalah			
4. Klik pesan			
	5. Sistem mengecek koneksi internet	√	
	6. Menampilkan pesan “koneksi internet bermasalah”	√	
	7. Menampilkan dialog pemesanan	√	

## D.19. Mengelola data hadiah

No. <i>Use case</i>	UC19		
Nama <i>use case</i>	Mengelola data hadiah		
Aktor	<i>Admin</i>		
PreKondisi	<i>Admin</i> memilih menu hadiah		
PostKondisi	<i>Admin</i> berhasil melihat, mengubah, menghapus dan menambah data hadiah		
Aliran Normal Melihat data hadiah			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik tombol detail			
	2. Menampilkan dialog detail hadiah	√	
Aliran Normal hapus data hadiah			

1. Klik tombol detail			
	2. Menampilkan dialog detail hadiah	√	
3. Klik hapus			
	4. Sistem menghapus data hadiah dari database	√	
	5. Menampilkan pesan “data hadiah berhasil dihapus”	√	
	6. Menampilkan halaman daftar hadiah	√	
Aliran Alternatif Koneksi internet bermasalah			
7. Klik hapus			
	8. Sistem mengecek koneksi internet	√	
	9. Menampilkan pesan “koneksi internet bermasalah”	√	
	10. Menampilkan halaman daftar hadiah	√	
Aliran Normal Ubah hadiah			
8. Klik tombol detail			
	10. Menampilkan dialog detail hadiah yang berisi: <ul style="list-style-type: none"> <li>- Tanggal buat</li> <li>- Nama</li> <li>- Harga (<i>M-Poin</i>)</li> <li>- Jumlah</li> <li>- Merek</li> </ul>	√	

	- Deskripsi		
11. Klik ubah			
	12. Menampilkan halaman form ubah hadiah yang berisi: <ul style="list-style-type: none"> <li>- Foto</li> <li>- Nama (teks)</li> <li>- Deskripsi (teks)</li> <li>- Harga (angka)</li> <li>- Merek (teks)</li> <li>- Jumlah (angka)</li> </ul>	√	
13. Melakukan perubahan pada data hadiah			
14. Klik simpan			
	15. Sistem menyimpan perubahan data hadiah ke database	√	
	16. Menampilkan pesan “data hadiah berhasil diperbarui”	√	
	17. Menampilkan halaman daftar hadiah	√	
Aliran Alternatif Koneksi internet bermasalah			
10. Klik simpan			
	11. Sistem mengecek koneksi internet	√	
	12. Menampilkan pesan “koneksi internet bermasalah”	√	
	13. Menampilkan halaman form ubah hadiah	√	
Aliran Alternatif			

Belum memilih foto			
10. Klik simpan			
	11. Sistem mengecek foto	√	
	12. Menampilkan pesan “pilih foto terlebih dahulu”	√	
	13. Menampilkan halaman form ubah hadiah	√	
Aliran Alternatif Nama, harga, jumlah, deskripsi dan merek kosong			
10. Klik simpan			
	11. Sistem mengecek nama, harga, jumlah, deskripsi dan merek	√	
	12. Menampilkan pesan “data tidak boleh kosong”	√	
	13. Menampilkan halaman form ubah hadiah	√	
Aliran Alternatif Harga tidak valid			
10. Klik simpan			
	11. Sistem mengecek harga	√	
	12. Menampilkan pesan “harga tidak valid”	√	
	13. Menampilkan halaman form ubah hadiah	√	
Aliran Alternatif Jumlah tidak valid			
10. Klik simpan			

	11. Sistem mengecek jumlah	√	
	12. Menampilkan pesan “jumlah tidak valid”	√	
	13. Menampilkan halaman form ubah hadiah	√	

D.20. Melihat data *M-Cash*

No. <i>Use case</i>	UC20		
Nama <i>use case</i>	Melihat data <i>M-Cash</i>		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> memilih menu <i>M-Cash</i>		
PostKondisi	<i>Customer</i> berhasil melihat riwayat transaksi <i>M-Cash</i>		
Aliran Normal Melihat riwayat <i>M-Cash</i>			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	1. Menampilkan daftar riwayat transaksi <i>M-Cash</i>	√	

D.21. Mengelola data *M-Cash*

No. <i>Use case</i>	UC21		
Nama <i>use case</i>	Mengelola data <i>M-Cash</i>		
Aktor	<i>Admin</i> , karyawan		
PreKondisi	<i>User</i> memilih menu <i>M-Cash</i>		
PostKondisi	<i>User</i> berhasil melihat riwayat transaksi dan menambahkan <i>M-Cash</i>		
Aliran Normal Melihat riwayat transaksi <i>M-Cash</i>			



Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
	1. Menampilkan daftar riwayat transaksi <i>M-Cash</i>	√	
Aliran Normal Menambah <i>M-Cash</i>			
	1. Menampilkan daftar riwayat transaksi <i>M-Cash</i>	√	
2. Memasukkan <i>username customer</i>			
3. Klik tombol cek			
	4. Menampilkan dialog form <i>M-Cash</i>	√	
5. Memasukkan nominal transaksi			
6. Klik konfirmasi			
	7. Sistem menambahkan <i>M-Cash</i> ke <i>customer</i>	√	
	8. Menampilkan pesan “transaksi <i>M-Cash</i> telah berhasil dilakukan”	√	
	9. Menampilkan dialog form <i>M-Cash</i>	√	
10. Klik selesai			

	11. Menampilkan halaman daftar riwayat transaksi <i>M-Cash</i>	√	
Aliran Alternatif Nomial tidak valid			
6. Klik konfirmasi			
	7. Sistem mengecek nominal	√	
	8. Menampilkan pesan “nominal tidak valid”	√	
	9. Menampilkan dialog <i>M-Cash</i>	√	
Aliran Alternatif Nominal kosong			
6. Klik konfirmasi			
	7. Sistem mengecek nominal	√	
	8. Menampilkan pesan “data tidak boleh kosong”	√	
	9. Menampilkan dialog <i>M-Cash</i>	√	
Aliran Alternatif Koneksi internet bermasalah			
6. Klik konfirmasi			

	7. Sistem mengecek koneksi internet	√	
	8. Menampilkan pesan “koneksi internet bermasalah”	√	

D.22. Melihat data *M-Poin*

No. <i>Use case</i>	UC22		
Nama <i>use case</i>	Melihat data <i>M-Poin</i>		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> memilih menu <i>M-Poin</i>		
PostKondisi	<i>Customer</i> berhasil melihat riwayat <i>M-Poin</i> yang didapatkan		
Aliran Normal Melihat riwayat <i>M-Poin</i>			
Aksi Aktor	Reaksi Sistem		
	1. Menampilkan daftar riwayat <i>M-Poin</i> diperoleh	√	

D.23. *Logout*

No. <i>Use case</i>	UC23		
Nama <i>use case</i>	<i>Logout</i>		
Aktor	<i>Customer</i> , karyawan, <i>admin</i>		
PreKondisi	<i>User</i> telah <i>login</i> terlebih dahulu kedalam sistem dan berada halaman <i>home</i> masing-masing <i>user</i>		

PostKondisi	<i>User</i> keluar dari sistem dan akan diteruskan ke halaman <i>login</i>		
Aliran Normal			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Pilih menu			
2. Pilih keluar			
	3. Menampilkan pesan “ <i>logout</i> berhasil”	√	
	4. Menampilkan halaman <i>login</i>	√	
Aliran Alternatif Koneksi internet bermasalah			
2. Pilih keluar			
	3. Sistem mengecek koneksi internet	√	
	4. Menampilkan pesan “Koneksi internet bermasalah”	√	
	5. Menampilkan halaman <i>home</i>	√	

## D.24. Verifikasi pesanan

No. <i>Use case</i>	UC24
Nama <i>use case</i>	Verifikasi pesanan
Aktor	Karyawan

PreKondisi	Karyawan pilih menu verifikasi		
PostKondisi	Karyawan berhasil verifikasi pesanan		
Aliran Normal Verifikasi pesanan			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Masukkan kode pesanan			
2. Klik cek			
	3. Menampilkan dialog detail pesanan	√	
4. Klik konfirmasi			
	5. Merubah status pesanan dan menyimpannya ke database	√	
	6. Menampilkan pesan “pesanan berhasil diverifikasi”	√	
Aliran Alternatif Kode kosong			
2. Klik cek			
	3. mengecek kode pesanan	√	
	4. menampilkan pesan “masukkan kode pesanan”	√	
Aliran Alternatif Internet bermasalah			
4. klik konfirmasi			
	5. mengecek koneksi internet	√	

	6. menampilkan pesan “koneksi internet bermasalah”	√	
Aliran Alternatif Salah <i>outlet</i>			
4. klik konfirmasi			
	5. mengecek <i>outlet</i>	√	
	6. menampilkan pesan “salat <i>outlet</i> ”	√	
Aliran Alternatif Masih diproses			
4. klik konfirmasi			
	5. mengecek pesanan	√	
	6. menampilkan pesan “masih diproses”	√	
Aliran Alternatif Sudah diambil			
4. klik konfirmasi			
	5. mengecek pesanan	√	
	6. menampilkan pesan “pesanan sudah diambil”	√	

## D.25. Verifikasi hadiah

No. <i>Use case</i>	UC24
Nama <i>use case</i>	Verifikasi hadiah
Aktor	<i>Admin</i>
PreKondisi	<i>Admin</i> pilih menu verifikasi

PostKondisi	Admin berhasil verifikasi hadiah		
Aliran Normal Verifikasi hadiah			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Masukkan kode hadiah			
2. Klik cek			
	3. Menampilkan dialog detail hadiah	√	
4. Klik konfirmasi			
	5. Merubah status pesanan dan menyimpannya ke database	√	
	6. Menampilkan pesan “pesanan berhasil diverifikasi”	√	
Aliran Alternatif Kode kosong			
2. Klik cek			
	3. mengecek kode pesanan	√	
	4. menampilkan pesan “masukkan kode hadiah”	√	
Aliran Alternatif Internet bermasalah			
4. klik konfirmasi			
	5. mengecek koneksi internet	√	
	6. menampilkan pesan “koneksi	√	



	internet bermasalah”		
Aliran Alternatif Sudah ditukar			
4. klik konfirmasi			
	5. mengecek pesanan	√	
	6. menampilkan pesan “pesanan sudah diambil”	√	

## D.26. Rasa favorit

No. <i>Use case</i>	UC26		
Nama <i>use case</i>	Melihat rasa favorit		
Aktor	<i>Customer</i>		
PreKondisi	<i>Customer</i> berada pada <i>home customer</i>		
PostKondisi	<i>Customer</i> berhasil melihat daftar rasa favorit		
Aliran Normal Melihat daftar rasa favorit			
Aksi Aktor	Reaksi Sistem	Berhasil	Gagal
1. Klik lihat semua			
	2. Menampilkan halaman daftar rasa favorit	√	

## E. Kode Program

### E.1. Melakukan Pemesanan

```
package
com.universitaskjember.basirudin.aplikasiMacarina.view.customer.home.tok
o;

import android.app.Activity;
import android.app.Dialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AlertDialog;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Spinner;
import android.widget.TextView;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.universitaskjember.basirudin.aplikasiMacarina.R;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelCustomer;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelMacarina;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelPengguna;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelPesanan;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelToko;
import com.universitaskjember.basirudin.aplikasiMacarina.util.Util;

public class Dialog_pemesanan extends Dialog implements
View.OnClickListener {
    private Activity activity;
    private String strRasa = "", strToko = "", strAlamat = "", strFoto
= "";

    public Dialog_pemesanan(Activity activity) {
        super(activity);
        this.activity = activity;
    }

    @Override
    public void onClick(View v) {

    }

    public void getKode(String rasa, String toko, String foto) {
        strRasa = rasa;
        strToko = toko;
        strFoto = foto;
    }
}
```

```

    }

    private TextView txtRasa, txtToko, txtAlamat, txtHarga,
    txtHargaAkhir, txtPoin, btnPesan, btnBatal;
    private EditText txtJumlah, txtDiskon;
    private ImageButton btnKurang, btnTambah, btnInfo;
    private Spinner spinKemasan, spinLevel, spinBayar;

    private DatabaseReference reference;

    private Util util = new Util();

    private ArrayAdapter<String> adapterKemasan, adapterLevel,
    adapterBayar;

    String[] arrKemasan = {Util.util_get_kemasan_ecopack,
    Util.util_get_kemasan_pouch, Util.util_get_kemasan_extra,
    Util.util_get_kemasan_box};
    String[] arrLevel = {Util.util_get_level_1, Util.util_get_level_2,
    Util.util_get_level_3, Util.util_get_level_4, Util.util_get_level_5};
    String[] arrBayar = {Util.util_get_pembayaran_dilokasi,
    Util.util_get_pembayaran_mcash};

    private String strHarga = "", strTotalHarga = "", strHargaAkhir =
    "", strPoin = "", strStatus = "", strJumlah = "", strDiskon = "",
    strBayar = "", strKemasan = "", strLevel = "";
    private String strUsername = "", strNomor = "", strUrutan = "",
    strNama = "", strCashku = "", strNomorToko = "", strSisaCash = "";

    private int intJumlah;
    private double dJumlah, dHarga, dHargaAkhir, dDiskon, dTotalHarga,
    dCashku, dPoin, dSisaCash;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.customer_dialog_pemesanan);

        txtRasa = findViewById(R.id.customer_dialog_pesanan_rasa);
        txtToko = findViewById(R.id.customer_dialog_pesanan_toko);
        txtAlamat = findViewById(R.id.customer_dialog_pesanan_alamat);
        txtHarga =
    findViewById(R.id.customer_dialog_pesanan_total_harga);
        txtHargaAkhir =
    findViewById(R.id.customer_dialog_pesanan_harga_setelah_diskon);
        txtPoin =
    findViewById(R.id.customer_dialog_pesanan_poin_didapat);
        txtJumlah = findViewById(R.id.customer_dialog_pesanan_jumlah);
        txtDiskon = findViewById(R.id.customer_dialog_pesanan_diskon);

        btnPesan =
    findViewById(R.id.customer_dialog_pesanan_btn_pesan);
        btnBatal =
    findViewById(R.id.customer_dialog_pesanan_btn_batal);
        btnKurang =
    findViewById(R.id.customer_dialog_pesanan_btn_berkurang);
        btnTambah =
    findViewById(R.id.customer_dialog_pesanan_btn_bertambah);
        btnInfo = findViewById(R.id.customer_dialog_pesanan_btn_info);

        spinBayar =
    findViewById(R.id.customer_dialog_pesanan_pilih_pembayaran);
        spinKemasan =

```

```
findViewById(R.id.customer_dialog_pesanan_pilih_kemasan);
    spinLevel =
findViewById(R.id.customer_dialog_pesanan_pilih_level);

    txtRasa.setText(strRasa);
    txtToko.setText(strToko);
    txtJumlah.setText("1");

    getUsername();
    getAlamat();
    getDiskon();

    adapterKemasan = new ArrayAdapter<String>(getContext(),
android.R.layout.simple_spinner_dropdown_item, arrKemasan);
    spinKemasan.setAdapter(adapterKemasan);
    spinKemasan.setSelection(0);
    spinKemasan.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
            strKemasan = (String)
spinKemasan.getItemAtPosition(position);
            switch (strKemasan) {
                case Util.util_get_kemasan_ecopack:
                    strHarga = "6000";
                    harga();
                    break;
                case Util.util_get_kemasan_pouch:
                    strHarga = "10000";
                    harga();
                    break;
                case Util.util_get_kemasan_extra:
                    strHarga = "15000";
                    harga();
                    break;
                case Util.util_get_kemasan_box:
                    strHarga = "20000";
                    harga();
                    break;
                default:
            }
        }
    });
    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
});

    adapterLevel = new ArrayAdapter<String>(getContext(),
android.R.layout.simple_spinner_dropdown_item, arrLevel);
    spinLevel.setAdapter(adapterLevel);
    spinLevel.setSelection(0);
    spinLevel.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
            strLevel = (String)
spinLevel.getItemAtPosition(position);
        }
    });
    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
}
```

```

    });

    adapterBayar = new ArrayAdapter<String>(getContext(),
    android.R.layout.simple_spinner_dropdown_item, arrBayar);
    spinBayar.setAdapter(adapterBayar);
    spinBayar.setSelection(0);
    spinBayar.setOnItemClickListener(new
    AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View
    view, int position, long id) {
            strBayar = (String)
    spinBayar.getItemAtPosition(position);
            harga();
        }
        @Override
        public void onNothingSelected(AdapterView<?> parent) {
        }
    });

    btnKurang.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            pilihKurang();
        }
    });

    btnTambah.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            pilihTambah();
        }
    });

    btnPesan.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            cekInternet();
        }
    });

    btnBatal.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            dismiss();
        }
    });

    btnInfo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            notif("Diskon", "Diskon hanya berlaku untuk pemesanan
    yang menggunakan M-Cash", "tutup", null);
        }
    });
}

private void cekInternet() {
    if (!Util.get_internet(getContext())) {
        notif("Gagal", Util.util_get_pesan_no_internet, null,
    "batal");
    } else {
        pesan();
    }
}
}

```



```

private void getDiskon() {
    reference =
    FirebaseDatabase.getInstance().getReference().child(Util.util_data_Maca
    rina).child("toko");
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
    dataSnapshot) {
            if (dataSnapshot.exists()) {
                ModelMacarina m =
    dataSnapshot.getValue(ModelMacarina.class);
                txtDiskon.setText(m.getDiskon());
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError
    databaseError) {
        }
    });
}

private void pesan() {
    dCashku = Double.valueOf(strCashku);
    dSisaCash = dCashku - dHargaAkhir;
    strSisaCash = Util.get_double_number_to_string(dSisaCash);
    if (strBayar.equalsIgnoreCase(Util.util_get_pembayaran_mcash))
    {
        if (dCashku < dHargaAkhir) {
            notif("Gagal", "M-Cash Anda tidak mencukupi untuk
    melakukan pemesanan produk", "tutup", null);
        } else {
            simpan();
            updateCash();
        }
    } else {
        simpan();
    }
}

private void updateCash() {
    reference =
    FirebaseDatabase.getInstance().getReference().child(Util.util_data_cust
    omer).child(strUsername);
    reference.child("cash").setValue(strSisaCash);
}

private void simpan() {
    if (strBayar.equalsIgnoreCase(Util.util_get_pembayaran_mcash))
    {
        strStatus = Util.util_get_status_lunas;
    } else {
        strStatus = Util.util_get_status_belum_lunas;
    }

    int intUrut = Integer.parseInt(strUrutan) + 1;
    String strKode = strNomor + String.valueOf(intUrut);

    FirebaseDatabase.getInstance().getReference().child(Util.util_data_cust
    omer)
        .child(strUsername).child("noUrut")
        .setValue(String.valueOf(intUrut));
    ModelPesanan m = new ModelPesanan(strFoto, strKode, strToko,
    strUsername, strNama, strNomor, strNomorToko, strRasa, strAlamat,

```

```

strKemasan,
        strLevel, strJumlah, strBayar, strDiskon, strHarga,
strHargaAkhir, strPoin, strStatus, Util.util_get_kondisi_belum_ditukar,
Util.util_get_tanggal_sekarang(),
        "-", "-", "-", "-",
Util.util_get_status_pesanan_dipesan);
        reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_pesanan);
        reference.child(strKode).setValue(m).addOnCompleteListener(new
OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    btnBatal.setText("SELESAI");
                    notif("Berhasil", "Pemesanan telah berhasil
dilakukan", "selesai", null);
                }
            }
        });
    }

    private void getUsername() {
        reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_pengguna).child(Util.get_UID());
        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                if (dataSnapshot.exists()) {
                    ModelPegguna m =
dataSnapshot.getValue(ModelPegguna.class);
                    strUsername = m.getUsername();
                    getData(strUsername);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError
databaseError) {
                }
            }
        });
    }

    private void getData(String username) {
        reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_customer).child(username);
        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                if (dataSnapshot.exists()) {
                    ModelCustomer m =
dataSnapshot.getValue(ModelCustomer.class);
                    strUrutan = m.getNoUrut();
                    strNomor = m.getNomor();
                    strNama = m.getNama();
                    strCashku = m.getCash();
                }
            }
        });
    }

```



```

    }
    @Override
    public void onCancelled(@NonNull DatabaseError
databaseError) {
    }
    });
}

private void pilihKurang() {
    strJumlah = txtJumlah.getText().toString().trim();
    intJumlah = Integer.parseInt(strJumlah) - 1;

    txtJumlah.setText(String.valueOf(intJumlah));
    if (intJumlah < 2) {
btnKurang.setBackgroundResource(R.drawable.style_bulat_kecil_gray);
        btnKurang.setEnabled(false);
    }
    harga();
}

private void pilihTambah() {
    strJumlah = txtJumlah.getText().toString().trim();
    intJumlah = Integer.parseInt(strJumlah) + 1;

    txtJumlah.setText(String.valueOf(intJumlah));

btnKurang.setBackgroundResource(R.drawable.style_bulat_kecil_merah);
    btnKurang.setEnabled(true);

    harga();
}

private void harga() {
    if (strBayar.equalsIgnoreCase(Util.util_get_pembayaran_mcash))
{
        strDiskon = txtDiskon.getText().toString().trim();
    } else {
        strDiskon = "0";
    }

    strJumlah = txtJumlah.getText().toString().trim();
    dJumlah = Double.valueOf(strJumlah);
    dDiskon = Double.valueOf(strDiskon);
    dHarga = Double.valueOf(strHarga);

    dTotalHarga = dJumlah * dHarga;
    dHargaAkhir = dTotalHarga - (dTotalHarga * dDiskon / 100);

    strTotalHarga = Util.get_double_number_to_string(dTotalHarga);
    strHargaAkhir = Util.get_double_number_to_string(dHargaAkhir);

    dPoin = dTotalHarga / 100;
    strPoin = Util.get_double_number_to_string(dPoin);

    txtPoin.setText(Util.get_rupiah(strPoin));
    txtHarga.setText(Util.get_rupiah(strTotalHarga));
    txtHargaAkhir.setText(Util.get_rupiah(strHargaAkhir));
}

private void getAlamat() {
    reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_toko

```

```

).child(strToko);
reference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        if (dataSnapshot.exists()) {
            ModelToko m =
dataSnapshot.getValue(ModelToko.class);
            strAlamat = m.getAlamat();
            strNomorToko = m.getNomor();
            txtAlamat.setText(strAlamat);
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError
databaseError) {
    }
});
}

private void notif(String judul, String pesan, final String
positif, String negatif) {
    AlertDialog.Builder builder = new
AlertDialog.Builder(getContext())
        .setCancelable(false)
        .setTitle(judul)
        .setMessage(pesan)
        .setPositiveButton(positif, new OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {
                if (positif.equalsIgnoreCase("coba lagi")) {
                    cekInternet();
                } else if (positif.equalsIgnoreCase("tutup")) {
                    dialog.cancel();
                } else {
                    dismiss();
                }
            }
        })
        .setNegativeButton(negatif, new OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {
                dialog.cancel();
            }
        });
    AlertDialog alertDialog = builder.create();
    alertDialog.show();
}
}

```

## E.2. Verifikasi Pesanan

```

package
com.universitaskember.basirudin.aplikasiMacarina.view.karyawan.verifika
si;

import android.app.Activity;
import android.app.Dialog;

```

```

import android.content.DialogInterface;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AlertDialog;
import android.view.View;
import android.widget.TextView;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.universitaskjember.basirudin.aplikasiMacarina.R;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelCustomer;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelKaryawan;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelLeaderboard
;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelPengguna;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelPesanan;
import com.universitaskjember.basirudin.aplikasiMacarina.util.Util;

public class Dialog_verifikasi extends Dialog implements
View.OnClickListener {
    private Activity activity;
    private String strKode = "";

    public Dialog_verifikasi(Activity activity) {
        super(activity);
        this.activity = activity;
    }

    public void getKode(String kode) {
        strKode = kode;
    }

    @Override
    public void onClick(View v) {

    }

    private TextView txtKode, txtPemesan, txtToko,
        txtKondisi, txtRasa, txtJumlah,
        txtKemasan, txtLevel, txtBayar,
        txtStatus, txtHargaSatuan, txtHargaTotal,
        txtHargaDiskon, txtDiskon,
        btnKonfirmasi, btnBatal, txtPoin, txtPesanan;

    private String strPemesan = "", strToko = "", strKondisi = "",
        strRasa = "", strJumlah = "", strKemasan = "",
        strLevel = "", strBayar = "", strStatus = "",
        strHargaSatuan = "", strHargaTotal = "", strHargaDiskon =
        "",
        strDiskon = "", strPoin = "", strPesanan = "";

    private String strUserKaryawan = "", strTokoKaryawan = "";

```

```

    private String strUsername = "", strPoinCustomer = "",
    strPoinLeaderboard = "", strDesc = "";

    private double dHarga, dDiskon, dHargaTotal, dPoin, dJumlah;
    private double dPoinCustomer, dPoinLeaderboard, dPoinToCustomer,
    dPoinToLeaderboard, dDesc, dDescLeaderboard;

    private String btn_internet = "tutup & ulangi",
    btn_selesai = "selesai",
    btn_toko = "tutup",
    btn_kondisi = "tutup",
    btn_pesanan = "tutup";

    private DatabaseReference reference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.karyawan_dialog_verifikasi);
        getUsername();

        txtKode = findViewById(R.id.karyawan_dialog_verifikasi_kode);
        txtPemesan =
        findViewById(R.id.karyawan_dialog_verifikasi_pemesan);
        txtToko = findViewById(R.id.karyawan_dialog_verifikasi_outlet);
        txtKondisi =
        findViewById(R.id.karyawan_dialog_verifikasi_kondisi);
        txtRasa = findViewById(R.id.karyawan_dialog_verifikasi_rasa);
        txtJumlah =
        findViewById(R.id.karyawan_dialog_verifikasi_jumlah);
        txtKemasan =
        findViewById(R.id.karyawan_dialog_verifikasi_kemasan);
        txtLevel = findViewById(R.id.karyawan_dialog_verifikasi_level);
        txtBayar =
        findViewById(R.id.karyawan_dialog_verifikasi_pembayaran);
        txtStatus =
        findViewById(R.id.karyawan_dialog_verifikasi_status);
        txtHargaSatuan =
        findViewById(R.id.karyawan_dialog_verifikasi_harga_kemasan);
        txtHargaTotal =
        findViewById(R.id.karyawan_dialog_verifikasi_harga_total);
        txtHargaDiskon =
        findViewById(R.id.karyawan_dialog_verifikasi_harga_diskon);
        txtDiskon =
        findViewById(R.id.karyawan_dialog_verifikasi_diskon);
        btnKonfirmasi =
        findViewById(R.id.karyawan_dialog_verifikasi_btn_konfirmasi);
        btnBatal =
        findViewById(R.id.karyawan_dialog_verifikasi_btn_batal);
        txtPoin = findViewById(R.id.karyawan_dialog_verifikasi_poin);
        txtPesanan =
        findViewById(R.id.karyawan_dialog_verifikasi_pesanan);

        getPesanan();

        btnKonfirmasi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                cekInternet();
            }
        });

        btnBatal.setOnClickListener(new View.OnClickListener() {

```

```
        @Override
        public void onClick(View v) {
            dismiss();
        }
    });
}

@Override
public void onBackPressed() {
    super.onBackPressed();
    dismiss();
}

private void cekInternet() {
    if (!Util.get_internet(getContext())) {
        pesan("Gagal",
            Util.util_get_pesan_no_internet,
            btn_internet);
    } else {
        konfirmasi();
    }
}

private void konfirmasi() {
    if (strToko.equalsIgnoreCase(strTokoKaryawan)) {
        if (strPesanan.equalsIgnoreCase(Util.util_get_status_pesanan_selesai)) {
            if (strKondisi.equalsIgnoreCase(Util.util_get_kondisi_belum_ditukar)) {
                simpan();
            } else {
                pesan("Gagal", "Mohon maaf, produk sudah diambil",
                    btn_kondisi);
            }
        } else {
            pesan("Gagal", "Mohon maaf, produk masih dalam
                proses.", btn_pesanan);
        }
    } else {
        pesan("Gagal", "Mohon maaf, produk Macarina yang dipesan
            bukan berasal dari " + strTokoKaryawan + ", melainkan dari " + strToko,
            btn_toko);
    }
}

private void simpan() {
    String kode = Util.get_tanggal_dan_waktu_sekarang()
        .replace(" ", "")
        .replace("/", "")
        .replace(":", "")
        .toUpperCase();
    String[] arrChildPoin = {"tanggal", "poin"},
        arrValuePoin = {Util.get_tanggal_dan_waktu_sekarang(),
            strPoin};
    String[] arrChild = {"status", "kondisi", "pesanan",
        "tanggalTukar", "bulanTukar", "tahunTukar"},
        arrValue = {Util.util_get_status_lunas,
            Util.util_get_kondisi_sudah_ditukar,
            Util.util_get_status_pesanan_diambil,
            Util.get_tanggal(),
            Util.get_bulan(),
            Util.get_tahun()};
    dPoin = Double.valueOf(strPoin);
}
```



```

        dPoinToCustomer = dPoin + dPoinCustomer;
        dPoinToLeaderboard = dPoin + dPoinLeaderboard;
        dDescLeaderboard = Double.valueOf(Util.util_get_desc) -
        dDescLeaderboard;

        String[] arrChildLeader = {"poin", "desc"},
        arrValueLeader =
        {Util.get_double_number_to_string(dPoinToLeaderboard),
        Util.get_double_number_to_string(dDescLeaderboard)};

        System.out.println("poinku: " + String.valueOf(dPoin));
        System.out.println("poinku to C: " +
        String.valueOf(dPoinToCustomer));
        System.out.println("poinku to L: " +
        String.valueOf(dPoinToLeaderboard));

        for (int i = 0; i < arrChild.length; i++) {
            FirebaseDatabase.getInstance().getReference()
                .child(Util.util_data_pesanan)
                .child(strKode).child(arrChild[i])
                .setValue(arrValue[i]);
        }

        for (int i = 0; i < arrChildLeader.length; i++) {
            FirebaseDatabase.getInstance().getReference()
                .child(Util.util_data_leaderboard)
                .child(strUsername)
                .child(arrChildLeader[i])
                .setValue(arrValueLeader[i]);
        }

        for (int i = 0; i < arrChildPoin.length; i++) {
            FirebaseDatabase.getInstance().getReference()
                .child(Util.util_data_poin)
                .child(strUsername)
                .child(kode)
                .child(arrChildPoin[i])
                .setValue(arrValuePoin[i]);
        }

        FirebaseDatabase.getInstance().getReference()
            .child(Util.util_data_customer)
            .child(strUsername).child("poin")
            .setValue(Util.get_double_number_to_string(dPoinToCustomer))
            .addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    if (task.isSuccessful()) {
                        pesan("Berhasil",
                            "Status pesanan dengan kode " +
                            strKode + " telah berhasil diperbarui.",
                            btn_selesai);
                    }
                }
            });
    }

    private void getPesanan() {
        reference =
        FirebaseDatabase.getInstance().getReference().child(Util.util_data_pesa
        nan).child(strKode);
    }

```

```

reference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        if (dataSnapshot.exists()) {
            ModelPesanan m =
dataSnapshot.getValue(ModelPesanan.class);
            strPemesan = m.getPembeli();
            strUsername = m.getUserPembeli();
            strToko = m.getToko();
            strKondisi = m.getKondisi();
            strRasa = m.getRasa();
            strJumlah = m.getJumlah();
            strKemasan = m.getKemasan();
            strLevel = m.getLevel();
            strBayar = m.getPembayaran();
            strStatus = m.getStatus();
            strHargaSatuan = m.getHarga();
            strHargaDiskon = m.getHargaDiskon();
            strDiskon = m.getDiskon();
            strPoin = m.getPoin();
            strPesanan = m.getPesanan();

            getPoinCustomer();

            dHarga = Double.valueOf(strHargaSatuan);
            dJumlah = Double.valueOf(strJumlah);
            dHargaTotal = dHarga * dJumlah;

            strHargaTotal =
Util.get_double_number_to_string(dHargaTotal);

            txtKode.setText(strKode);
            txtPemesan.setText(strPemesan);
            txtToko.setText(strToko);
            txtKondisi.setText(strKondisi);
            txtRasa.setText(strRasa);
            txtJumlah.setText(strJumlah + " Produk");
            txtKemasan.setText(strKemasan);
            txtLevel.setText(strLevel);
            txtBayar.setText(strBayar);
            txtStatus.setText(strStatus);
            txtHargaSatuan.setText("Rp. " +
Util.get_rupiah(strHargaSatuan));
            txtHargaTotal.setText("Rp. " +
Util.get_rupiah(strHargaTotal));
            txtHargaDiskon.setText("Rp. " +
Util.get_rupiah(strHargaDiskon));
            txtDiskon.setText(strDiskon + "%");
            txtPoin.setText(Util.get_rupiah(strPoin) + "
Poin");

            txtPesanan.setText(strPesanan);

            if
(strKondisi.equalsIgnoreCase(Util.util_get_kondisi_belum_ditukar)) {
                btnKonfirmasi.setTextColor(getContext().getResources().getColor(R.color
.colorPrimary));
            } else {
                btnKonfirmasi.setTextColor(getContext().getResources().getColor(R.color
.warnaGray));
            }
        }
    }
}

```



```

        if
        (strPesanan.equalsIgnoreCase(Util.util_get_status_pesanan_selesai)) {
        btnKonfirmasi.setTextColor(getContext().getResources().getColor(R.color
        .colorPrimary));
        } else {
        btnKonfirmasi.setTextColor(getContext().getResources().getColor(R.color
        .warnaGray));
        }
    }
}

@Override
public void onCancelled(@NonNull DatabaseError
databaseError) {
}
});
}

private void getPoinCustomer() {
    reference =
    FirebaseDatabase.getInstance().getReference().child(Util.util_data_cust
    omer).child(strUsername);
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            if (dataSnapshot.exists()) {
                ModelCustomer m =
dataSnapshot.getValue(ModelCustomer.class);
                strPoinCustomer = m.getPoin();
                dPoinCustomer = Double.valueOf(strPoinCustomer);

                getPoinLeaderboard();
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {
        }
    });
}

private void getPoinLeaderboard() {
    reference =
    FirebaseDatabase.getInstance().getReference().child(Util.util_data_lead
    erboard).child(strUsername);
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            if (dataSnapshot.exists()) {
                ModelLeaderboard m =
dataSnapshot.getValue(ModelLeaderboard.class);
                strPoinLeaderboard = m.getPoin();
                strDesc = m.getDesc();
                dPoinLeaderboard =
                Double.valueOf(strPoinLeaderboard);

```

```

        dDesc = Double.valueOf(strDesc);
    } else {
        strPoinLeaderboard = "0";
        dPoinCustomer = Double.valueOf(strPoinCustomer);
    }
}

@Override
public void onCancelled(@NonNull DatabaseError
databaseError) {

}

});
}

private void getUsername() {
    reference =
    FirebaseDatabase.getInstance().getReference().child(Util.util_data_peng
guna).child(Util.get_UID());
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            if (dataSnapshot.exists()) {
                ModelPegguna m =
                dataSnapshot.getValue(ModelPegguna.class);
                strUserKaryawan = m.getUsername();

                getToko();
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

        }

    });
}

private void getToko() {
    reference =
    FirebaseDatabase.getInstance().getReference().child(Util.util_data_kary
awan).child(strUserKaryawan);
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            if (dataSnapshot.exists()) {
                ModelKaryawan m =
                dataSnapshot.getValue(ModelKaryawan.class);
                strTokoKaryawan = m.getToko();

            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

        }

    });
}
}

```

```

private void pesan(String judul, String pesan, final String
positif) {
    AlertDialog.Builder builder = new
AlertDialog.Builder(getContext())
        .setCancelable(false)
        .setTitle(judul)
        .setMessage(pesan)
        .setPositiveButton(positif, new OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {
                if (positif.equalsIgnoreCase(btn_selesai)) {
                    dismiss();
                } else {
                    dialog.cancel();
                }
            }
        });
    AlertDialog alertDialog = builder.create();
    alertDialog.show();
}
}

```

### E.3. Melakukan Penukaran Hadiah

```

package
com.universitaskjember.basirudin.aplikasiMacarina.adapter.customer.home;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.universitaskjember.basirudin.aplikasiMacarina.R;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelCustomer;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelHadiah;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelPengguna;
import com.universitaskjember.basirudin.aplikasiMacarina.util.Util;
import

```

```

com.universitasmember.basirudin.aplikasiMacarina.view.customer.home.hadiah.DialogDetailHadiah;

import java.util.List;

import de.hdodenhof.circleimageview.CircleImageView;

public class AdapterHadiah extends
RecyclerView.Adapter<AdapterHadiah.MyViewHolder> {
    private DatabaseReference reference;

    String username = "";
    double poinku;
    String strSisa = "";

    private Context context;
    private Activity activity;
    private List<ModelHadiah> model;

    public AdapterHadiah(Context context, Activity activity,
List<ModelHadiah> model) {
        this.context = context;
        this.activity = activity;
        this.model = model;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup
viewGroup, int i) {
        View view =
LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.customer_1
ist_hadiah, viewGroup, false);
        getUsername();
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MyViewHolder mv, int i) {
        final ModelHadiah m = model.get(i);
        mv.jenis.setText(m.getJenis());
        mv.judul.setText(m.getJudul());
        mv.jumlah.setText(m.getJumlah());
        mv.harga.setText(Util.get_rupiah(m.getHarga()) + " M-Poin");

        Glide.with(context).load(m.getUrlFoto()).into(mv.foto);
        mv.btnDetail.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                DialogDetailHadiah dialog = new
DialogDetailHadiah(activity);
                dialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
                dialog.getKode(m.getKode());
                dialog.setCancelable(true);
                dialog.show();
            }
        });

        final String kode = m.getKode();

        mv.btnTukar.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View v) {
            if (!Util.get_internet(context)) {
                dialogPesan("Gagal",
Util.util_get_pesan_no_internet, "tutup");
            } else {
                if (Integer.parseInt(m.getJumlah()) == 0) {
                    dialogPesan("Gagal", "Maaf, hadiah sudah
habis", "tutup");
                } else {
                    double dHarga = Double.valueOf(m.getHarga());
                    double hasil = poinku - dHarga;
                    strSisa =
Util.get_double_number_to_string(hasil);

                    if (poinku < dHarga) {
                        dialogPesan("Gagal", "Maaf, M-Poin Anda
kurang", "Kembali");
                    } else {
                        int intJumlah =
Integer.parseInt(m.getJumlah()) - 1;
                        String jumlahHadiah =
String.valueOf(intJumlah);
                        Toast.makeText(context, "kode: " + kode +
", sisa: " + jumlahHadiah, Toast.LENGTH_SHORT).show();
                        FirebaseDatabase.getInstance()
                            .getReference()
                            .child(Util.util_data_hadiah)
                            .child(kode)
                            .child("jumlah")
                            .setValue(jumlahHadiah);

                        FirebaseDatabase.getInstance()
                            .getReference()
                            .child(Util.util_data_customer)
                            .child(username)
                            .child("poin")
                            .setValue(strSisa);

                        simpan(m);
                    }
                }
            }
        });
    }

    @Override
    public int getItemCount() {
        return model.size();
    }

    public class MyViewHolder extends RecyclerView.ViewHolder {
        private CircleImageView foto;
        private TextView judul, harga, jenis, jumlah;
        private Button btnTukar, btnDetail;

        public MyViewHolder(@NonNull View itemView) {
            super(itemView);
            jumlah =
itemView.findViewById(R.id.customer_list_hadiah_jumlah);
            foto =
itemView.findViewById(R.id.customer_list_hadiah_foto);

```



```

        judul =
itemView.findViewById(R.id.customer_list_hadiah_judul);
        harga =
itemView.findViewById(R.id.customer_list_hadiah_harga);
        jenis =
itemView.findViewById(R.id.customer_list_hadiah_jenis);
        btnTukar =
itemView.findViewById(R.id.customer_list_hadiah_btn_tukar);
        btnDetail =
itemView.findViewById(R.id.customer_list_hadiah_btn_detail);
    }
}

private void simpan(ModelHadiah m) {
    String UID = Util.get_UID();
    final String kode = UID.substring(0, 3) +
Util.get_kode_hadiah();
    ModelHadiah mh = new ModelHadiah(m.getUrlFoto(),
        m.getJudul(),
        kode.toUpperCase(),
        "-",
        m.getHarga(),
        m.getJenis(),
        "1",
        m.getDeskripsi(),
        "-",
        Util.util_get_kondisi_belum_ditukar,
        username,
        "1",
        Util.get_tanggal_dan_waktu_sekarang());

    reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_hadi
ah_customer);
    reference.child(kode).setValue(mh).addOnCompleteListener(new
OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()) {
                dialogPesan("Berhasil", "Terima kasih, transaksi
berhasil dilakukan. Sisa poin: " + Util.get_rupiah(strSisa) + " M-
Poin.", "Terima kasih");
            }
        }
    });
}

private void getUsername() {
    String UID = Util.get_UID();
    reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_peng
guna).child(UID);
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            if (dataSnapshot.exists()) {
                ModelPengguna m =
dataSnapshot.getValue(ModelPengguna.class);
                username = m.getUsername();
                getDataPengguna(username);
            }
        }
    })
}

```



```

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

            }

        });
    }

    private void getDataPengguna(String username) {
        reference =
        FirebaseDatabase.getInstance().getReference().child(Util.util_data_cust
omer).child(username);
        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                if (dataSnapshot.exists()) {
                    ModelCustomer m =
dataSnapshot.getValue(ModelCustomer.class);
                    String poin = m.getPoin();
                    poinku = Double.valueOf(poin);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError
databaseError) {

            }

        });
    }

    private void dialogPesan(String judul, String pesan, String tombol)
{
        AlertDialog.Builder builder = new AlertDialog.Builder(context)
            .setCancelable(false)
            .setTitle(judul)
            .setMessage(pesan)
            .setPositiveButton(tombol, new
Dialoginterface.OnClickListener() {
                @Override
                public void onClick(Dialoginterface dialog, int
which) {

                    dialog.cancel();

                }
            });
        AlertDialog alertDialog = builder.create();
        alertDialog.show();
    }
}

```

#### E.4. Melihat *Leaderboard*

```

package
com.universitasmember.basirudin.aplikasiMacarina.view.customer.menu.lea
derboard;

import android.content.DialogInterface;
import android.content.Intent;

```

```

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.universitaskjember.basirudin.aplikasiMacarina.R;
import
com.universitaskjember.basirudin.aplikasiMacarina.adapter.customer.menu.
leaderboard.Customer_adapter_leaderboard;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelLeaderboard
;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelMacarina;
import
com.universitaskjember.basirudin.aplikasiMacarina.model.ModelPengguna;
import com.universitaskjember.basirudin.aplikasiMacarina.util.Util;
import
com.universitaskjember.basirudin.aplikasiMacarina.view.customer.Home_cus
tomer;

import java.util.ArrayList;
import java.util.List;

public class Customer_leaderboard extends AppCompatActivity {
    private RecyclerView rvLeader;
    private ImageView btnBack, btnInfo;
    private TextView txtReward, txtPoinku;

    private List<ModelLeaderboard> model;
    private Customer_adapter_leaderboard adapter;

    private DatabaseReference reference;

    private String strUsername = "";

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.customer_activity_leaderboard);

        rvLeader = findViewById(R.id.customer_leaderboard_recycleview);
        btnBack = findViewById(R.id.customer_leaderboard_btn_back);
        btnInfo = findViewById(R.id.customer_leaderboard_btn_info);
        txtReward = findViewById(R.id.customer_leaderboard_reward);
        txtPoinku = findViewById(R.id.customer_leaderboard_poinku);

        getLeaderboard();
        model = new ArrayList<>();
        adapter = new Customer_adapter_leaderboard(this, model);
        RecyclerView.LayoutManager layoutManager = new
LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);

```

```

rvLeader.setLayoutManager(layoutManager);
rvLeader.setAdapter(adapter);

btnInfo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        pesan();
    }
});

btnBack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        kembali();
    }
});
}

private void getLeaderboard() {
    getUsername();
    getReward();
    getData();
}

@Override
public void onBackPressed() {
    super.onBackPressed();
    kembali();
}

private void kembali() {
    startActivity(new Intent(getApplicationContext(),
Home_customer.class));
}

private void getUsername() {
    reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_peng
guna).child(Util.get_UID());
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            if (dataSnapshot.exists()) {
                ModelPengguna m =
dataSnapshot.getValue(ModelPengguna.class);
                strUsername = m.getUsername();

                getPoin();
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

        }
    });
}

private void getPoin() {
    reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_lead

```

```

erboard).child(strUsername);
reference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        if (dataSnapshot.exists()) {
            ModelLeaderboard m =
dataSnapshot.getValue(ModelLeaderboard.class);
            String poin = m.getPoin();

            txtPoinku.setText(Util.get_rupiah(poin) + " M-
Poin");
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError
databaseError) {
    }
});
}

private void getReward() {
reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_Maca
rina).child("leaderboard");
reference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        if (dataSnapshot.exists()) {
            ModelMacarina m =
dataSnapshot.getValue(ModelMacarina.class);
            String reward = m.getLeaderboard();

            txtReward.setText(Util.get_rupiah(reward) + " M-
Cash");
        } else {
            txtReward.setText(Util.get_rupiah("0") + " M-
Cash");
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError
databaseError) {
    }
});
}

private void getData() {
reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_lead
erboard);
reference.orderByChild("desc").addValueEventListener(new
ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        if (dataSnapshot.exists()) {
            model.clear();

```

```

        for (DataSnapshot snapshot:
dataSnapshot.getChildren()) {
            ModelLeaderboard m =
snapshot.getValue(ModelLeaderboard.class);
            String nama = m.getNama();
            String foto = m.getFoto();
            String poin = m.getPoin();
            String desc = m.getDesc();
            String user=m.getUsername();

            m = new ModelLeaderboard(foto, user, nama,
poin, desc);

            model.add(m);
            adapter.notifyDataSetChanged();
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError
databaseError) {
        }
    });
}

private void pesan() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this)
        .setCancelable(false)
        .setTitle("Informasi")
        .setMessage("Poin bulanan akan di reset ulang ke 0 pada
setiap awal bulan yang bersamaan dengan pemberian reward kepada
Customer yang berada pada urutan No 1")
        .setPositiveButton("tutup", new
Dialoginterface.OnClickListener() {
            @Override
            public void onClick(Dialoginterface dialog, int
which) {
                dialog.cancel();
            }
        });
    AlertDialog alertDialog = builder.create();
    alertDialog.show();
}
}
}

```

```

package
com.universitasmember.basirudin.aplikasiMacarina.adapter.customer.menu.
leaderboard;

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;

```



```

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.universitaskjember.basirudin.aplikasiMacarina.R;
import com.universitaskjember.basirudin.aplikasiMacarina.model.ModelLeaderboard;
;
import com.universitaskjember.basirudin.aplikasiMacarina.model.ModelPengguna;
import com.universitaskjember.basirudin.aplikasiMacarina.util.Util;

import java.util.List;

import de.hdodenhof.circleimageview.CircleImageView;

public class Customer_adapter_leaderboard extends
RecyclerView.Adapter<Customer_adapter_leaderboard.MyViewHolder> {
    private Context context;
    private List<ModelLeaderboard> model;
    String strUsername="";
    private DatabaseReference reference;

    public Customer_adapter_leaderboard(Context context,
List<ModelLeaderboard> model) {
        this.context = context;
        this.model = model;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup
viewGroup, int i) {
        View view =
LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.customer_l
ist_leaderboard, viewGroup, false);
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull final MyViewHolder mv, int i)
{
        final ModelLeaderboard m = model.get(i);
        i++;
        mv.no.setText(String.valueOf(i) + ".");
        mv.nama.setText(m.getNama());
        mv.poin.setText(Util.get_rupiah(m.getPoin())+" Poin");
        Glide.with(context).load(m.getFoto()).into(mv.foto);

        if (i == 1) {
            mv.medali.setVisibility(View.VISIBLE);
            mv.medali.setImageResource(R.drawable.ic_medal_gold);
        } else if (i == 2) {
            mv.medali.setVisibility(View.VISIBLE);
            mv.medali.setImageResource(R.drawable.ic_medal_silver);
        } else if (i == 3) {
            mv.medali.setVisibility(View.VISIBLE);
            mv.medali.setImageResource(R.drawable.ic_medal_bronze);
        } else {
            mv.medali.setVisibility(View.GONE);
        }
    }
}

```



```
String UID = Util.get_UID();
reference =
FirebaseDatabase.getInstance().getReference().child(Util.util_data_peng
guna).child(UID);
reference.addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
if (dataSnapshot.exists()) {
ModelPegguna md =
dataSnapshot.getValue(ModelPegguna.class);
strUsername = md.getUsername();

if (m.getUsername().equalsIgnoreCase(strUsername)) {
mv.simbol.setVisibility(View.VISIBLE);
} else {
mv.simbol.setVisibility(View.GONE);
}
}
}

@Override
public void onCancelled(@NonNull DatabaseError
databaseError) {
}
});
}

@Override
public int getItemCount() {
return model.size();
}

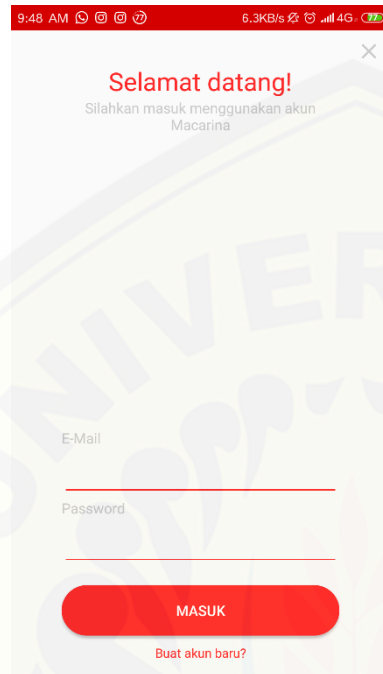
public class MyViewHolder extends RecyclerView.ViewHolder {
private TextView no, nama, poin;
private ImageView simbol;
private CircleImageView medali, foto;

public MyViewHolder(@NonNull View itemView) {
super(itemView);
no =
itemView.findViewById(R.id.customer_list_leaderboard_no_urut);
nama =
itemView.findViewById(R.id.customer_list_leaderboard_nama);
poin =
itemView.findViewById(R.id.customer_list_leaderboard_poin);
medali =
itemView.findViewById(R.id.customer_list_leaderboard_medali);
foto =
itemView.findViewById(R.id.customer_list_leaderboard_foto);
simbol=itemView.findViewById(R.id.customer_list_leaderboard_simbol);
}
}
}
```

F. Tampilan Aplikasi

a. Umum

1. Masuk



9:48 AM 6.3KB/s 4G

Selamat datang!  
Silahkan masuk menggunakan akun  
Macarina

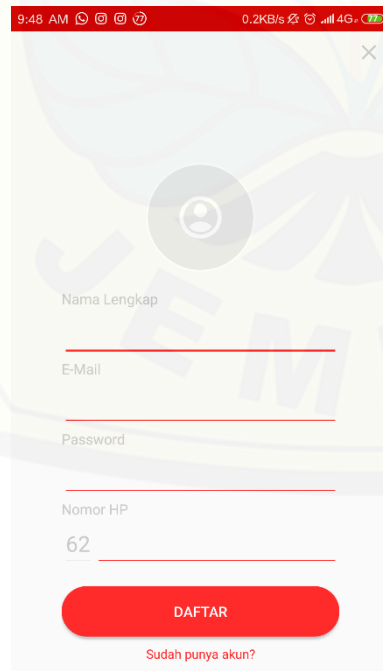
E-Mail

Password

MASUK

Buat akun baru?

2. Daftar



9:48 AM 0.2KB/s 4G

Nama Lengkap

E-Mail

Password

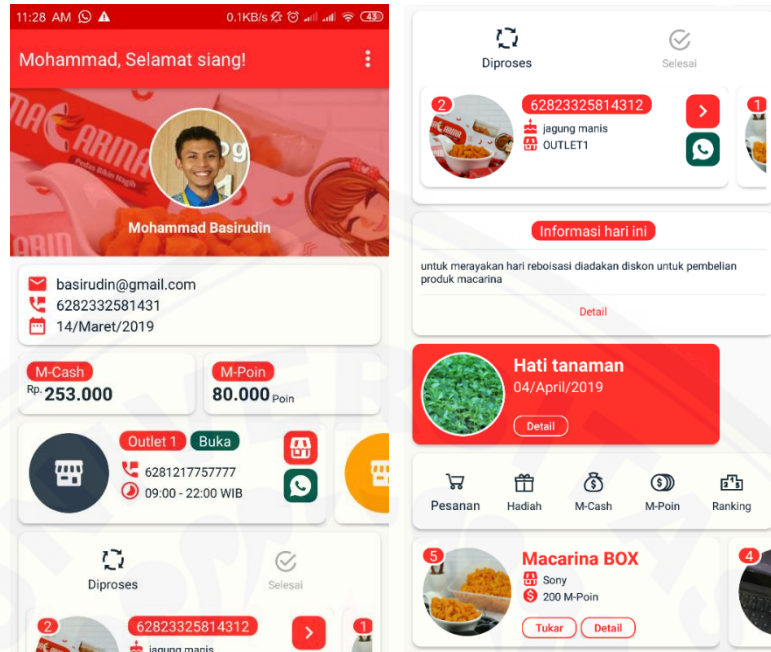
Nomor HP  
62

DAFTAR

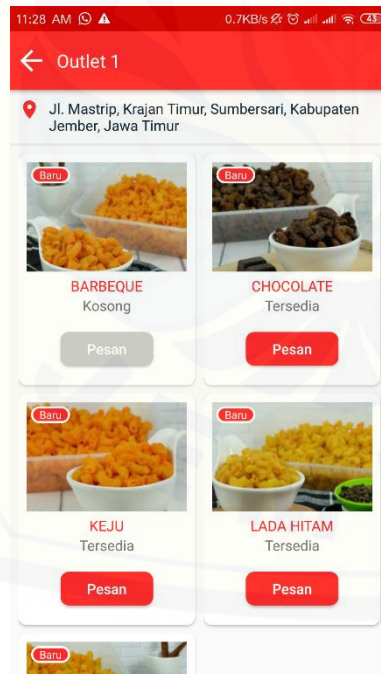
Sudah punya akun?

b. Customer

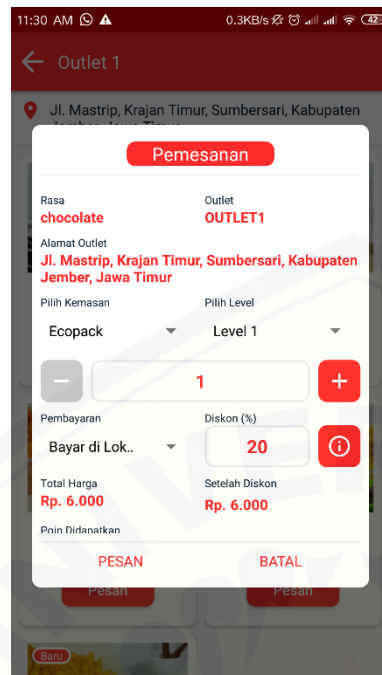
1. Home



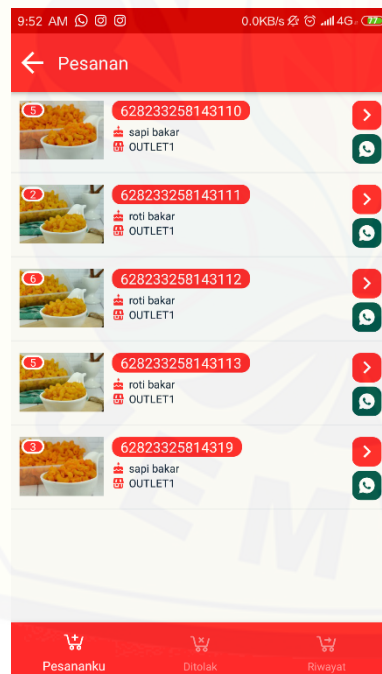
2. Daftar Rasa



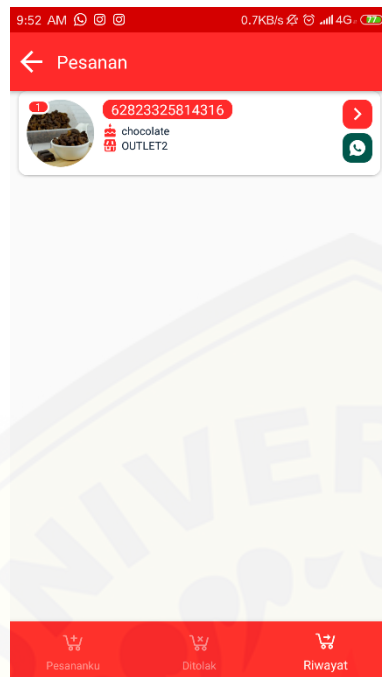
3. Pemesanan



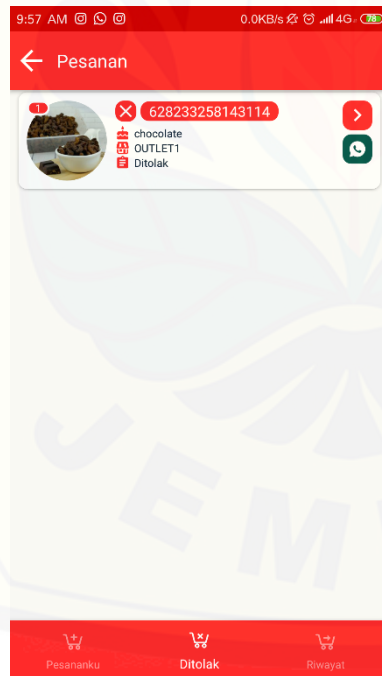
#### 4. Pesanan belum diproses



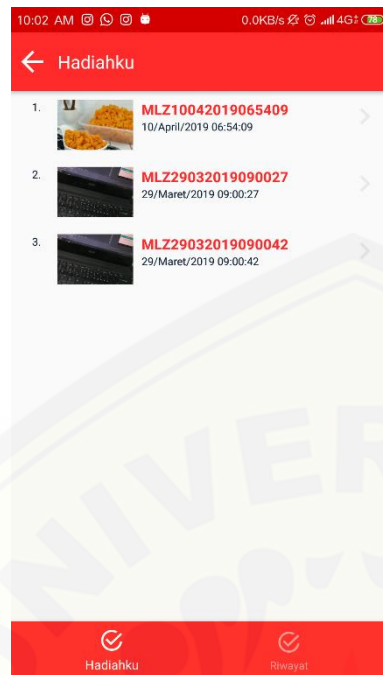
#### 5. Pesanan sudah diambil



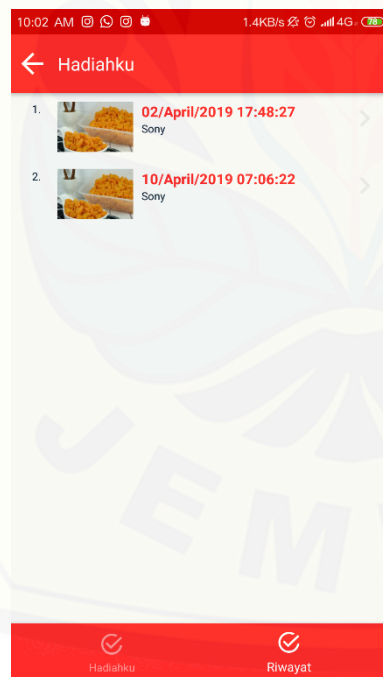
6. Pesanan ditolak



7. Hadiah belum ditukar

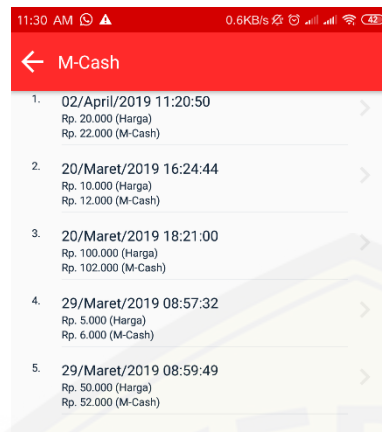


#### 8. Hadiah sudah ditukar

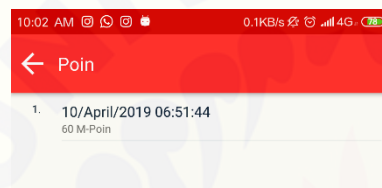


#### 9. Riwayat *M-Cash*

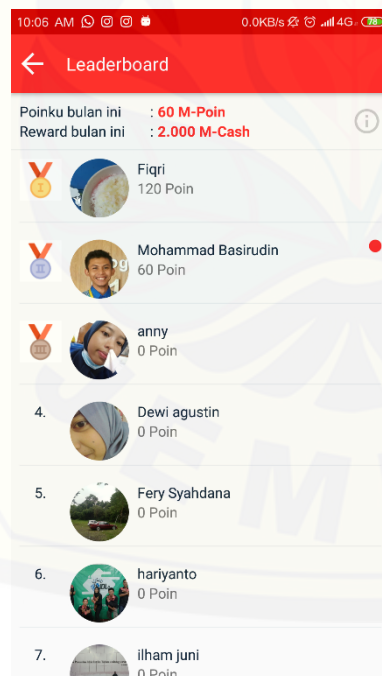




## 10. Riwayat *M-Poin*

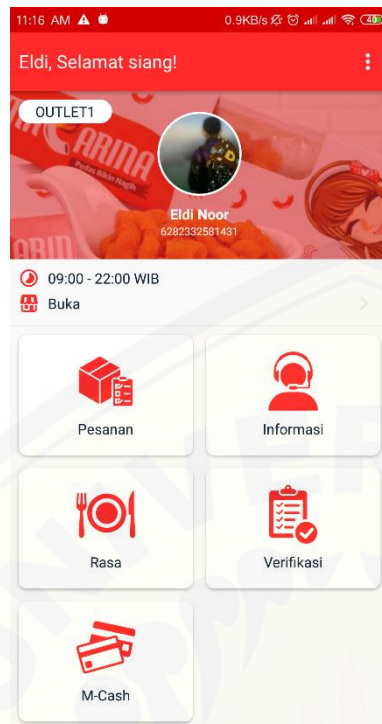


## 11. *Leaderboard*



### c. Karyawan

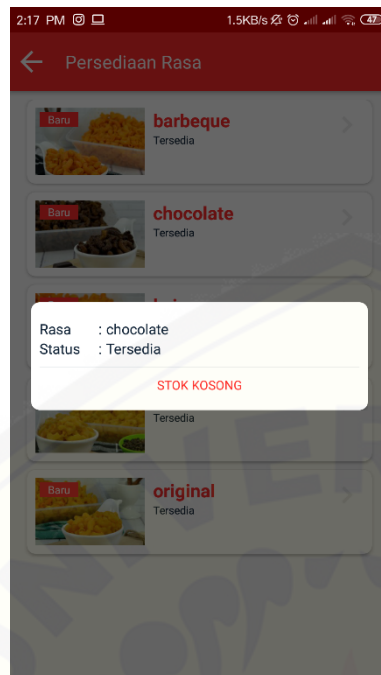
#### 1. *Home*



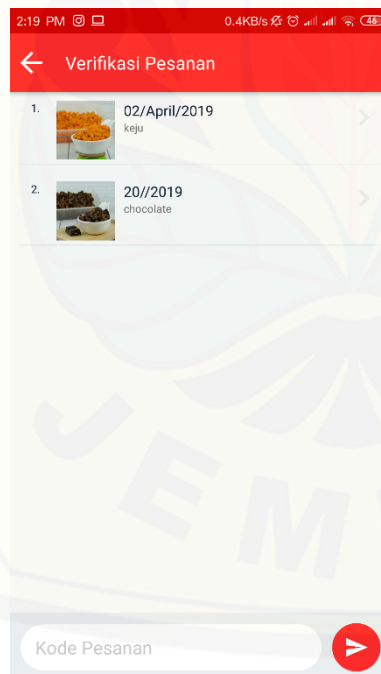
## 2. Daftar Rasa



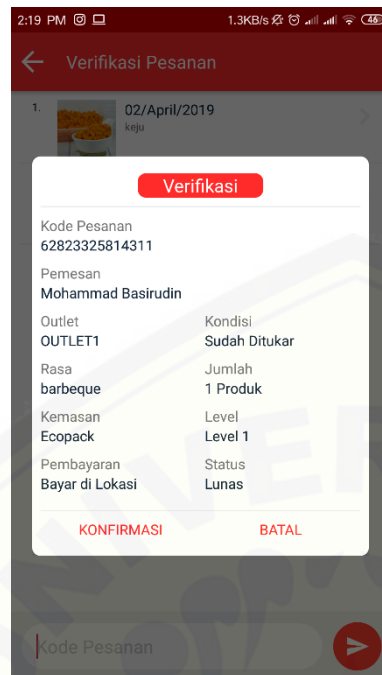
## 3. Melakukan Perubahan Status Persediaan



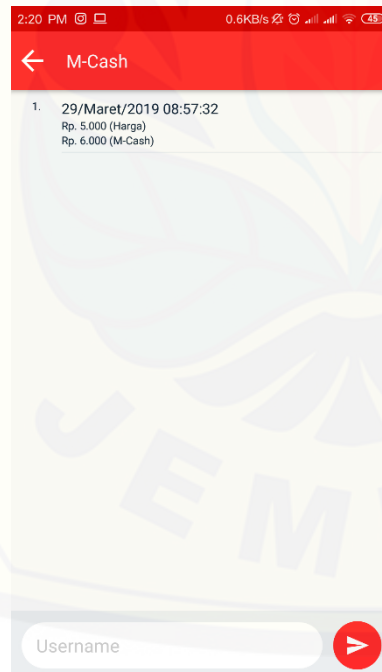
#### 4. Riwayat Verifikasi Pesanan



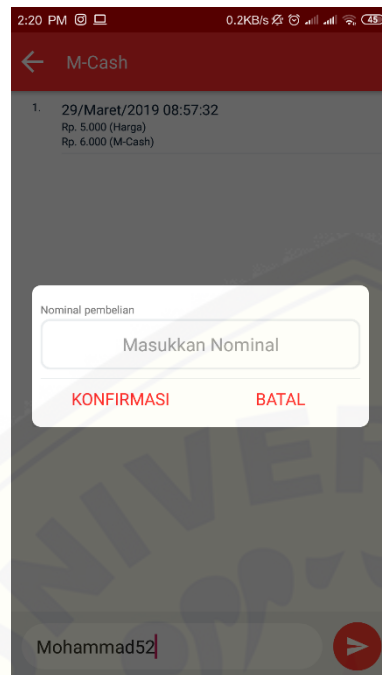
#### 5. Verifikasi Pesanan



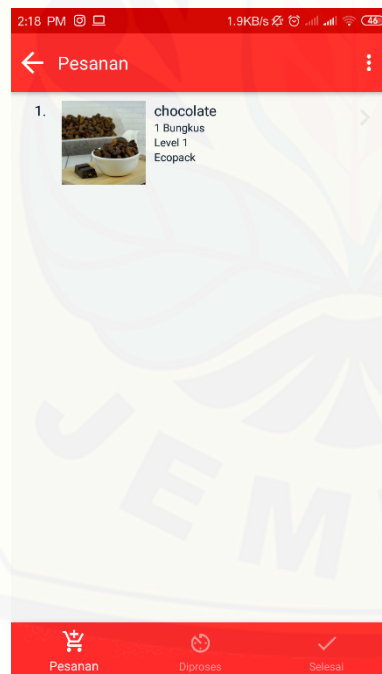
#### 6. Riwayat *M-Cash*



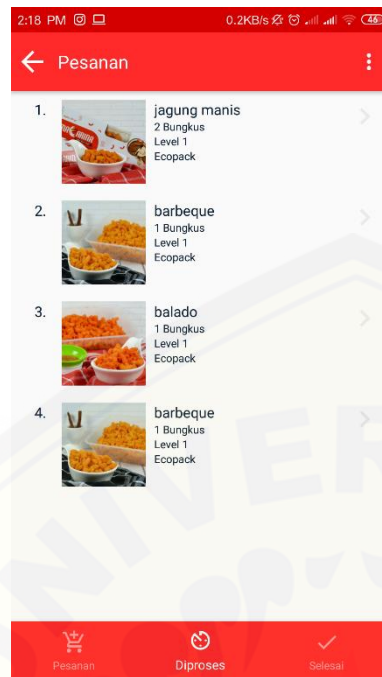
#### 7. Pengisian *M-Cash*



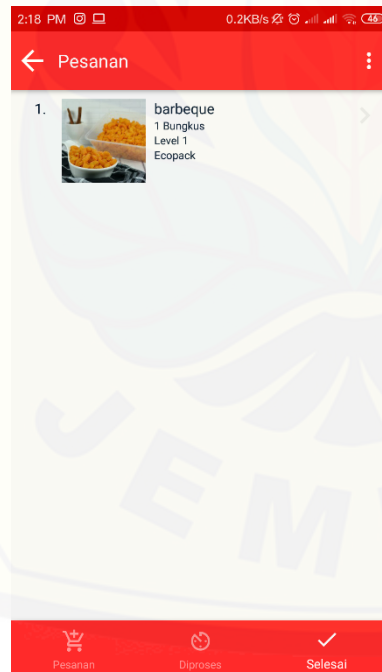
#### 8. Pesanan Belum Diproses



#### 9. Pesanan Sedang Diproses

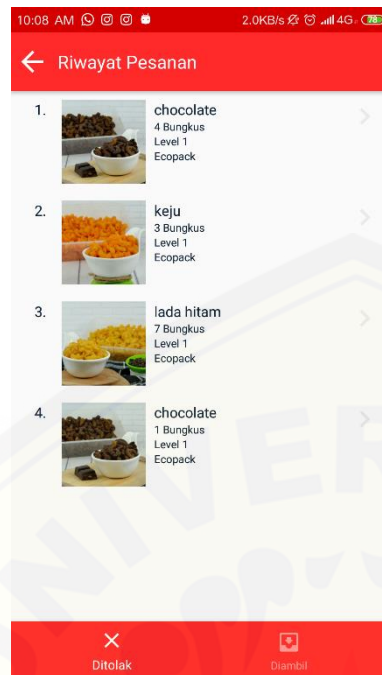


#### 10. Pesanan Sudah Selesai Diproses

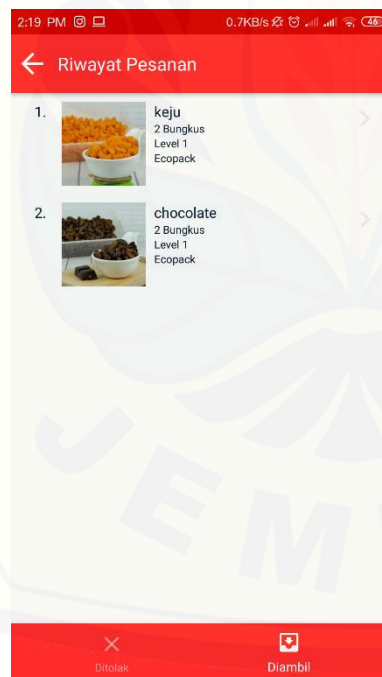


#### 11. Pesanan Ditolak



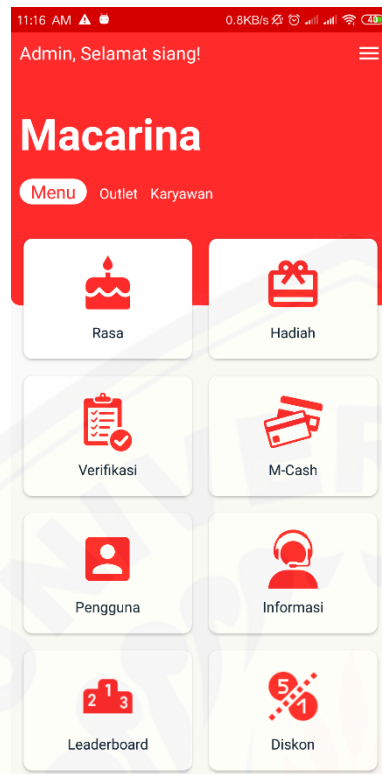


## 12. Pesanan Sudah Diambil

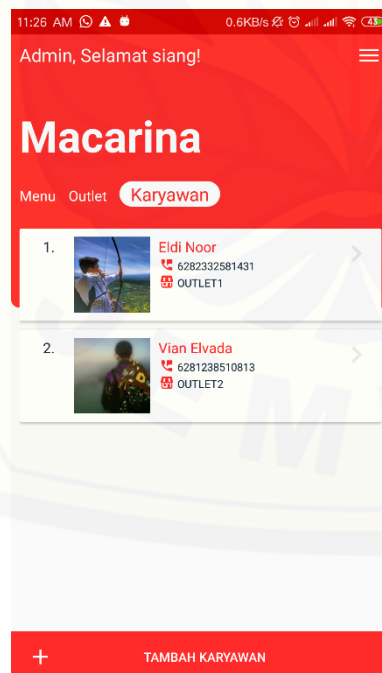


d. *Admin*

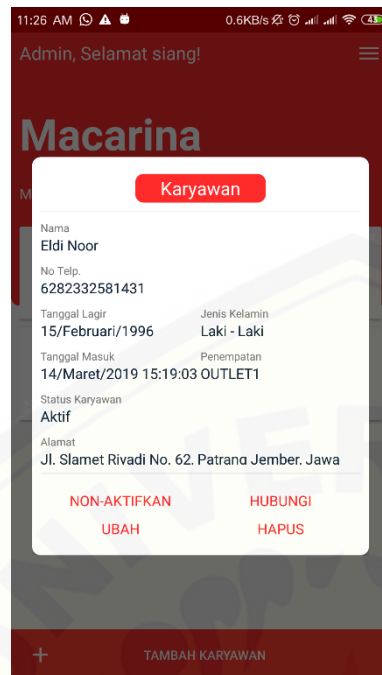
1. Home



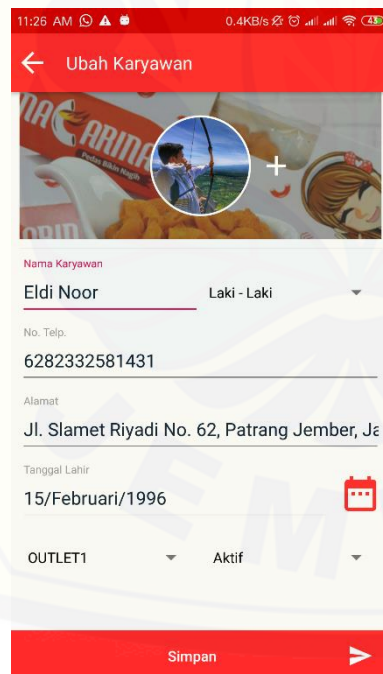
## 2. Daftar Karyawan



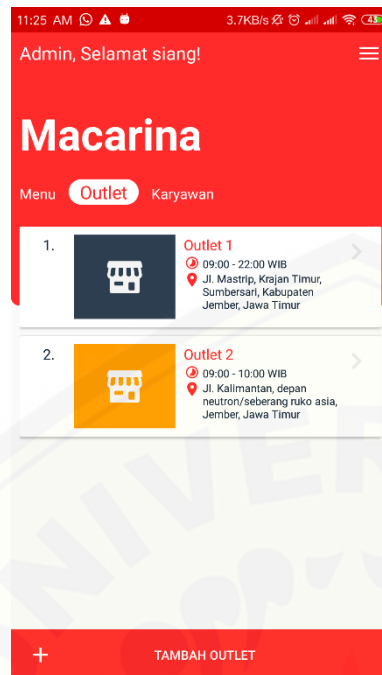
## 13. Detail Karyawan



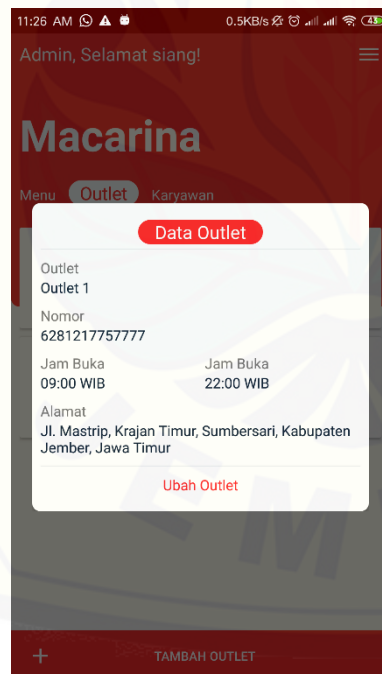
#### 14. Mengubah Data Karyawan



#### 3. Daftar Outlet



### 15. Detail *Outlet*



### 16. Menambah *Outlet*

11:26 AM 0.9KB/s

← Tambah Outlet

+

Nama Outlet  
**Outlet 3**

Jam Buka **07:00 WIB** Jam Tutup **22:00 WIB**

Status Outlet  
**Buka**

Telp. Outlet  
**082332581431**

Alamat Outlet  
**Jl. Jawa**

Simpan

#### 17. Mengubah *Outlet*

11:26 AM 0.3KB/s

← Ubah Data Outlet

+

Nama Outlet  
**Outlet 1**

Jam Buka **09:00 WIB** Jam Tutup **22:00 WIB**

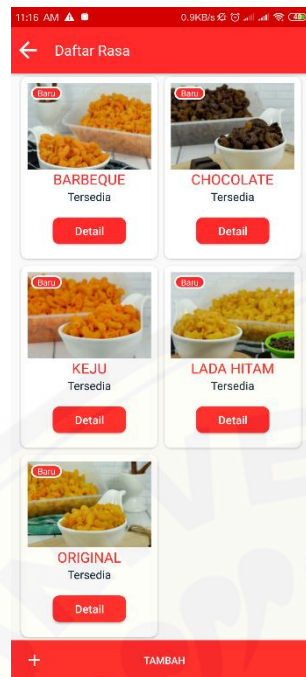
Status Outlet  
**Buka**

Telp. Outlet  
**628121775777**

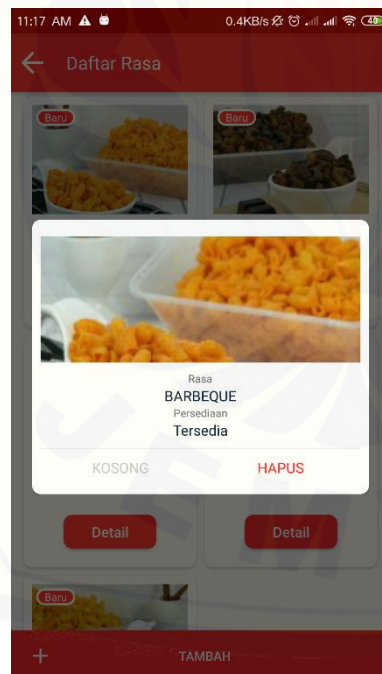
Alamat Outlet  
**Jl. Mastrip, Krajan Timur, Sumbersari, Kabu**

Simpan

#### 4. Daftar Rasa

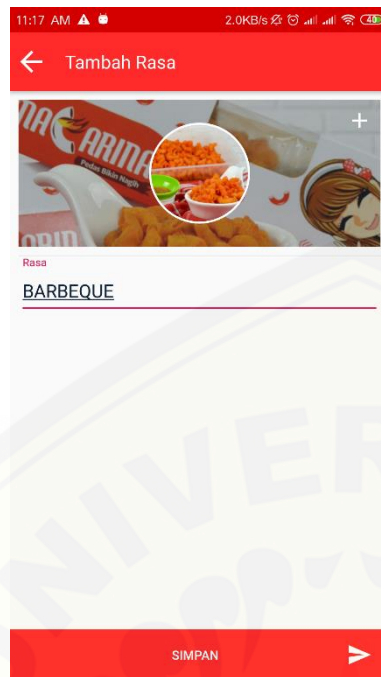


18. Detail Rasa

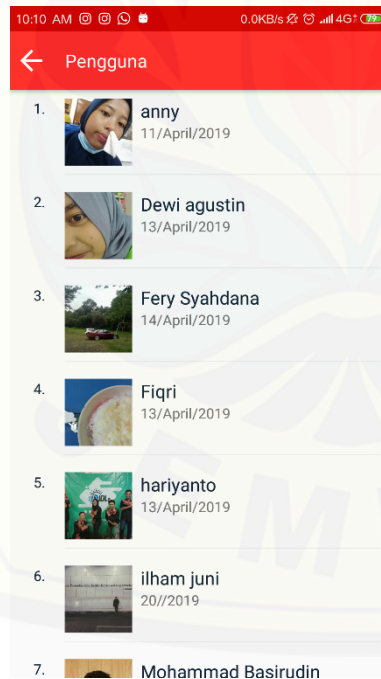


19. Menambah Rasa

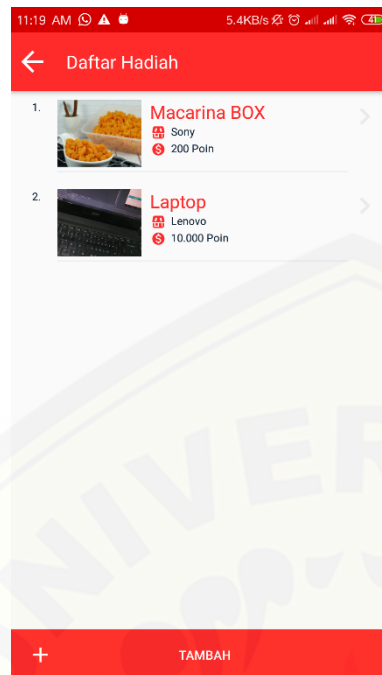




## 5. Daftar Pengguna



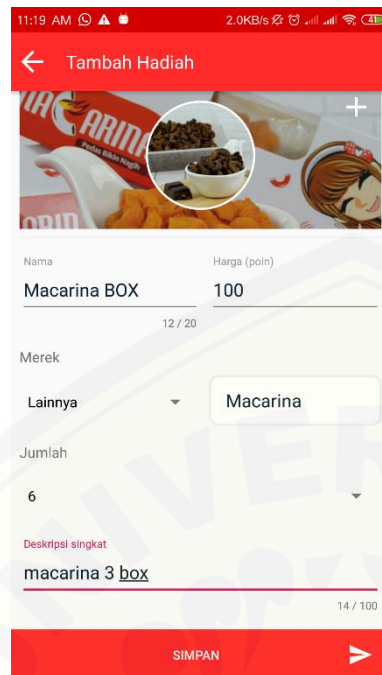
## 6. Daftar Hadiah



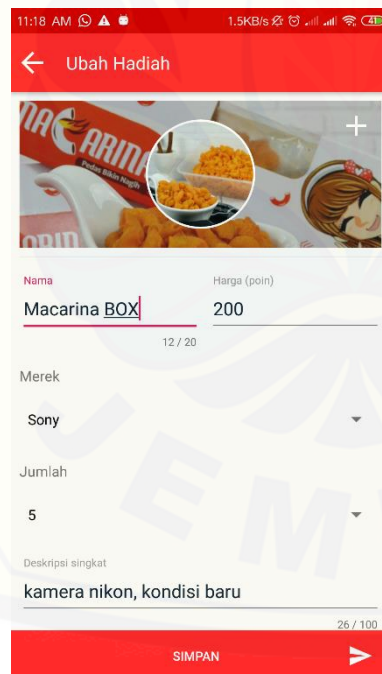
## 20. Detail Hadiah



## 21. Menambah Hadiah



## 22. Mengubah Hadiah



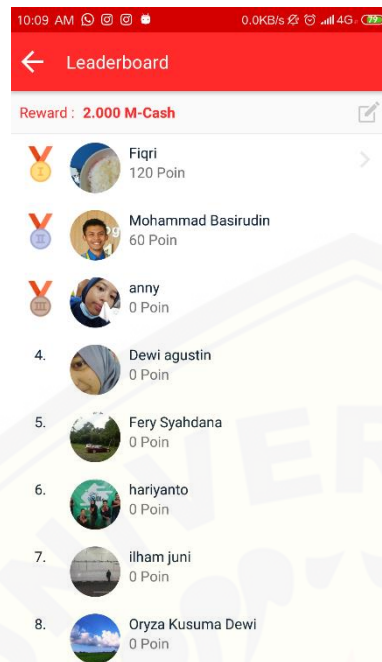
## 7. Riwayat *M-Cash*



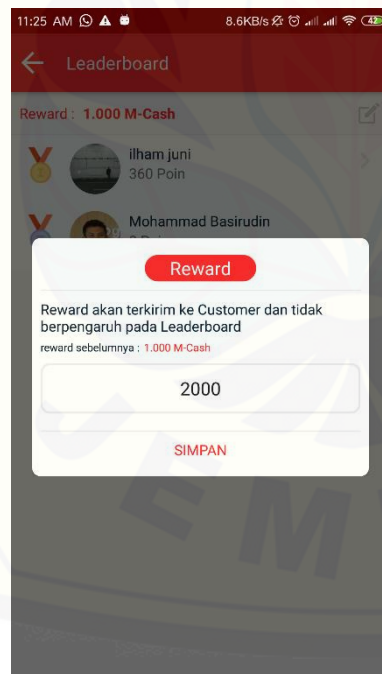
### 23. Pengisian *M-Cash*



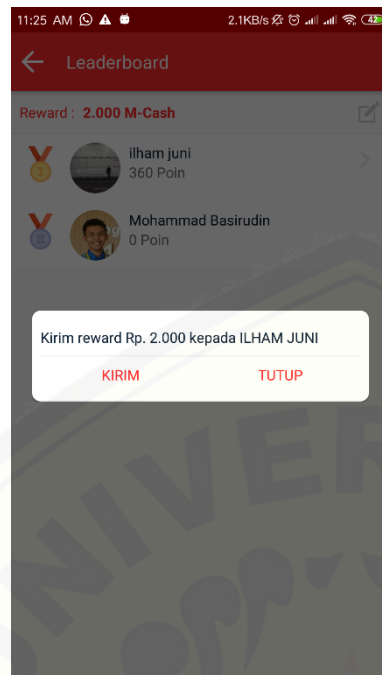
### 8. *Leaderboard*



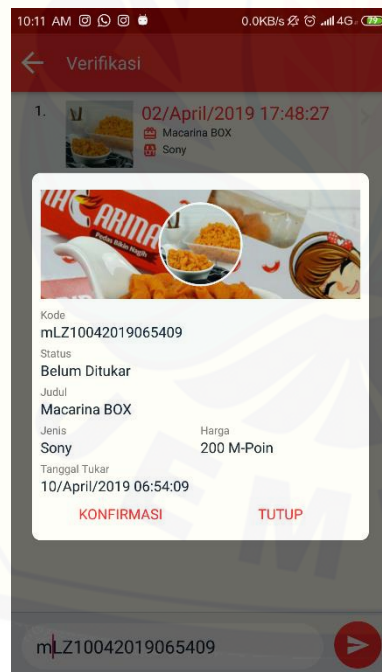
#### 24. Mengubah *Reward*



#### 25. Mengirim *Reward*

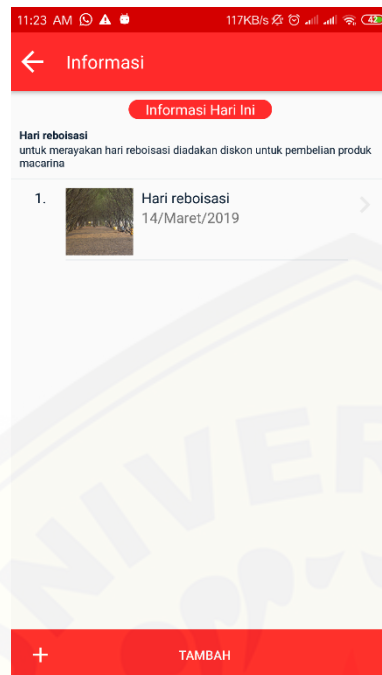


## 9. Verifikasi Hadiah

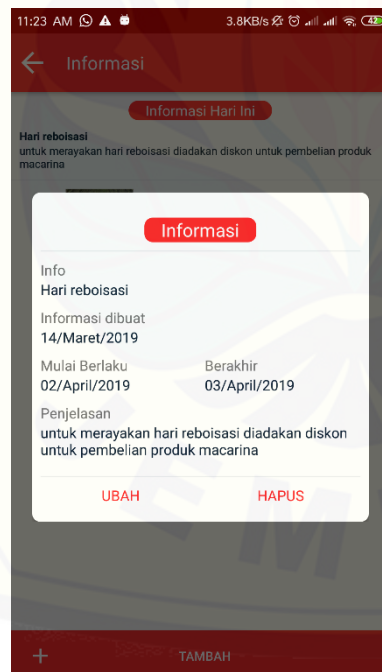


## 10. Daftar Info

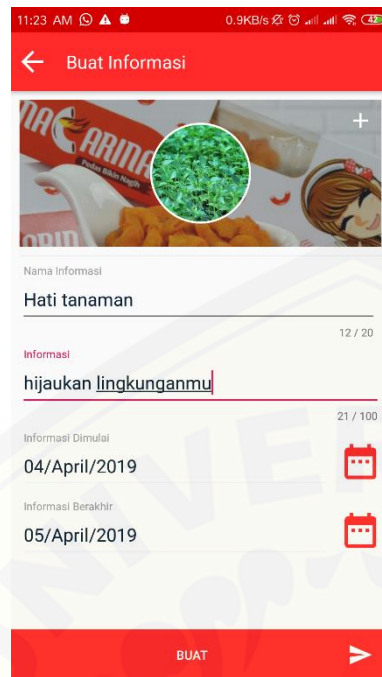




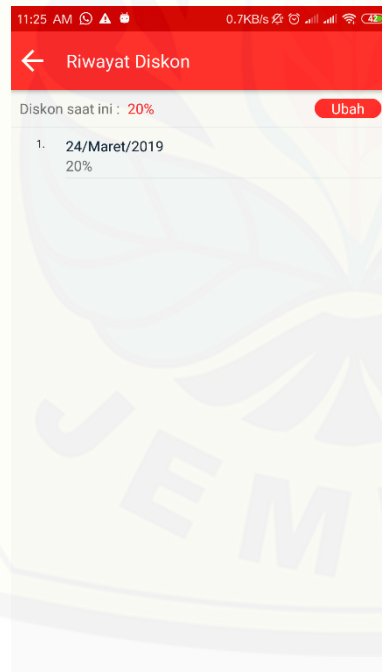
## 26. Detail Informasi



## 27. Menambah Informasi



## 11. Riwayat Diskon



## 28. Merubah Diskon

