

**SM-MART:
APLIKASI CLIENT SUPERMARKET ONLINE
BERBASIS ANDROID**

TUGAS AKHIR

Oleh :

Monalisa 3311101044

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2014**

LEMBAR PENGESAHAN

Batam, 7 Juli 2014

Pembimbing,

Nur Zahрати Janah, M. Sc

112087

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3311101044

Nama : Monalisa

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

SM-Smart: Aplikasi Client Supermarket Online Berbasis Android

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain,
2. tidak melakukan pemalsuan data,
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik.

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Batam, 7 Juli 2014

Monalisa

3311101044

HALAMAN PERSEMBAHAN

Ku persembahkan Tugas Akhir ini teruntuk :

Motivator dan inspirator dalam hidupku, kedua orang tuaku, *Ibunda Nurmala* dan *Ayahanda Bahami*.

KATA PENGANTAR

Innalhamda Lillah, wassholatu wassalamu ‘alaa Rasulillah wa’ala alihi washohbihi.

Puji syukur kehadirat Allah SWT. atas segala nikmat dan karuniaNya yang tak terhingga di bulan Ramadhan yang penuh berkah ini, hingga Laporan Tugas Akhir yang berjudul “SM-Mart, Client Aplikasi Supermarket Online Berbasis Android” ini dapat terselesaikan. Sholawat beriring salam selalu tercurahkan pada junjungan kita, manusia mulia yang paling kita rindukan, Muhammad Sallahu ‘Alaihi Wasallam.

Laporan ini ditulis guna melengkapi syarat kelulusan Program Studi Teknik Informatika Politeknik Negeri Batam. Dalam pengerjaan Laporan Tugas Akhir ini, Penulis banyak mendapatkan arahan dan bimbingan dari berbagai pihak. Oleh sebab itu, dalam kesempatan ini Penulis hendak mengucapkan ungkapan terima kasih tak terhingga kepada:

1. **Allah SWT.**, yang senantiasa memberikan limpahan anugerah dan nikmat, berupa nikmat iman, nikmat Islam dan nikmat kesehatan yang selalu mengiringi Penulis.
2. **Orang Tua Tersayang, Ibunda Nurmala dan Ayahanda Bahami**, yang selalu menjadi alasan bagi Penulis untuk menjalani seluruh tahapan kehidupan. Terimakasih tak terhingga untuk segala kasih sayang, motivasi, dorongan semangat yang tak henti mengalir hingga saat sekarang ini.
3. **Kakak-kakak dan Adikku, Mariah, Marini, Bahadi, Bahara, Izhar**, terimakasih sudah menjadi saudara-saudara yang luar biasa.
4. **Kementrian Pendidikan dan Kebudayaan, Dirjen Pendidikan Tinggi**, yang telah memberikan kesempatan Penulis mengenyam pendidikan tinggi ini tanpa biaya sepeser pun melalui Program Beasiswa Bidikmisi. Semoga semakin memberikan manfaat dalam mencerdaskan kehidupan bangsa.
5. **Ibu Nur Zahрати Janah, M. Sc.**, selaku Pembimbing Tugas Akhir, yang telah bersedia memberikan masukan, nasihat dan dukungannya kepada

Penulis. Terimakasih untuk waktu, tenaga dan pikirannya. Semoga kebaikan Ibu senantiasa dibalas oleh Allah dengan balasan yang indah.

6. **Bapak Tri Ramadani Arjo, S. ST.**, selaku dosen wali Teknik Informatika angkatan 2011.
7. **Bapak Dwi Ely Kurniawan, M. Kom.**, selaku dosen pengampu mata kuliah Tugas Akhir.
8. **Teman seperjuangan TA, Suharjo**, yang telah sangat banyak membantu dalam penyelesaian Tugas Akhir ini, semoga semua kebaikanmu akan kembali padamu.
9. **Aldilla Yulandina, Asti Nuzulul Hidayah, Rasinta Br. Bangun, Okta Susilowati**, terimakasih telah menjadi tempat berbagi suka dan duka.
10. **Teman-teman IF.B**, terimakasih atas kebersamaan selama 3 tahun ini. Semoga apa yang kita impikan menjadi kenyataan di masa yang akan datang.
11. **Kak Devina Prahasanti Putri**, yang selalu menjadi kakak, guru serta teman yang selalu memberikan dorongan dan referensi dalam pengerjaan Tugas Akhir dan Laporan Tugas Akhir ini.
12. **Keluarga Besar Forum Indonesia Muda diseluruh Indonesia**, yang senantiasa memberikan inspirasi, dukungan dan semangat serta doa. Semoga terus menginspirasi dan berkontribusi bagi bangsa dan Negara.
13. **Bapak/Ibu, Pak De/Bu De, Om/Tante, Mas/Mba, Abang/Kakak, Adik dan berbagai pihak**, yang telah ikut serta membantu dan memberi kritik, saran doa, dukungan dan sejenisnya. Mohon maaf jika “Yang kecil tak disebut nama, yang besar tak disebut gelar, baik yang disengaja ataupun tidak”, sehingga Laporan Tugas Akhir dapat terselesaikan dengan baik. Semoga Allah SWT selalu memberikan nikmat, hikmah, rezeki dan pahala yang berlimpah untuk kita, Amin.

Penulis menyadari bahwa dalam penyusunan Laporan Tugas Akhir ini tentunya terdapat banyak kesalahan dikarenakan keterbatasan kemampuan yang penulis miliki. Oleh karena itu, Penulis sangat mengharapkan saran dan kritik yang

membangun demi kesempurnaan pada masa yang akan datang. Semoga Laporan Tugas Akhir ini dapat memberikan manfaat baik bagi penulis maupun pembaca.

Batam, 7 Juli 2014

Penulis

ABSTRAK

SM-Mart : APLIKASI CLIENT SUPERMARKET ONLINE BERBASIS ANDROID

Oleh

Monalisa (3311101044)

Belanja merupakan hal yang tidak dapat dilepaskan dari kehidupan sehari-hari. Pada umumnya, manusia berbelanja guna memenuhi kebutuhan yang tidak dapat di produksi sendiri. Namun, aktivitas berbelanja menghabiskan banyak waktu. Sehingga muncullah beberapa toko online yang menawarkan proses berbelanja yang lebih mudah. Akan tetapi, biasanya toko online tidak menjual kebutuhan sehari-hari. Oleh karena itu, perlu dibuat sebuah aplikasi yang mampu mempermudah pemesanan kebutuhan sehari-hari.

SM-Mart : aplikasi client supermarket online berbasis Android ini dapat diakses oleh perangkat dengan sistem operasi Android minimal versi 2.2(Froyo) dan terkoneksi dengan jaringan internet. Aplikasi ini dibangun dengan paradigma pengembangan berorientasi objek.

Aplikasi ini berhasil dibangun dengan memiliki kemampuan untuk menampilkan katalog yang diambil dari server, menangani proses pemesanan barang, menangani proses atur akun pembeli, dan mampu menampilkan status pesanan.

Kata kunci: Belanja, Supermarket Online, Android.

ABSTRACT

SM-Mart : ONLINE SUPERMARKET CLIENT APPLICATION BASED ON ANDROID

By

Monalisa (3311101044)

Shopping is unavoidable form people's daily activities. Usually, people shop to fullfill their needs that cannot be produced by themselves. However, shopping activities need a lot of time. Therefore, online shops emerge to offer easier shopping process. Unfortunately, most online shops do not sell grocery product. Hence, we need to develop an application that can handle the order of daily needs easly.

SM-Mart, an Android-based online client application that can be accessed by Android devices, minimum version 2.2 (Froyo) and are connected to internet connection. This application is built with object oriented software development paradigm.

It has been built successfully and has ability to display catalog that is fetched from the server, handle order process, set customer's account and display customer's order status.

Keywords: Shopping, Online Supermarket, Android.

DAFTAR ISI

SM-MART:	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
BAB II LANDASAN TEORI	4
2.1 Penelitian Sebelumnya.....	4
2.2 Toko Modern	4
2.3 E-Commerce	6
2.4 M-Commerce	8
2.5 Unified Modeling Language(UML)	9
2.6 Java	24
2.7 Android	25
2.8 MySQL	33
2.9 PHP	34
2.10 CSS	34
BAB III ANALISIS DAN PERANCANGAN	36
3.1 Analisis Sistem	36
3.1.1 Deskripsi Umum Sistem	36

3.1.2	Kebutuhan Fungsional	37
3.1.3	Kebutuhan Fungsional	37
3.1.4	Batasan Sistem	37
3.2	Perancangan Data	38
3.2.1	Entity Relationship Diagram.....	38
3.2.2	Struktur Tabel.....	39
3.3	Perancangan Data	42
3.3.1	Use Case Diagram.....	42
3.3.2	Skenario Use Case.....	43
3.3.3	Activity Diagram.....	45
3.3.4	Communication Diagram.....	50
3.3.5	Class Diagram	55
3.3.6	Sequence Diagram	56
3.4	Perancangan Antarmuka	62
3.4.1	Perancangan Antarmuka Login.....	62
3.4.2	Perancangan Antarmuka Kategori Barang.....	63
3.4.3	Perancangan Antarmuka Katalog Barang	64
3.4.4	Perancangan Antarmuka Pesan Barang	65
3.4.5	Perancangan Antarmuka Keranjang Belanja	66
3.4.6	Perancangan Antarmuka Konfirmasi Jenis Pengantaran	67
3.4.7	Perancangan Antarmuka Atur Akun	68
3.4.8	Perancangan Antarmuka Ganti Kata sandi	69
3.4.9	Perancangan Antarmuka Ganti Alamat dan Nomor HP	70
3.4.10	Perancangan Antarmuka Pesanan Saya	71
3.4.11	Perancangan Antarmuka Aturan Belanja.....	72
3.4.12	Perancangan Antarmuka Fitur Option	73
BAB IV IMPLEMENTASI DAN PENGUJIAN		74
4.1	Implementasi Struktur Tabel Basisdata	74
4.2	Implementasi Proses	77
4.2.1	Implementasi Proses Login	77
4.2.2	Implementasi Proses Tampil Kategori Barang	79

4.2.3	Implementasi Proses Tampil Katalog	81
4.2.4	Implementasi Proses Pesan Barang.....	83
4.2.5	Implementasi Proses Atur Akun	89
4.2.6	Implementasi Proses Tampil Pesanan Saya	93
4.2.7	Implementasi Proses Tampil Aturan Belanja.....	95
4.3	Pengujian	98
BAB V KESIMPULAN DAN SARAN		101
5.1	Kesimpulan	101
5.2	Saran	101

DAFTAR GAMBAR

Gambar 1 Sistem	13
Gambar 2 Use Case	14
Gambar 3 Actor	14
Gambar 4 Relationship	15
Gambar 5 Action States	16
Gambar 6 Action Flow	16
Gambar 7 Object Flow	17
Gambar 8 Communication Diagram	18
Gambar 9 Classes	19
Gambar 10 Active Class	20
Gambar 11 Visibility	20
Gambar 12 Asosiasi	21
Gambar 13 Actor	22
Gambar 14 Activation	22
Gambar 15 Lifeline	23
Gambar 16 Destroying Object	23
Gambar 17 Loop	24
Gambar 18 Arsitektur android	26
Gambar 19 Deskripsi Umum Sistem	36
Gambar 20 Entity Relationship Diagram	38
Gambar 21 Use Case Diagram	43
Gambar 22 Activity Diagram login	45
Gambar 23 Activity Diagram tampilKategori	46
Gambar 24 Activity Diagram tampilKatalog	46
Gambar 25 Activity Diagram pesanBarang	47
Gambar 26 Activity Diagram aturAkun	48
Gambar 27 Activity Diagram tampilPesananSaya	49
Gambar 28 Activity Diagram tampilAturanBelanja	49
Gambar 29 Communication Diagram login	50

Gambar 30 Communication Diagram tampilKategori	51
Gambar 31 Communication Diagram tampilKatalog	52
Gambar 32 Communication Diagram pesanBarang	52
Gambar 33 Communication Diagram aturAkun	53
Gambar 34 Communication Diagram tampilPesananSaya	53
Gambar 35 Communication Diagram tampilAturanBelanja.....	54
Gambar 36 Class Diagram	55
Gambar 37 Sequence Diagram login	56
Gambar 38 Sequence Diagram tampilKategori	57
Gambar 39 Sequence Diagram tampilKatalog.....	58
Gambar 40 Sequence Diagram pesanBarang	58
Gambar 41 Sequence Diagram aturAkun	59
Gambar 42 Sequence Diagram tampilPesananSaya	60
Gambar 43 Sequence Diagram tampilPesananSaya	61
Gambar 44 Perancangan Antarmuka Login.....	62
Gambar 45 Perancangan Antarmuka Kategori Barang	63
Gambar 46 Perancangan Antarmuka Katalog Barang	64
Gambar 47 Perancangan Antarmuka Pesan Barang.....	65
Gambar 48 Perancangan Antarmuka Keranjang belanja	66
Gambar 49 Perancangan Antarmuka Konfirmasi Jenis Pengantaran	67
Gambar 50 Perancangan Antarmuka Atur Akun	68
Gambar 51 Perancangan Antarmuka Ganti Kata Sandi.....	69
Gambar 52 Perancangan Antarmuka Ganti Alamat dan Nomor HP.....	70
Gambar 53 Perancangan Antarmuka Pesanan Saya.....	71
Gambar 54 Perancangan Antarmuka Aturan Belanja	72
Gambar 55 Perancangan Antarmuka Fitur Option.....	73
Gambar 56 Implementasi Antarmuka Login.....	77
Gambar 57 Implementasi Antarmuka Kategori Barang.....	80
Gambar 58 Implementasi Antarmuka Katalog	82
Gambar 59 Implementasi Antarmuka Detail Barang.....	84
Gambar 60 Implementasi Antarmuka Keranjang Belanja	85

Gambar 61 Implementasi Antarmuka Jenis Pengantaran	86
Gambar 62 Implementasi Antarmuka Atur Akun.....	89
Gambar 63 Implementasi Antarmuka Ganti Kata Sandi.....	90
Gambar 64 Implementasi Antarmuka Ganti Alamat dan Nomor HP	91
Gambar 65 Implementasi Antarmuka Pesanan Saya	94
Gambar 66 Implementasi Antarmuka Aturan Belanja.....	96

DAFTAR TABEL

Tabel 1 Perbandingan Penelitian Sebelumnya	4
Tabel 2 Perbandingan <i>Super Market Online</i>	4
Tabel 3 Messages	23
Tabel 4 Batasan Sistem	38
Tabel 5 Struktur Tabel t_pembeli	39
Tabel 6 Struktur Tabel t_pesanan	39
Tabel 7 Struktur Tabel t_barang	40
Tabel 8 Struktur Tabel t_kategori	40
Tabel 9 Struktur Tabel t_karyawan	41
Tabel 10 Struktur Tabel t_detail_pesanan.....	41
Tabel 11 Struktur Tabel t_pengantaran.....	41
Tabel 12 Struktur Tabel t_satuan	42
Tabel 13 Struktur Tabel t_stok.....	42
Tabel 14 Deskripsi Perancangan Antarmuka Login	62
Tabel 15 Deskripsi Perancangan Antarmuka Kategori Barang	63
Tabel 16 Deskripsi Perancangan Antarmuka Katalog Barang.....	64
Tabel 17 Deskripsi Perancangan Antarmuka Pesan Barang	65
Tabel 18 Deskripsi Perancangan Antarmuka Keranjang belanja.....	66
Tabel 19 Deskripsi Perancangan Antarmuka Konfirmasi Jenis Pengantaran.....	67
Tabel 20 Deskripsi Perancangan Antarmuka Atur Akun.....	68
Tabel 21 Deskripsi Perancangan Antarmuka Ganti Kata Sandi	69
Tabel 22 Deskripsi Perancangan Antarmuka Ganti Alamat dan Nomor HP	70
Tabel 23 Deskripsi Perancangan Antarmuka Pesanan Saya	71
Tabel 24 Deskripsi Perancangan Antarmuka Aturan Belanja.....	72
Tabel 25 Deskripsi Perancangan Antarmuka Fitur Option	73
Tabel 26 Implementasi Struktur Tabel t_pembeli.....	74
Tabel 27 Implementasi Struktur Tabel t_pesanan.....	74
Tabel 28 Implementasi Struktur Tabel t_barang.....	75
Tabel 29 Implementasi Struktur Tabel t_kategori	75

Tabel 30 Implementasi Struktur Tabel t_karyawan	75
Tabel 31 Implementasi Struktur Tabel t_detail_pesanan.....	75
Tabel 32 Implementasi Struktur Tabel t_pengantaran	76
Tabel 33 Implementasi Struktur Tabel t_satuan	76
Tabel 34 Implementasi Struktur Tabel t_stok.....	76
Tabel 35 Implementasi Struktur Tabel t_sementara	77
Tabel 36 Daftar Kode Sumber Halaman Login	78
Tabel 37 Penjelasan Kode clp.php	78
Tabel 38 Daftar Kode Sumber Halaman Kategori Barang	80
Tabel 39 Daftar Kode Sumber Halaman Kategori Barang	81
Tabel 40 Daftar Kode Sumber Halaman Katalog Barang.....	82
Tabel 41 Penjelasan Kode list.php	82
Tabel 42 Daftar Kode Sumber Halaman Detail Barang.....	84
Tabel 43 Daftar Kode Sumber Halaman Keranjang Belanja.....	85
Tabel 44 Daftar Kode Sumber Halaman Jenis Pengantaran	86
Tabel 45 Penjelasan Kode proses_fifo.php.....	86
Tabel 46 Penjelasan Kode wishlist.php	88
Tabel 47 Penjelasan Kode pilih.php.....	88
Tabel 48 Daftar Kode Sumber Halaman Atur Akun.....	89
Tabel 49 Daftar Kode Sumber Halaman Ganti Kata Sandi	90
Tabel 50 Daftar Kode Sumber Halaman Ganti Alamat dan Nomor HP	92
Tabel 51 Penjelasan Kode proses_ganti_password.php.....	92
Tabel 52 Penjelasan Kode proses_ganti_gan.php.....	92
Tabel 53 Daftar Kode Sumber Halaman Pesanan Saya	94
Tabel 54 Penjelasan Kode pesanansaya.php.....	95
Tabel 55 Daftar Kode Sumber Halaman Aturam Belanja	96
Tabel 56 Penjelasan Kode cara_belanja.php.....	97

BAB I

PENDAHULUAN

1.1 Latar Belakang

Berbelanja merupakan suatu kegiatan yang tidak dapat dilepaskan dari kehidupan manusia. Hal ini disebabkan karena segala sesuatu yang diperlukan manusia tentu saja tidak bisa dibuat sendiri sehingga berbelanja adalah alternatif yang dipilih untuk memenuhi semua kebutuhan yang diperlukan khususnya kebutuhan sehari-hari.

Selama ini, manusia menggunakan cara tradisional dalam melakukan kegiatan belanja yaitu belanja langsung. Belanja langsung adalah kegiatan yang dilakukan dengan mendatangi penjual, sehingga pembeli dapat memilih dan melihat contoh barang yang dibutuhkan secara langsung. Namun, tidak semua orang merasa nyaman dengan melakukan proses belanja langsung dikarenakan beberapa alasan. Salah satunya adalah keterbatasan waktu yang dimiliki dari pembeli yang memiliki kesibukan, misalnya pekerja.

Oleh karena itu, mulai muncul toko *online* yang menawarkan cara berbelanja secara *online*. Belanja berbasis *online* memanfaatkan jaringan Internet dalam setiap prosesnya. Penjual hanya perlu menampilkan barang dagangannya melalui foto dan mencantumkan harganya. Sedangkan pembeli tidak harus keluar rumah untuk mendapatkan barang yang dibutuhkan dan dapat menghemat waktu. Saat ini terdapat beberapa jenis toko *online* mulai dari *e-commerce*, *website*, *weblog*, forum dan media sosial seperti *facebook*, *kaskus*, *lazada.co.id*, *tokobagus.com* dan yang lainnya. Namun pada kenyataannya, belanja *online* juga memiliki beberapa kelemahan. Misalnya, sering kali foto barang yang terdapat di website tidak sesuai dengan aslinya, butuh waktu yang lama dan ongkos kirim yang besar untuk transaksi jual-beli yang berjauhan, dan rentan terhadap kasus penipuan, serta hanya menangani pembelian untuk barang-barang tertentu (tidak termasuk kebutuhan sehari-hari).

Kini, salah satu teknologi yang sedang naik daun adalah *smart phone* yang berbasis Android. Hampir seluruh vendor *smart phone* telah menggunakan Android sebagai sistem operasinya. Dengan perkembangan Android dan segala kelebihannya, membuat banyak pengguna *smartphone* memilih untuk menggunakan perangkat dengan sistem operasi Android.

Oleh karena ini perlu adanya aplikasi berbasis Android yang dapat menangani proses jual beli kebutuhan sehari-hari. Dengan aplikasi ini diharapkan masyarakat akan lebih mudah untuk melakukan belanja dan bukan tidak mungkin dengan teknologi ini dapat menjadi solusi dari permasalahan jual beli kebutuhan sehari-hari.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun *client* SM-Mart aplikasi supermarket berbasis Android?
2. Bagaimana agar aplikasi dapat memproses pemesanan barang yang dilakukan melalui *client* Android?
3. Bagaimana mengintegrasikan antara *server side* dan *client side* pada aplikasi SM-Mart?

1.3 Batasan Masalah

Beberapa batasan masalah yang akan dibahas dalam Tugas Akhir ini adalah sebagai berikut :

1. Aplikasi ini hanya menangani *client side*.
2. Aplikasi ini tidak menangani proses pembayaran.
3. Aplikasi ini tidak menangani keamanan sistem.
4. Validasi login menggunakan id *member card*.

1.4 Tujuan

Tujuan dari penulisan Tugas Akhir ini adalah :

1. Membangun *client* aplikasi supermarket *online* berbasis Android.
2. Mengintegrasikan data antara *server side* dan *client side*.

BAB II

LANDASAN TEORI

2.1 Penelitian Sebelumnya

Dalam tugas akhir ini referensi yang digunakan adalah TA yang berjudul "JobDroid"(Rizki dan Salidowati, 2012). Aplikasi Android ini bertujuan untuk memudahkan pencari kerja dalam mendapatkan informasi lowongan kerja yang tersedia untuknya secara cepat dan memudahkan dalam melakukan konfirmasi atas sebuah lowongan pekerjaan. Detail perbandingan dengan TA yang akan dibuat terlihat pada tabel 1.

Tabel 1 Perbandingan Penelitian Sebelumnya

Perbandingan	Tugas Akhir	Penelitian Sebelumnya
Bahasa Pemrograman	Java, PHP	Java, PHP
DBMS	MySQL	MySQL
Platform	Web dan Mobile Android	Web dan Mobile Android
Penggunaan aplikasi	Super Market	Lowongan Kerja

Selain referensi di atas, tugas akhir ini juga akan mengacu pada beberapa supermarket *online* yang sudah pernah dibangun sebelumnya. Supermarket *online* tersebut diantaranya adalah 54yur.com, ocado.com, dan carrefour.co.id. Dalam berbagai aspek, terdapat beberapa perbedaan diantara supermarket *online* tersebut dengan aplikasi yang akan dibangun. Perbedaan tersebut dapat dilihat pada tabel 2.

Tabel 2 Perbandingan Super Market Online

Nama Aplikasi	Platform	Aplikasi Android	
		Katalog Barang	Pemesanan
54yur.com	Website	-	-
Ocado.com	Website	-	-
Carrefour.co.id	Website, Android	Ya	Tidak
SM-Mart	Website, Android	Ya	Ya

2.2 Toko Modern

Toko Modern adalah toko dengan sistem pelayanan mandiri, menjual berbagai jenis barang secara eceran yang berbentuk *Minimarket*, *Supermarket*, *Department Store*, *Hypermarket* ataupun grosir yang berbentuk *Perkulakan*. Batasan Toko

Modern ini dipertegas dalam Permendag RI Nomor:53/M-DAG/PER/12/2008 bab IV, dalam hal luas lantai penjualan sebagai berikut:

- a) *Minimarket*, kurang dari 400 m² (empat ratus meter persegi)
- b) *Supermarket*, 400 m² (empat ratus meter persegi) sampai 5000 m² (lima ribu meter persegi)
- c) *Hypermarket*, lebih dari 5000 m² (lima ribu meter persegi).

Perbedaan istilah *minimarket*, *supermarket* dan *hypermarket* selain di ukuran luas lantai penjualan adalah sebagai berikut:

- a) *Minimarket*

Minimarket adalah toko berukuran relatif kecil yang merupakan pengembangan dari Mom & Pop Store, dimana pengelolaannya lebih modern, dengan jenis barang dagangan lebih banyak. Mom & Pop Store adalah toko berukuran relatif kecil yang dikelola secara tradisional, umumnya hanya menjual bahan pokok/kebutuhan sehari-hari yang terletak di daerah perumahan/pemukiman, biasa dikenal sebagai toko kelontong (Tambunan dkk, 2004:4).

- b) *Supermarket*

Supermarket adalah toko ritel yang operasinya cukup besar, berbiaya rendah, margin rendah, volume penjualan tinggi, terkelompok berdasarkan lini produk, *self-service*, dirancang untuk memenuhi kebutuhan konsumen, seperti daging, hasil produk olahan, makanan kering, makanan basah, serta item-item produk non-food seperti mainan, majalah, toiletries dan sebagainya (Sopiah,2008). Dalam perkembangannya, format supermarket tidak terlalu banyak dipilih lagi. Sebab, dalam hal kedekatan lokasi dengan konsumen, supermarket kalah bersaing dengan *minimarket* (yang umumnya berlokasi di perumahan penduduk), sementara untuk range pilihan barang, supermarket tersaingi oleh *hypermarket* (yang menawarkan pilihan barang yang jauh lebih banyak) (Pandian,2009).

c) Hypermarket

Hypermarket merupakan toko ritel yang dijalankan dengan mengkombinasikan model *discount store*, supermarket dan *warehouse store* di satu tempat. Barang-barang yang ditawarkan meliputi produk grosiran, minuman, *hardware*, bahan bangunan, perlengkapan *automobile*, perabot rumah tangga, dan juga *furniture* (Sopiah,2008).

2.3 E-Commerce

Electronic Commerce (e-commerce) adalah proses pembelian, penjualan atau pertukaran produk, jasa dan informasi melalui jaringan komputer. *E-commerce* merupakan bagian dari *e-business*, di mana cakupan e-business lebih luas, tidak hanya sekedar perniagaan tetapi mencakup juga pengkolaborasi mitra bisnis, pelayanan nasabah, lowongan pekerjaan dan lain-lain. Selain teknologi jaringan *www*, *e-commerce* juga memerlukan teknologi basis data atau pangkalan data (*database*), e-surat atau surat elektronik (*e-mail*), dan bentuk teknologi non komputer yang lain seperti halnya sistem pengiriman barang, dan alat pembayaran untuk *e-commerce* ini (Siregar,2010).

E-commerce di definisikan sebagai proses pembelian dan penjualan produk, jasa, dan informasi yang dilakukan secara elektronik dengan memanfaatkan jaringan komputer. Salah satu jaringan yang digunakan adalah Internet (Hildamizanthi, 2011). Dalam mengimplementasikan *e-commerce* tersedia suatu integrasi rantai nilai dari infrastrukturnya, yang terdiri dari tiga lapis. Pertama infrastruktur sistem distribusi (*flow of good*); kedua, infrastruktur pembayaran (*flow of money*); dan ketiga, infrastruktur sistem informasi (*flow of information*). Agar dapat terintegrasinya sistem rantai suplai dan *supplier*, ke pabrik ke gudang distribusi jasa transportasi hingga ke pelanggan maka diperlukan integrasi *enterprise system* untuk menciptakan *supply chain visibility*. Ada tiga faktor yang faktor dicermati oleh kita jika ingin membangun toko *e-commerce* yaitu: *variability*, *visibility*, dan *velocity* (Sukamjati,2009).

E-commerce dapat diklasifikasikan berdasarkan beberapa aspek, berikut ini klasifikasi *e-commerce* berdasarkan pada sifat transaksinya yaitu (Turban, E, & King, D., 2002):

1) Business to Business (B2B)

Proses transaksi *e-commerce* bertipe B2B melibatkan perusahaan atau organisasi yang dapat bertindak sebagai pembeli atau penjual.

2) Business to Consumer (B2C)

Pada *e-commerce* bertipe B2C transaksi terjadi dalam skala kecil sehingga tidak hanya organisasi tetapi juga individu dapat terlibat pada pelaksanaan transaksi tersebut. Tipe *e-commerce* ini biasa disebut dengan e-tailing.

3) Business to Business to Consumer (B2B2C)

Pada *e-commerce* tipe ini, sebuah perusahaan menyediakan produk atau jasa kepada sebuah perusahaan lainnya. Perusahaan lain tersebut kemudian menyediakan produk atau jasa kepada individu yang bertindak sebagai konsumen.

4) Consumer to Business (C2B)

Pada *e-commerce* tipe ini, pihak individu menjual barang atau jasanya melalui Internet atau media elektronik lainnya kepada organisasi atau perusahaan yang berperan sebagai konsumen.

5) Consumer to Consumer (C2C)

Pada *e-commerce* tipe ini, konsumen menjual produk atau jasa yang dimilikinya secara langsung kepada konsumen lainnya.

6) Mobile Commerce (M-Commerce)

Mobile commerce merupakan salah satu tipe *e-commerce* dimana transaksi jual beli dan aktivitas bisnis yang terjadi dilakukan melalui media jaringan tanpa kabel.

7) Intrabusiness *E-commerce*

Aktivitas bisnis yang termasuk kedalam intrabusiness *e-commerce* diantaranya proses pertukaran barang, jasa, atau informasi antar unit dan individu yang terdapat pada suatu organisasi atau perusahaan.

8) Business to Employees (B2E)

B2E merupakan subset dari kategori intrabusiness *e-commerce* dimana perusahaan menyediakan pelayanan, informasi, atau produk pada individu pegawainya.

9) Collaborative Commerce

Saat individu atau grup melakukan komunikasi atau berkolaborasi secara *online*, maka dapat dikatakan bahwa mereka terlibat dalam collaborative commerce.

10) Non-business *E-commerce*

Non-business *e-commerce* merupakan *e-commerce* yang dilakukan pada organisasi yang tidak berorientasi untuk mendapatkan keuntungan seperti institusi akademis, organisasi sosial dsb.

11) E-government

E-government merupakan e-commerce yang dilakukan oleh pemerintah.

2.4 M-Commerce

M-commerce mengacu pada transaksi *E-commerce* yang dilakukan dalam lingkungan nirkabel, terutama melalui Internet. Seperti aplikasi *E-commerce* biasa, *M-commerce* dapat ditransaksikan melalui Internet, jalur komunikasi pribadi, *smart cards*, dan infrastruktur lainnya. *M-commerce* menciptakan peluang bisnis untuk memberikan layanan baru untuk pelanggan yang ada dan menarik pelanggan baru (Rainer & Cegielski, 2011).

Perangkat yang digunakan dalam *m-commerce* pada umumnya, perangkat *end user* yang digunakan pada proses *m-commerce* adalah sebagai berikut:

- a) *handphone*
- b) *smartphone*
- c) *PDA*
- d) *Laptop*
- e) *Earpiece (Personal Area Network)*

Setiap perangkat memiliki karakteristik yang berbeda-beda sehingga dapat mempengaruhi tingkat penggunaannya.

2.5 Unified Modeling Language(UML)

Unified Modeling Language(UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten L. Jeffery dkk, 2004). Sementara menurut Henderi (2007) Unified Modeling Language (UML) adalah sebuah bahasa pemodelan yang telah menjadi standar dalam industri software untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Bahasa Pemodelan UML lebih cocok untuk pembuatan perangkat lunak dalam bahasa pemrograman berorientasi objek (C++, Java, VB.NET), namun demikian tetap dapat digunakan pada bahasa pemrograman prosedural (Turck, 2007).

Unified Modeling Language (UML) biasa digunakan untuk (Henderi, 2007):

- a) Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b) Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c) Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagram*.
- d) Membuat model *behavior* "yang menggambarkan kebiasaan atau sifat sebuah sistem" dengan *state transition diagram*.
- e) Menyatakan arsitektur implementasi fisik menggunakan *component* dan *deployment diagram*.
- f) Menyampaikan atau memperluas fungsionalitas dengan *stereotypes* (Turck, 2007)

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam

mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi *diagram*. UML mempunyai banyak *diagram* yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. *Diagram-diagram* tersebut digunakan untuk (Henderi dkk, 2008):

- a) Mengkomunikasikan ide
- b) Melahirkan ide-ide baru dan peluang-peluang baru
- c) Menguji ide dan membuat prediksi
- d) Memahami struktur dan relasi-relasinya

Pemodelan menggunakan *Unified Modeling Language (UML)* merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan menggunakan UML merupakan pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan *diagram* yang dikelompokkan menjadi lima persektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah.

Berikut ini adalah penjelasan mengenai berbagai *diagram* UML serta tujuannya:

1. Model Use Case *Diagram*

Use Case Diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *use case diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dalam sistem itu. *Use Case* secara naratif digunakan secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi.

2. *Diagram* Struktur Statis

UML menawarkan dua *diagram* untuk memodelkan struktur statis sistem informasi yaitu:

- a) *Class diagram*: menggambarkan struktur object sistem. *Diagram* ini menunjukkan *class* object yang menyusun sistem dan juga hubungan anatara dua *class* objek tersebut
- b) *Object Diagram*: serupa dengan *class diagram*, tetapi object *diagram* memodelkan instance object actual dengan menunjukkan nilai-nilai saat ini dyri atribut instance. *Object Diagram* menyajikan "snapshot/potret" tentang objek sistem pada point waktu tertentu. *Diagram* ini tidak digunakan sesering *Class diagram*, tetapi saat digunakan dapat membantu seorang developer memahami struktur sistem secara lebih baik.

3. *Diagram* Interaksi

Diagram interaksi memodelkan sebuah interaksi, terdiri dari satu set objek, hubungan-hubungannya, dan pesan yang terkirim di antara objek. Model *diagram* ini memodelkan *behavior* (kelakuan) sistem yang dinamis dan UML memiliki dua *diagram* untuk tujuan ini, yaitu:

- a) *Sequence Diagram*: secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah use case atau operasi. *Diagram* ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi atau timing apa
- b) *Communication Diagram*: serupa dengan *diagram* rangkain/sekuensi, tetapi tidak fokus pada timing atau sekuensi pesan. *Diagram* ini justru menggambarkan interaksi (atau kolaborasi) antara objek dalam sebuah format jaringan.

Sequence diagram dan *communication diagram* merupakan isomorphic artinya kita dapat mengubah dari satu *diagram* ke *diagram* lain.

4. *State Diagram*

UML memiliki *diagram* untuk memodelkan *behavior* objek khusus yang kompleks (*statechart*) dan sebuah *diagram* untuk memodelkan *behavior* dari sebuah use case atau sebuah metode, yaitu:

- a) *Statechart Diagram*: digunakan untuk memodelkan *behavior* objek khusus yang dinamis. *Diagram* ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan *event* (kejadian) yang menyebabkan objek beralih dari satu *state* ke *state* lain.
- b) *Activity Diagram*: secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun use case. *Activity diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

5. *Diagram Implementasi*

Diagram implementasi juga memodelkan struktur sistem informasi, yaitu:

- a) *Component Diagram*: digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen software sistem. *Komponen diagram* dapat digunakan untuk menunjukkan bagaimana kode pemrograman dibagi menjadi modul-modul (atau komponen)
- b) *Diagram penguraian/Deployment Diagram*: digunakan untuk mendeskripsikan arsitektur fisik dalam istilah 'node' untuk *hardware* dan *software* dalam sistem. *Diagram* ini menggambarkan konfigurasi komponen-komponen *software real-time*, prosesor, dan peralatan yang membentuk arsitektur sistem.

Tidak semua jenis *diagram* di atas digunakan dalam perancangan aplikasi SM-Mart ini melainkan hanya lima saja yaitu: *Use Case Diagram*, *Activity Diagram*, *Communication Diagram*, *Class diagram*, dan *Sequence Diagram*.

1) Use Case Diagram

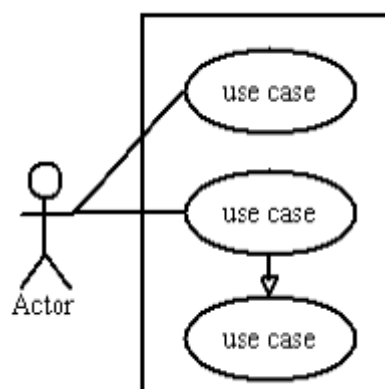
a) Definisi *Use Case Diagram*

Use Case diagram adalah model fungsional sebuah sistem yang menggunakan *actor* dan *use case*. *Use Case* adalah layanan (*services*) atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunanya (Henderi, 2008). *Use Case* adalah suatu pola atau gambaran yang menunjukkan kelakuan atau kebiasaan sistem. Setiap *Use Case* adalah suatu urutan (*sequence*) transaksi yang saling berhubungan dan dilakukan oleh sebuah *actor* dan sistem dalam bentuk sebuah dialog (Henderi, 2007). *Use Case Diagram* dibuat untuk memvisualisasikan/menggambarkan hubungan antara *Actor* dan *Use Case*. *Use Case Diagram* mempresentasikan kegunaan atau fungsi-fungsi sistem dari perspektif pengguna.

b) Simbol dan Notasi *Use Case Diagram*

1. Sistem

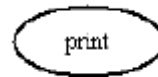
Gambar batasan (*boundaries*) sebuah sistem menggunakan persegi panjang yang berisi *use case*. Tempatkan *actor* yang terlibat pada setiap *use case* pada bagian luar *boundaries* sistem (gambar 1)



Gambar 1 Sistem

2. Use Case

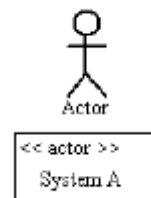
Gambar *use case* menggunakan lingkaran berbentuk bulat telur (oval). Beri nama oval tersebut dengan kata kerja yang menggambarkan fungsi-fungsi sistem (gambar 2)



Gambar 2 Use Case

3. Actor

Actor adalah para pengguna dari sebuah sistem. Kadangkala sebuah sistem adalah merupakan actor bagi sistem yang lain, beri nama actors sistem tersebut dengan setreotipe (bentuk klise/tiruan) actor (gambar 3). Actor adalah seseorang atau sesuatu yang harus berinteraksi dengan sistem atau sistem yang dibangun/dikembangkan.

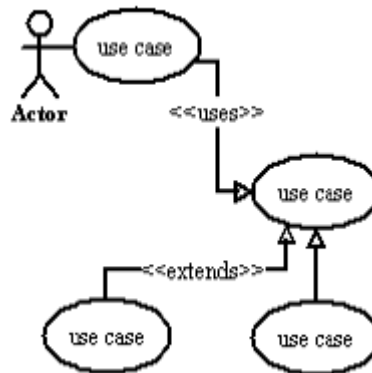


Gambar 3 Actor

4. Relationship

Relationship adalah ilustrasi atau gambaran relasi/hubungan antara sebuah actor dan use case dengan sebuah garis sederhana. Untuk relasi-relasi antara use cases, gunakan tanda anak panah-anak panah pada "uses" yang lainnya atau gunakan "extends". Suatu relasi "uses" mengindikasikan bahwa ada use case yang dibutuhkan oleh use case yang lain untuk melakukan sebuah permintaan (*task*). Sementara suatu relasi "extends"

mengindikasikan beberapa alternatif opsi (pilihan) tertentu pada tingkatan yang lebih bawah yang ada pada use case (gambar 4).



Gambar 4 Relationship

2) Activity Diagram

1. Definisi Activity Diagram

Activity Diagram adalah *diagram* yang menggambarkan sifat dinamis secara alamiah sebuah sistem dalam bentuk model aliran dan kontrol dari aktivitas ke aktivitas lainnya (Henderi, 2007). Sebuah aktivitas merepresentasikan suatu operasi pada beberapa *class* dalam sistem yang menghasilkan suatu perubahan keadaan (*state*) dari sistem tersebut. Secara khusus, *activity diagram* biasa digunakan untuk memodelkan *diagram* alir sebuah sistem kerja (*workflow*) atau proses bisnis (prosedur bisnis) dan operasi-operasi secara internal (Randy, 2008). Sementara menurut (Jeffery, 2004), *activity diagram* adalah sebuah *diagram* yang dapat digunakan untuk menggambarkan secara grafis aliran proses bisnis, langkah-langkah sebuah use case atau logika *behavior* (metode) object. Karena sebuah *activity diagram* adalah bentuk khusus dari *statechart diagram*, maka *activity diagram* kadangkala digunakan untuk memodelkan suatu kebiasaan sesuai dengan ketentuan/kaedah bisnis.

Untuk dapat membangun *activity diagram* yang baik, berikut proses yang unggul untuk membangun *activity diagram*:

1. Tambahkan poin awal dan akhir pada sebuah use case
2. Tambahkan sebuah kegiatan untuk tiap langkah utama pada use case(atau tiap langkah utama setiap pelaku yang menginisialisasi)
3. Tambahkan transisi dari setiap kegiatan ke kegiatan lain, poin keputusan, atau poin akhir
4. Tambahkan bar sinkronisasi di mana kegiatan dilakukan secara paralel.

2. Simbol dan Notasi Dasar Activity Diagram

1. Action States

Action state digambarkan dalam bentuk empat persegi panjang yang pada sudut-sudutnya melingkar (gambar 5)



Gambar 5 Action States

2. Action Flow

Action digambarkan dalam bentuk anak panah yang mengilustrasikan relasi antara action pada state (gambar 6)

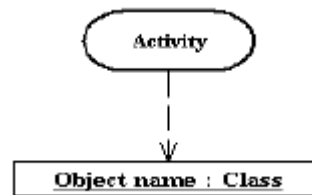


Gambar 6 Action Flow

3. Object Flow

Object flow menunjuk kepada kegiatan penciptakan (*create*) dan modifikasi oleh objek melalui activities. Sebuah tanda panah objek

flow dari suatu aksi kepada suatu objek berarti bahwa aksi tersebut menciptakan atau mempengaruhi objek tersebut. Sementara suatu tanda panah dari objek kepada suatu aksi mengindikasikan bahwa aksi state tersebut menggunakan objek (gambar 7)



Gambar 7 Object Flow

3) Communication Diagram

a) Denifisi Communication Diagram

Communication Diagram mendeskripsikan pola interaksi antar object. *Diagram* ini menunjukkan object-object yang terlibat dalam interaksi melalui *link* dan *message* yang dikirim antar object.

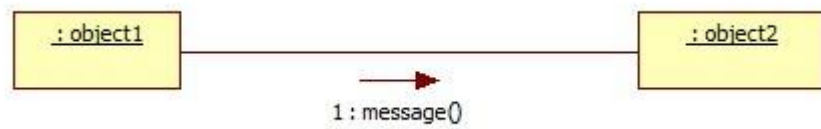
Communication diagram, macam dari *interaction diagram*, memberi tekanan pada hubungan-hubungan antar partisipan yang berbeda dalam sebuah interaksi. *Communication diagram* tidak menggambar setiap partisipan sebagai sebuah garis alir dan menunjukkan pesan-pesan dengan arah vertikal seperti yang dilakukan *sequence diagram*, tetapi memungkinkan untuk menempatkan partisipan-partisipan secara bebas, untuk menggambarkan hubungan-hubungan yang menunjukkan bagaimana partisipan berhubungan, dan menggunakan penomoran untuk menunjukkan bagian dari pesan-pesan.

Dalam UML 1.x, *diagram* ini disebut sebagai *collaboration diagram*.

b) Simbol dan Notasi Dasar Communication Diagram

Simbol dan notasi yang digunakan pada *communication diagram* antara lain:

1. Object dapat direpresentasikan dengan cara sebagai berikut:
 - a. NamaObject>NamaClass
 - b. NamaObject
 - c. :NamaClass
2. *Link* adalah hubungan antar object yang dapat digunakan untuk mengirim *message*. *Link* digambarkan dengan garis utuh antar dua objek.
3. *Message* : pengertiannya sama dengan *message* pada *sequence diagram*.



Gambar 8 Communication Diagram

4) Class Diagram

a) Definisi Class Diagram

Class adalah kumpulan objek-objek yang mempunyai struktur, *behavior*, relasi, dan semantic/kata yang serupa. *Class-class* ditentukan/ditemukan dengan cara memeriksa objek-objek dalam *sequence diagram* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. *Class* berikutnya diberi nama menggunakan kata benda sesuai dengan domain/bagian/kelompoknya (Jeffery dkk, 2004).

Class diagram adalah *diagram* yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Karena itu, *class diagram* merupakan tulang punggung bagi hampir setiap metode

berorientasi objek termasuk UML (Henderi, 2008). Sementara menurut (Jeffery dkk,2004) *class diagram* adalah gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukkan *class-class* objek yang menyusun sebuah sistem dan juga hubungan antara *class* objek tersebut.

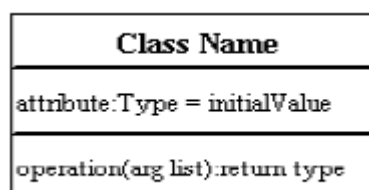
Elemen-elemen class diagram dalam pemodelan UML terdiri dari: *Class-class*, struktur *class*, sifat *class* (*class behavior*), perkumpulan/gabungan (*association*), pengumpulan/kesatuan (*agregation*), ketergantungan (*dependency*), relasi-relasi turunannya, keberagaman dan indikator navigasi, dan role name (peranan/tugas nama).

b) Simbol dan Notasi Class Diagram

Classes merepresentasikan sebuah abstraksi dari entitas-entitas dengan sifat-sifat atau karakteristik yang bersifat umum. Asosiasi merepresentasikan (hubungan) antara *classes*.

1. Classes

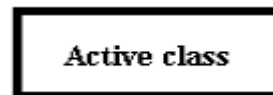
Class digambarkan dengan bentuk empat persegi panjang yang dibagi kedalam ruang-ruang terpisah (*compartments*). Nama *class* ditempatkan pada bagian pertama (rata tengah, dicetak tebal, dan diawali huruf besar), daftar atribut diletakan pada bagian kedua, dan tuliskan operasi-operasi pada *class* dibagian ketiga (gambar 9).



Gambar 9 Classes

2. Active Class

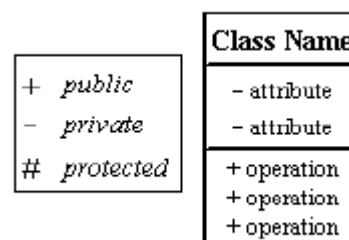
Active Class adalah *class* yang memulai dan mengontrol aliran/arus/arah aktifitas. Sementara *passive class* menyimpan data dan melayani *class* yang lain. Gambarkan *active class* pada sebuah folder tebal dibagian tengah (gambar 10).



Gambar 10 Active Class

3. Visibility

Gunakan penilai visibility untuk menandakan siapa yang bisa mengakses informasi-informasi yang diisi kedalam sebuah *class*. Private visibility artinya informasi yang ada dalam sebuah *class* disembunyikan/dipartisi dari pihak luar. Public visibility mengizinkan semua *class* yang lainnya untuk melihat nilai informasi. Protected visibility mengizinkan *class-class* yang ada yang merupakan turunannya untuk mengakses informasi yang ada didalamnya karena mereka merupakan *class* turunan dari *class* induknya (gambar 11).

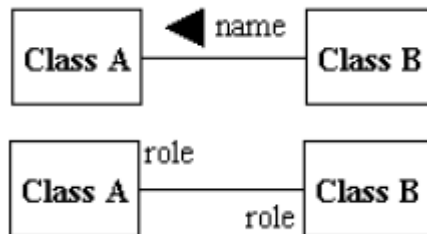


Gambar 11 Visibility

4. Asosiasi

Asosiasi adalah representasi/gambaran relasi statis diantara *class-class*. Tempatkan nama asosiasi pada bagian atas atau di bawah garis asosiasi. Aturan merepresentasikan arah bagi

kedua kelas untuk saling berhubungan satu sama lain (gambar 12).



Gambar 12 Asosiasi

Dalam asosiasi terdapat informasi yang disebut dengan multiplicity. Multiplicity adalah jumlah kejadian minimum dan maksimum dari satu object/class untuk satu kejadian tunggal dari object/class yang terkait (Jeffery dkk, 2004).

5) Sequence Diagram

a) Definisi Sequence Diagram

Sequence *Diagram* adalah suatu *diagram* yang memperlihatkan/menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa "pesan/*message*" (Henderi, 2008). Sequence *diagram* dapat juga didefinisikan sebagai suatu *diagram* yang menggambarkan interaksi-interaksi yang ada antar *class* dalam suatu hubungan perubahan dengan sebuah pesan pada akhir waktu (Randy, 2008). Sequence *diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan output tertentu. Sequence *diagram* diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan (Henderi, 2007).

b) Simbol dan Notasi Sequence Diagram

1. Actor

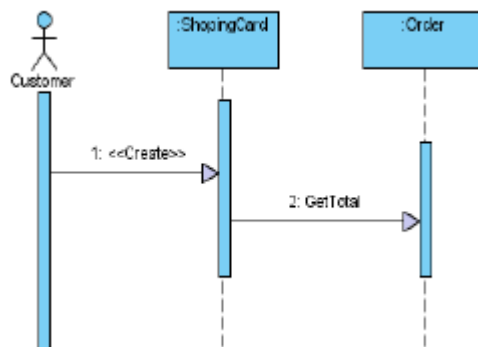
Actor merupakan pelaku yang berhubungan langsung dengan sistem(gambar 13)



Gambar 13 Actor

2. Activation

Activation boxes merepresentasikan waktu yang dibutuhkan oleh sebuah objek untuk melaksanakan sebuah tugas/perintah secara lengkap (gambar 14).

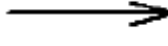
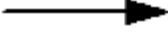
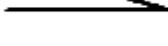
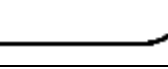



Gambar 14 Activation

3. Messages

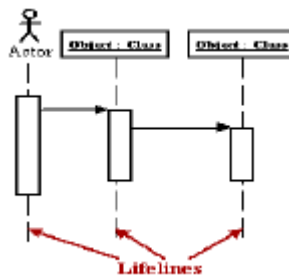
Message adalah anak panah yang merepresentasikan komunikasi antara objek. *Message* berguna untuk mengirimkan perintah kepada LifeLine. *Message* bisa berasal dari actor kepada LifeLine atau dari LifeLine kepada LifeLine yang lain. Gunakan setengah garis anak panah untuk merepresentasikan pesan-pesan asynchronous. Pesan asynchronous dikirim dari sebuah objek yang tidak akan menunggu respon dari penerima sebelum melanjutkan perintahnya. Beberapa tipe *message* untuk *sequence diagram* dan *communication diagram* digambarkan pada tabel(tabel 3).

Tabel 3 Messages

Arrow	Message Type
	Simple
	Synchronous
	Asynchronous
	Balking
	Time out

4. Lifeline

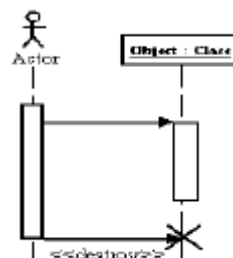
Lifeline adalah tanda garis pisah yang mengindikasikan kehadiran objek pada saat terakhir/akhir waktu (gambar 15).



Gambar 15 Lifeline

5. Destroying Object

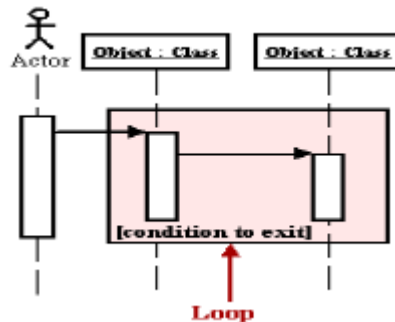
Objek dapat di-akhiri secara lebih cepat menggunakan sebuah garis anak panah yang diberi label ”<<destroy>>” dan diujungnya diberi sebuah label X (gambar 16).



Gambar 16 Destroying Object

6. Loop

Suatu pengulangan atau loop dalam sebuah sequence *diagram* digambarkan seperti sebuah empat persegi panjang. Letakkan kondisi untuk keluar pada pengulangan di bagian kiri bawah di bawah kurang empat persegi panjang (gambar 17)



Gambar 17 Loop

2.6 Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java dikembangkan pada bulan Agustus 1991, dengan nama semula Oak. Pada Januari 1995, karena nama Oak dianggap kurang komersial, maka diganti menjadi Java. Pada Desember 1998, Sun memperkenalkan nama "Java 2" (J2) sebagai generasi kedua dari Java platform. Konvensi nama baru ini diterapkan untuk semua edisi Java yaitu Standard Edition (J2SE), Enterprise Edition (J2EE), dan Micro Edition (J2ME). Ada tiga platform Java yang telah didefinisikan, yang masing-masing diarahkan untuk tujuan tertentu dan untuk lingkungan komputasi yang berbeda-beda:

- 1) *Standard Edition* (J2SE): J2SE merupakan inti dari bahasa pemrograman Java. J2SE Didesain untuk dijalankan pada komputer desktop dan komputer workstations.
- 2) *Enterprise Edition* (J2EE): Dengan built-in yang mendukung untuk servlets, JSP, dan XML. Edisi ini ditujukan untuk aplikasi berbasis *server*.
- 3) *Micro Edition* (J2ME): Didesain untuk piranti dengan memori terbatas, layar display terbatas dan power pemrosesan yang juga terbatas.

2.7 Android

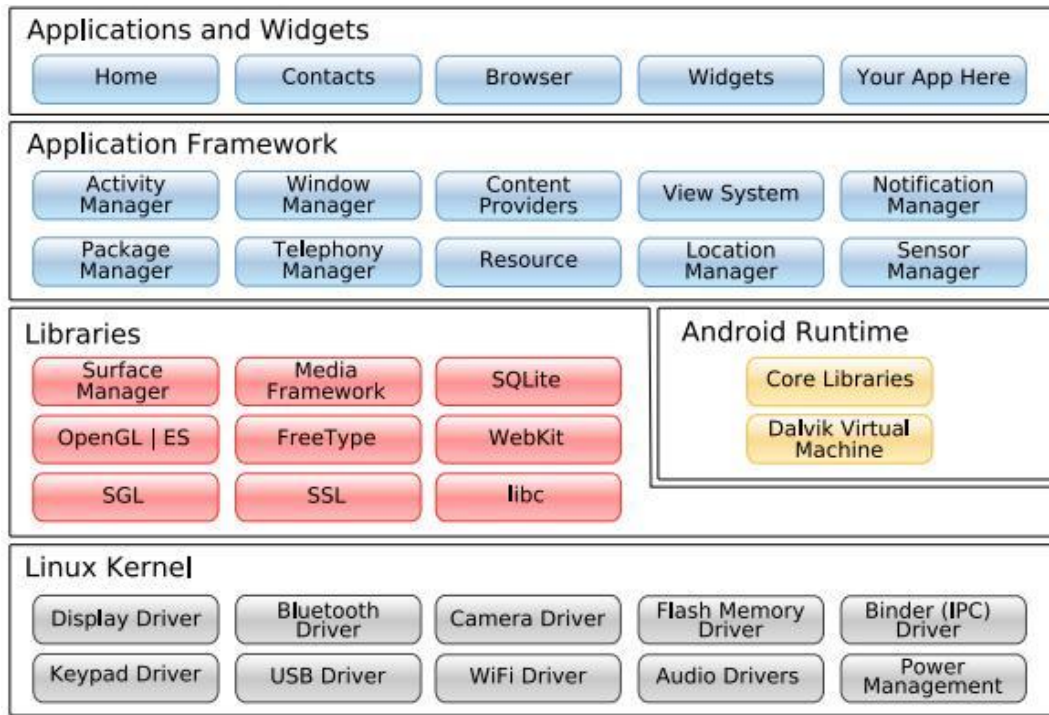
Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Android merupakan generasi *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan dibawah GNU, General Public Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah “copyleft” lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah terms. Android didistribusikan di bawah Lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen perangkat khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka ke masyarakat *open source*. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apa pun yang mereka inginkan. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apa pun yang mereka inginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis Android. Sebagian besar pengembang menggunakan *Eclipse* yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android. *Eclipse* adalah IDE yang paling populer untuk pengembangan Android, karena memiliki Android *plug-in* yang tersedia untuk memfasilitasi pengembangan Android. Selain itu, *Eclipse* juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembang aplikasi Android, ini terbukti dengan adanya penambahan *plugins* untuk *Eclipse* untuk membuat *project* Android di mana *source software* langsung dari situs resminya Google. Akan tetapi, hal di atas tidak menutup kemungkinan untuk menggunakan IDE yang lain seperti Netbeans untuk melakukan pengembangan Android (Safaat, 2012).

a) Arsitektur Android

Arsitektur Android dapat dilihat di gambar 18



Gambar 18 Arsitektur android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut:

1. *Application dan Widgets*

Application dan Widgets ini adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita unduh aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Application Frameworks*

Android adalah “Open Development Platform” yaitu Android menawarkan kepada pengembang atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembangan bebas untuk mengakses perangkat keras,

akses informasi *resource*, menjalankan *service background*, mengatur *alarm*, dan menambah status *notification*, dan sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

Dengan demikian bisa kita simpulkan *Applications Frameworks* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa SMS dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut:

- a) *Views*
- b) *Content Provider*
- c) *Resource Manager*
- d) *Notification Manager*
- e) *Activity Manager*

3. *Libraries*

Libraries ini adalah *layer* di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. *Libraries* berjalan di atas kernel. *Layer* ini meliputi berbagai *library C/C++* inti seperti *Libc* dan *SSL*, serta:

- a) *Libraries media* untuk pemutaran media audio dan video
- b) *Libraries* untuk manajemen tampilan
- c) *Libraries Graphics* mencakup *SGL* dan *OpenGL* untuk grafis 2D dan 3D
- d) *Libraries SQLite* untuk dukungan *database*

- e) *Libraries* SSL dan *Webkit* terintegrasi dengan *web browser* dengan *security*
- f) *Libraries* LiveWebcore mencakup modern *web browser* dengan *engine embedded web view*
- g) *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's

4. *Run Time*

Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine* (*DVM*) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android *Run Time* dibagi menjadi dua bagian yaitu:

- a) *Core Libraries*: Aplikasi Android dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan Virtual Machine Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa Java/c yang ditangani oleh *Core Libraries*.
- b) *Dalvik Virtual Machine*: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat Linux kernel untuk melakukan *thearding* dan manajemen tingkat rendah.

5. *Linux Kernel*

Linux Kernel adalah layer di mana inti dari sistem operasi dari Android berada. Linux kernel berisi *file-file system* yang mengatur sistem *processing, memory, resource, drivers*, dan sistem-sistem operasi Android lainnya. *Linux kernel* yang digunakan Android adalah Linux kernel *release 2.6* (Safaat: 2012).

b) Android SDK (Software Development Kit)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral, Android memberi pengembang kesempatan untuk membuat aplikasi yang mereka butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur-fitur Android yang paling penting adalah:

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile*.
3. *Integrated browser* berdasarkan *engine open source Webkit*.
4. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *opengl ES 1,0* (akselerasi *hardware* opsional).
5. *SQLite* untuk penyimpanan data.
6. *Media support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), *GSM Telephony* (tergantung *hardware*).
7. *Bluetooth* , *EDGE*, *3G*, dan *Wifi* (tergantung *hardware*).
8. Kamera, *GPS*, kompas, dan *accelerometer* (tergantung *hardware*).
9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori , dan *plugin* untuk IDE *Eclipse* (Safaat, 2012).

c) ADT (Android Development Tools)

ADT (*Android Development Tools*) adalah plugin yang didesain untuk IDE *Eclipse* yang memberikan kita kemudahan dalam mengembangkan

aplikasi Android dengan menggunakan IDE *Eclipse*. Dengan menggunakan ADT untuk *Eclipse* akan memudahkan kita dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya. Begitu juga kita dapat menjalankan aplikasi menggunakan Android SDK menggunakan *Eclipse*. Dengan ADT juga kita dapat melakukan pembuatan *package* Android (.apk) yang digunakan untuk distribusi aplikasi Android yang kita rancang.

Mengembangkan aplikasi Android dengan menggunakan ADT di *Eclipse* sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi Android. Berikut adalah versi ADT untuk *Eclipse* yang sudah dirilis:

1. ADT 12.0.0 (Juli 2011)
2. ADT 11.0.0 (Juni 2011)
3. ADT 10.0.1 (Maret 2011)
4. ADT 10.0.0 (Februari 2011)
5. ADT 9.0.0 (Januari 2011)
6. ADT 8.0.1 (Desember 2010)
7. ADT 8.0.0 (Desember 2010)
8. ADT 0.9.9 (September 2010)
9. ADT 0.9.8 (September 2010)
10. ADT 0.9.7 (Mei 2010)
11. ADT 0.9.6 (Maret 2010)
12. ADT 0.9.5 (Desember 2010)
13. ADT 0.9.4 (Oktober 2010)

Semakin tinggi *platform* Android yang digunakan , dianjurkan menggunakan ADT yang terbaru, karena biasanya munculnya *platform* baru diikuti oleh munculnya versi ADT yang terbaru. Proses instalasi ADT di *Eclipse* dapat dilakukan secara *on-line* maupun *offline* (Safaat, 2012).

d) Versi Android

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2010 diperkirakan hampir semua *vendor* seluler di dunia menggunakan Android sebagai sistem operasi. Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut:

1. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1 Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, meng-*upload video* ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth*, animasi layar, dan *keyboard* pada layar yang disesuaikan dengan sistem.

3. Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan indicator baterai dan *control applet* VPN. Fitur lainnya adalah gallery yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder* dan galeri yang diintegrasikan, CDMA/EVDO, 802.1x, VPN, Gestures, dan *Text-to-Speech engine*, kemampuan *dial* kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel,) pengadaan resolusi VWGA.

4. Android versi 2.0/2.1 (Éclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan antarmuka dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan *Bluetooth* 2.1.

5. Android versi 2.2 (Froyo: Frozen Yoghurt)

Pada bulan Mei 2010 Android versi 2.2 Rev 1 diluncurkan. Android inilah yang sekarang sangat banyak beredar di pasaran, salah satunya adalah dipakai di Samsung FX tab. Fitur yang tersedia di Android versi ini sudah kompleks di antaranya adalah:

- a) Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- b) *Dalvik Virtual Machine* dioptimalkan untuk perangkat mobile.
- c) Grafik: grafik di 2D dan grafis 3D berdasarkan *libraries* OpenGL.
- d) SQLite: untuk penyimpanan data.
- e) Mendukung media: *audio*, *video*, dan berbagai format gambar(MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- f) GSM, *Bluetooth*, EDGE, 3G, dan Wifi (*hardware independent*)
- g) Kamera, *Global Positioning System* (GPS), kompas, dan *accelerometer* (tergantung *hardware*).

6. Android versi 2.3 (Gingerbread)

Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan seperti berikut:

- a) *SIP-based VoIP*
- b) *Near Field Communications*(NFC)
- c) *Gyroscope* dan sensor

- d) *Multiple cameras support*
 - e) *Mixable audio effects*
 - f) *Download manager*
7. Android versi 3.0 (Honeycomb (API Level 11)) dirilis Februari 2011 sebagai Android 3.0 revisi 1 serta Android versi 3.0 revision 2 telah dirilis pada Juli 2011.
 8. Android versi 3.1 (Honeycomb (API Level 12)) dirilis Mei 2011, sedangkan Android 3.1 revisi 2 juga dirilis Mei 2011, serta Android 3.1 revision 3 dirilis pada Juli 2011.
 9. Android versi 3.2 (Honeycomb (API Level 13)) dirilis Juli 2011
 10. Android versi 4.0 (Ice Cream Sandwich) dirilis November 2011

Android versi 3.0 ke atas adalah generasi *platform* yang digunakan untuk tablet PC. Sementara versi 4.0 sudah merupakan *platform* yang bisa dipakai di *smartphone* dan tablet PC (Safaat, 2012).

2.8 MySQL

Ada banyak *server* database yang populer dan banyak digunakan orang adalah MySQL. Selain karena bisa didapat dengan mudah dan gratis, MySQL juga dapat berjalan pada sistem operasi apa pun. Berikut ini beberapa kelebihan yang dimiliki MySQL :

- a) Program database MySQL memang merupakan pasangan setia yang sering diintegrasikan dengan Apache.
- b) MySQL memiliki kemampuan *server multithread* sehingga dapat menangani banyak permintaan (*request*) layanan secara bersamaan.
- c) Dapat berjalan pada banyak sistem operasi (Linux, Windows, SunOS, MacOS).
- d) Memiliki metode enkripsi yang baik.
- e) Mampu menanganai data yang besar dan banyak.
- f) Menggunakan autentikasi pengguna (*user*) dan kata sandi (*password*) sehingga menjadikannya lebih aman.

2.9 PHP

PHP merupakan *scripting server side* yang paling populer dan banyak digunakan. Pada awalnya PHP memiliki singkatan *Personal Home Page tool* yang pertama kali dibuat dan dilengkapi oleh banyak pihak sehingga mengalami perkembangan yang sangat pesat. Singkatannya juga telah berubah menjadi singkat berulang, yakni PHP *Hypertext Preprocessor*.

Dengan adanya PHP, dunia situs Web menjadi lebih menarik dan interaktif karena dengan menggunakan PHP, para pengunjung bisa saling berkomunikasi satu sama lain. Di bawah ini beberapa keunggulan yang dimiliki oleh PHP :

- a) Mendukung penggunaan *cookie*.
- b) Memiliki fitur dalam menangani *session*.
- c) Mendukung integrasi dengan database.
- d) Mendukung *regular expression*.
- e) Penanganan kesalahan yang dibuat secara bertingkat.

2.10 CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman.

Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan

dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Keuntungan menggunakan CSS:

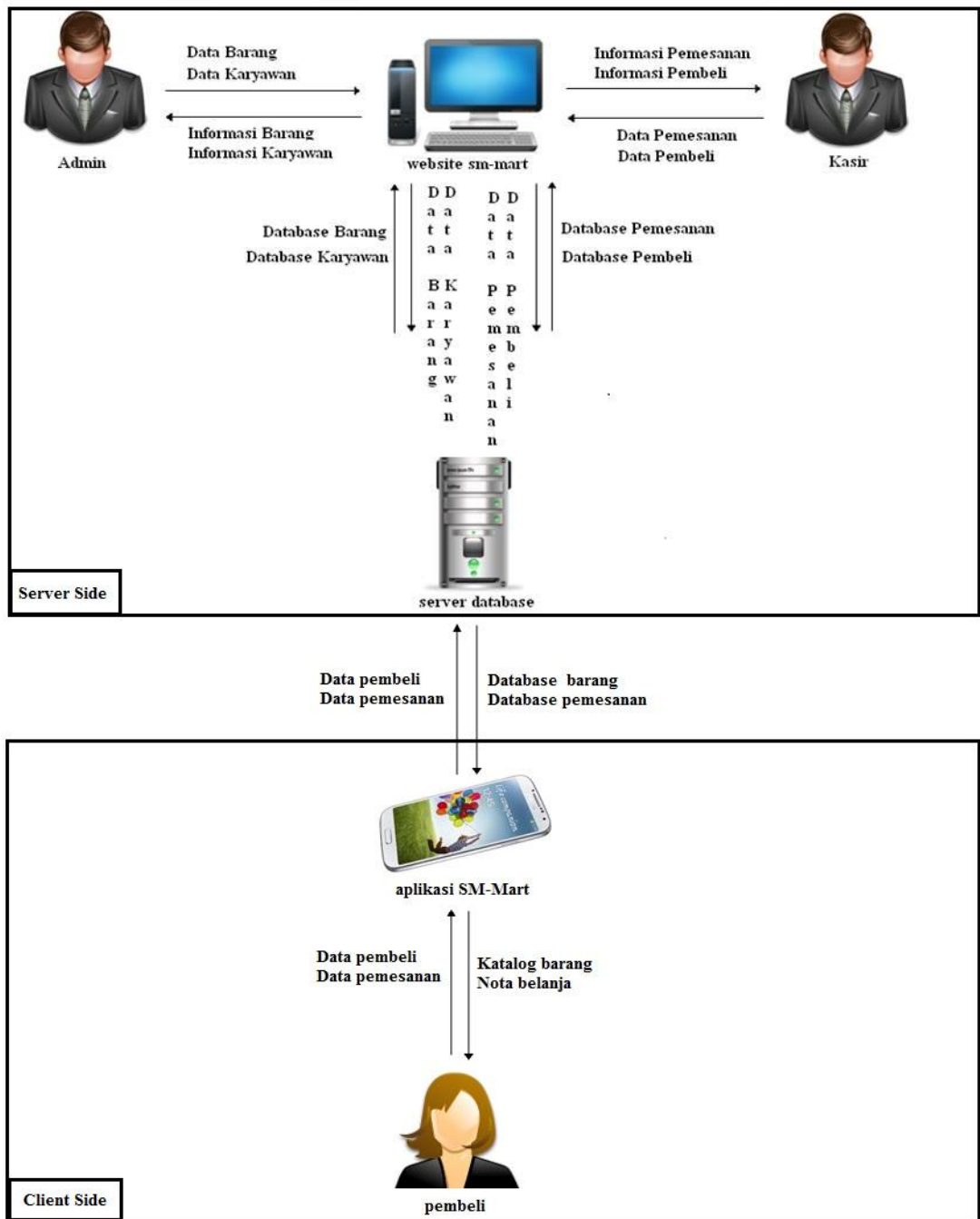
1. CSS memberikan keseragaman pada halaman web.
2. Dengan CSS dapat menghemat banyak waktu dan pekerjaan berulang.
3. Saat menggunakan CSS, perubahan tidak perlu dilakukan dalam setiap halaman web. Anda hanya perlu membuat perubahan dalam *style sheet*.
4. CSS memungkinkan Anda untuk memuat halaman web Anda dengan mudah.
5. *Layers* (Lapisan), seperti *item pop-up*, dapat digunakan dalam dokumen.
6. CSS membantu Anda memelihara halaman web Anda dengan mudah dan efektif.

BAB III ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

3.1.1 Deskripsi Umum Sistem

Deskripsi umum sistem aplikasi *client* SM-Mart dijelaskan pada gambar 19.



Gambar 19 Deskripsi Umum Sistem

Dalam laporan ini, penulis hanya akan menangani *client* side dari aplikasi SM-Mart yang dapat dilihat pada gambar di atas.

3.1.2 Kebutuhan Fungsional

- F-001 Aplikasi dapat menangani login dan logout untuk memastikan bahwa hanya pengguna yang sudah terdaftar saja yang dapat mengakses aplikasi.
- F-002 Aplikasi dapat menampilkan kategori-kategori barang yang dapat dipesan.
- F-003 Aplikasi dapat menampilkan katalog barang berdasarkan kategori barang yang dipilih.
- F-004 Aplikasi dapat menampilkan informasi barang di katalog barang seperti nama barang, harga, satuan, dan stok.
- F-005 Pembeli dapat melakukan pemesanan barang dengan cara memilih barang dari katalog dan menentukan kuantitasnya.
- F-006 Pembeli dapat menghapus suatu barang yang tidak jadi dipesan pada keranjang belanjanya sebelum dilakukan konfirmasi pemesanan.
- F-007 Aplikasi dapat menghitung total harga secara otomatis.
- F-008 Pembeli dapat memilih jenis pengantaran barang yang diinginkan.
- F-009 Pembeli dapat mengubah password.

3.1.3 Kebutuhan Fungsional

- NF-001 Aplikasi menggunakan Bahasa Indonesia

3.1.4 Batasan Sistem

Aplikasi ini memiliki batasan sistem untuk dapat dijalankan, yang dapat dilihat pada tabel 3.

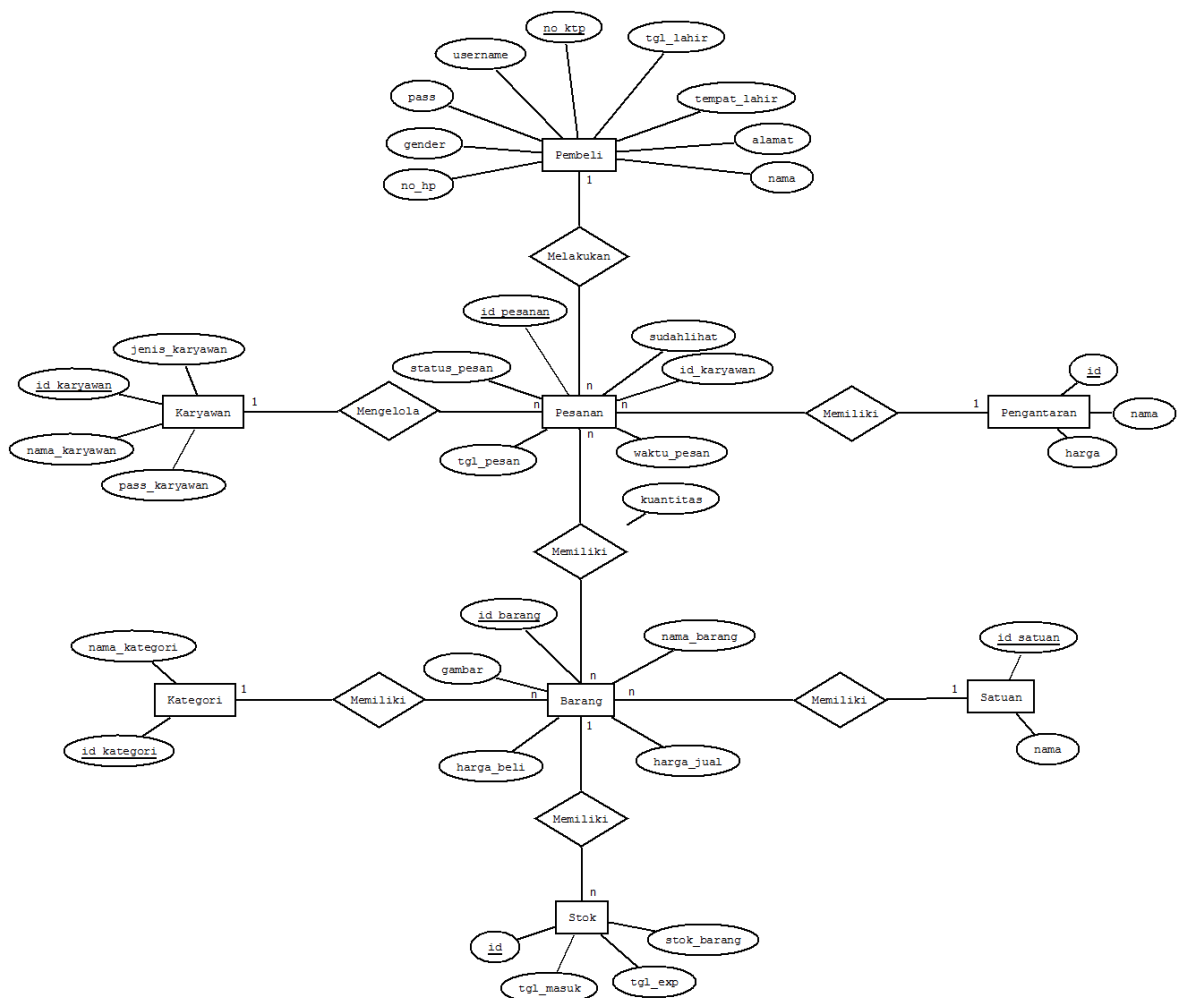
Tabel 4 Batasan Sistem

Perangkat	Spesifikasi
Mobile Phone	a) Terkoneksi jaringan Internet b) Sistem operasi Android c) Versi Android minimal 2.2 (Froyo)

3.2 Perancangan Data

3.2.1 Entity Relationship Diagram

Skema relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Gambar 20 merupakan penjelasan rangkaian basis data pada aplikasi SM-Mart yang akan dibangun:



Gambar 20 Entity Relationship Diagram

3.2.2 Struktur Tabel

Struktur tabel merupakan memaparkan rancangan isi tabel-tabel yang akan digunakan dalam aplikasi. Struktur tabel dalam aplikasi SM-Mart ini adalah sebagai berikut:

Tabel 5 Struktur Tabel t_pembeli

Nama Field	Tipe	Panjang	Kunci	Atribut
no_ktp	Varchar	20	PK	Not Null
username	Varchar	20		Not Null
pass	Varchar	20		Not Null
gender	Enum('L', 'P')			Not Null
tgl_lahir	Date			Not Null
tempat_lahir	Varchar	50		Not Null
alamat	Varchar	50		Not Null
no_hp	Varchar	12		Not Null
nama	Varchar	50		Not Null

Tabel t_pembeli digunakan untuk menyimpan data pembeli yang terdaftar. Data yang disimpan pada tabel ini meliputi username dan pass yang digunakan untuk proses login dalam aplikasi. Untuk keperluan tersebut, username masing-masing pembeli tidak boleh sama. Data lain seperti no_ktp, gender, tgl_lahir, tempat_lahir, alamat, no_hp, dan nama disimpan untuk mengetahui profil pembeli serta informasi kontakannya.

Tabel 6 Struktur Tabel t_pesanan

Nama Field	Tipe	Panjang	Kunci	Atribut
id_pesanan	Varchar	50	PK	Not Null
username	Varchar	20	FK	Not Null
id_karyawan	Int	10	FK	Not Null
tgl_pesan	Date			Not Null
waktu_pesan	Time			Not Null
status_pesan	Enum('DITERIMA', 'DIPROSES', 'SELESAI', 'SUDAHHANTAR', 'SUDAHJEMPUT')			Not Null
jenis_pengantaran	Enum('ANTAR', 'JEMPUT')			Not Null
sudahlihat	Enum('y', 'n')			Not Null

Tabel t_pesanan digunakan untuk menyimpan data pesanan masing-masing pembeli. Data yang disimpan meliputi id_pesanan yang digunakan untuk

mengidentifikasi data pesanan masing-masing pembeli. Dalam hal ini, data id_pesanan tidak boleh sama. Sedangkan data username dan id_karyawan disimpan untuk mengidentifikasi pembeli yang melakukan pesanan dan karyawan yang menangani pesanan tersebut. Data lainnya seperti data tgl_pesanan dan waktu_pesanan digunakan untuk mengetahui informasi tanggal dan waktu pesanan. Data status_pesanan disimpan untuk mengetahui status pesanan yang berisi keterangan “DITERIMA”, “DIPROSES”, “SELESAI”, “SUDAHHANTAR”, “SUDAHJEMPUT”. Selain itu, data lain yaitu jenis_pengantaran disimpan untuk mengetahui jenis pengantaran yang berupa “ANTAR” atau “JEMPUT”. Data ini akan digunakan pada proses konfirmasi pengantaran. Data sudahlihat digunakan untuk menampilkan jumlah pesanan yang belum dilihat.

Tabel 7 Struktur Tabel t_barang

Nama Field	Tipe	Panjang	Kunci	Atribut
id_barang	Varchar	50	PK	Not Null
nama_barang	Varchar	50		Not Null
gambar	Varchar	50		Not Null
id_satuan	int	10	FK	Not Null
id_kategori	int	10	FK	Not null
harga_jual	Int	50		Not null
harga_beli	Int	50		Not Null

Tabel t_barang digunakan untuk menyimpan data barang. Data id_barang disimpan untuk mengidentifikasi barang yang satu dengan yang lainnya. Untuk keperluan ini, data id_barang tidak boleh sama. Data id_kategori disimpan untuk digunakan dalam proses menampilkan katalog barang yang akan disesuaikan dengan kategori dari masing-masing barang. Data id_satuan disimpan untuk menyimpan data satuan dari tabel t_satuan. Data nama_barang, gambar, dan harga_jual disimpan untuk informasi barang yang akan digunakan dalam proses menampilkan katalog dan proses pemesanan barang. Data lain yaitu harga_beli disimpan untuk keperluan laporan keuangan.

Tabel 8 Struktur Tabel t_kategori

Nama Field	Tipe	Panjang	Kunci	Atribut
id_kategori	Int	2	PK	Not Null
nama_kategori	Varchar	20		Not Null

Tabel t_kategori digunakan untuk menyimpan data kategori barang. Tabel ini berisi data id_kategori yang disimpan untuk digunakan proses menampilkan kategori dan katalog barang. Data id_kategori tidak boleh sama. Data lain berupa data nama_kategori disimpan untuk informasi kategori barang yang digunakan dalam proses menampilkan kategori barang.

Tabel 9 Struktur Tabel t_karyawan

Nama Field	Tipe	Panjang	Kunci	Atribut
id_karyawan	Int	10	PK	Not null
nama_karyawan	Varchar	50		Not Null
pass_karyawan	Varchar	20		Not Null
jenis_karyawan	Enum('admin', 'kasir')			Not null

Tabel t_karyawan digunakan untuk menyimpan informasi karyawan. Data id_karyawan dan pass_karyawan disimpan untuk proses login karyawan. Dalam hal ini, id_karyawan tidak boleh sama. Data nama_karyawan disimpan untuk mengetahui profil karyawan. Data lainnya yaitu jenis_karyawan disimpan untuk keperluan pemberian hak akses terhadap karyawan yang terdiri dari dua jenis yaitu admin dan kasir.

Tabel 10 Struktur Tabel t_detail_pesanan

Nama Field	Tipe	Panjang	Kunci	Atribut
id_pesanan	Int	8	FK	Not null
id_barang	Varchar	50	FK	Not Null
kuantitas	Int	10		Not Null
harga_jual	int	50	FK	

Tabel t_detail_pesanan digunakan untuk menyimpan data detail pesanan. Data yang disimpan berupa id_pesanan dan id_barang yang digunakan untuk mengambil data pesanan dan data barang dari suatu data pemesanan. Data kuantitas dan harga_jual disimpan untuk proses penghitungan total harga.

Tabel 11 Struktur Tabel t_pengantaran

Nama Field	Tipe	Panjang	Kunci	Atribut
id	Int	10	PK	Not null
nama	Varchar	20		Not Null
harga	Int	50		Not Null

Tabel t_pengantaran digunakan untuk menyimpan data pengantaran. Data id disimpan untuk proses menentukan jenis pengantaran. Data nama disimpan sebagai informasi data pengantaran. Data harga disimpan untuk menentukan biaya pengantaran sesuai dengan jenisnya.

Tabel 12 Struktur Tabel t_satuan

Nama Field	Tipe	Panjang	Kunci	Atribut
id_satuan	Int	10	PK	Not null
nama	Varchar	20		Not Null

Tabel t_satuan digunakan untuk menyimpan informasi satuan barang. Data id_satuan disimpan untuk proses menampilkan data satuan barang. Data nama disimpan untuk informasi satuan barang.

Tabel 13 Struktur Tabel t_stok

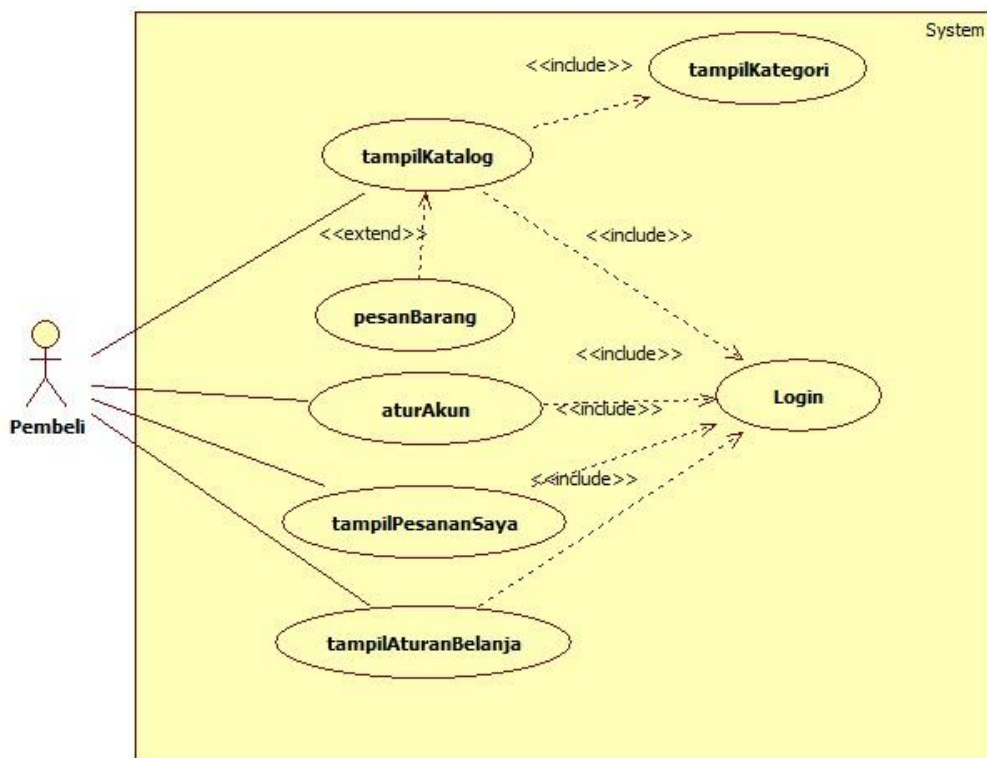
Nama Field	Tipe	Panjang	Kunci	Atribut
id	Int	10	PK	Not null
id_barang	Varchar	50	FK	Not Null
tgl_masuk	Date			Not Null
tgl_exp	Date			Not Null
stok_barang	Int	10		Not Null
id_kategori	Int	10	FK	Not Null

Tabel t_stok digunakan untuk menyimpan informasi stok. Data id disimpan untuk proses menampilkan informasi stok. Data id_barang disimpan untuk menampilkan data barang. Data tgl_masuk, tgl_exp dan stok_barang disimpan sebagai informasi stok barang. Data id_kategori disimpan untuk menampilkan data kategori.

3.3 Perancangan Data

3.3.1 Use Case Diagram

Use Case *Diagram* Aplikasi SM-Mart digunakan untuk menggambarkan hubungan external actor dengan Use Case yang terdapat dalam Aplikasi SM-Mart. Use Case *Diagram* Aplikasi SM-Mart dapat dilihat pada Gambar 21.



Gambar 21 Use Case Diagram

Secara umum, aplikasi ini memiliki satu actor yaitu Pembeli. Sistem dapat menampilkan katalog, namun sebelumnya Pembeli harus melakukan login. Setelah login, pembeli harus memilih kategori barang. Kemudian sistem akan menampilkan katalog barang sesuai dengan kategori yang dipilih oleh pembeli. Setelah itu, barulah pembeli dapat memesan barang. Selain itu, pembeli juga dapat mengatur informasi akunya berupa password, alamat, no. hp dan foto.

3.3.2 Skenario Use Case

Dari gambar 22 dapat dijelaskan bahwa *user* (pembeli) dapat melakukan login, tampilKatalog, tampilKategori, pesanBarang, aturAkun, tampilPesananSaya, tampilAturanBelanja. Skenario Use Case tersebut adalah:

1. Skenario Use Case login

Aktor	Pembeli
Kondisi Awal	Pembeli belum melakukan login dan halaman login ditampilkan
Kondisi Akhir	Pembeli masuk ke sistem dan halaman kategori barang ditampilkan
Skenario	1. Memasukkan <i>username</i> dan password

	2. Verifikasi oleh sistem.
	3. [login sukses] Halaman kategori barang ditampilkan
Skenario Alternatif	3.a. [login gagal] Muncul pesan kesalahan dan halaman login akan ditampilkan kembali.

2. Skenario Use Case tampilKategori

Aktor	Pembeli
Kondisi Awal	Pembeli telah melakukan login
Kondisi Akhir	Halaman katalog ditampilkan sesuai kategori barang yang dipilih
Skenario	<ol style="list-style-type: none"> 1. Kategori barang ditampilkan di layar 2. Pembeli memilih salah satu kategori barang yang diinginkan

3. Skenario Use Case tampilKatalog

Aktor	Pembeli
Kondisi Awal	Pembeli telah memilih salah satu kategori barang
Kondisi Akhir	Informasi detail barang ditampilkan.
Skenario	<ol style="list-style-type: none"> 1. Katalog barang ditampilkan 2. Pembeli memilih salah satu barang 3. Informasi nama barang, satuan, gambar, stok, dan harga barang ditampilkan

4. Skenario Use Case pesanBarang

Aktor	Pembeli
Kondisi Awal	Katalog/informasi barang ditampilkan
Kondisi Akhir	Daftar pemesanan pembeli telah tersimpan
Skenario	<ol style="list-style-type: none"> 1. Pembeli memilih barang yang ingin dipesan 2. Pembeli menentukan kuantitas masing-masing barang 3. [jika ada barang yang ingin dibatalkan atau diganti] ubah daftar pemesanan 4. Pembeli menentukan jenis pengantaran pesanan 5. Total belanja dan biaya pengantaran ditampilkan 6. Pembeli melakukan konfirmasi pemesanan

5. Skenario Use Case aturAkun

Aktor	Pembeli
Kondisi Awal	Pembeli telah melakukan login
Kondisi Akhir	Data akun telah tersimpan.
Skenario	<ol style="list-style-type: none"> 1. Pembeli mengubah data password, alamat dan nomor <i>handphone</i>
Skenario Alternatif	1.a. [data yang dimasukan tidak lengkap atau tidak sesuai format] muncul pesan kesalahan dan form atur data akun pembeli kembali ditampilkan

6. Skenario Use Case tampilPesananSaya

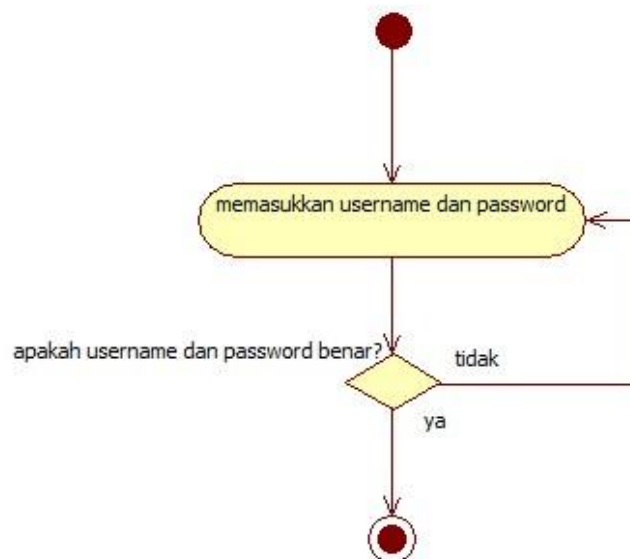
Aktor	Pembeli
Kondisi Awal	Pembeli telah melakukan login
Kondisi Akhir	Halaman Pesanan Saya ditampilkan
Skenario	1. Pembeli memilih halaman Pesanan Saya

7. Skenario Use Case tampilAturanBelanja

Aktor	Pembeli
Kondisi Awal	Pembeli telah melakukan login
Kondisi Akhir	Halaman Aturan Belanja ditampilkan
Skenario	1. Pembeli memilih halaman Aturan Belanja

3.3.3 Activity Diagram

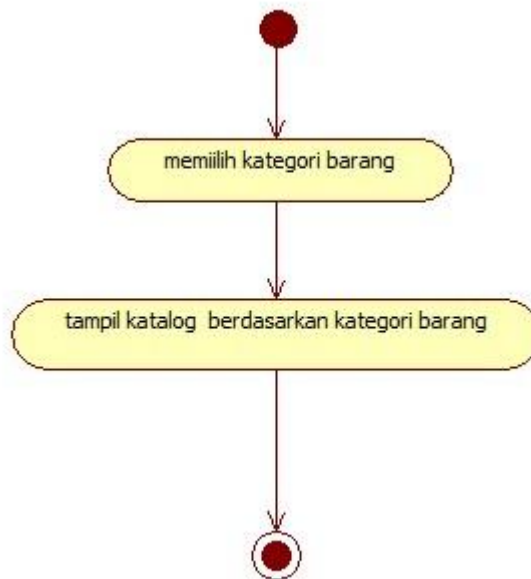
1. Activity Diagram login



Gambar 22 Activity Diagram login

Pada Gambar 22 dijelaskan bahwa untuk melakukan login, pembeli memasukkan username dan password. Kemudian sistem memvalidasi apakah username dan password benar atau salah. Jika benar maka aktivitas login selesai, jika salah maka sistem akan mengulangi proses login.

2. Activity Diagram tampilKategori



Gambar 23 Activity Diagram tampilKategori

Pada Gambar 23 dijelaskan bahwa untuk menampilkan kategori, pembeli harus memilih salah satu kategori. Lalu sistem akan menampilkan katalog berdasarkan kategori yang dipilih

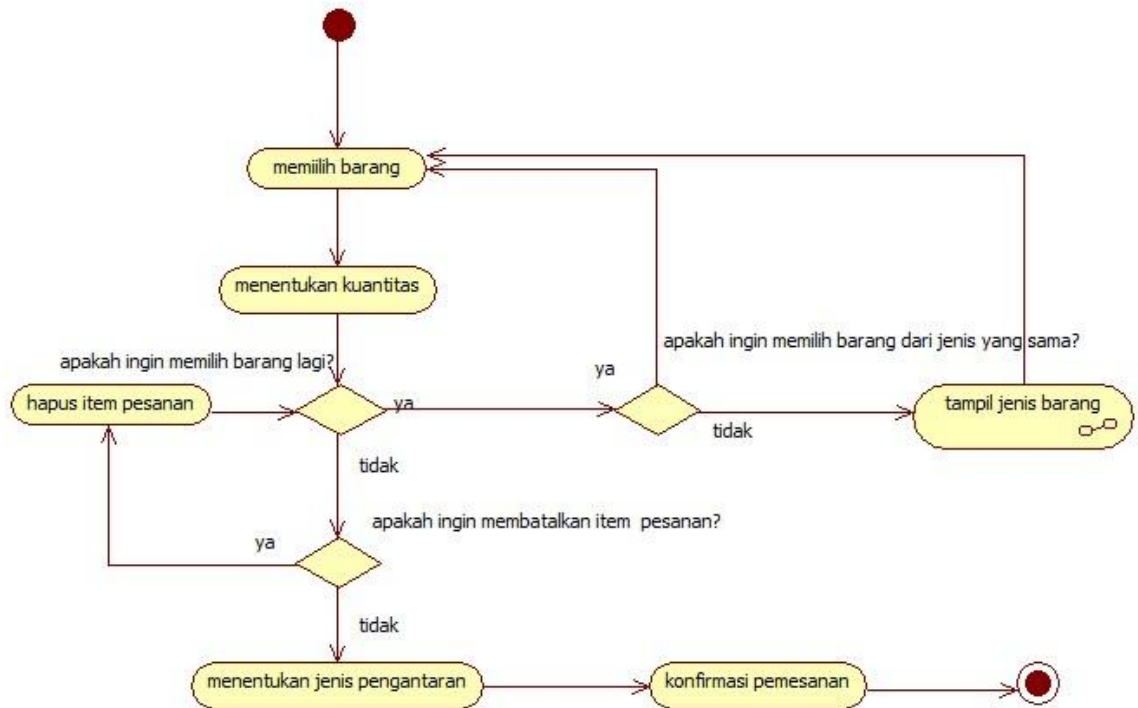
3. Activity Diagram tampilKatalog



Gambar 24 Activity Diagram tampilKatalog

Pada Gambar 24 dijelaskan bahwa untuk menampilkan katalog, pembeli harus memilih salah satu barang sehingga sistem akan menampilkan detail barang sesuai dengan barang yang dipilih.

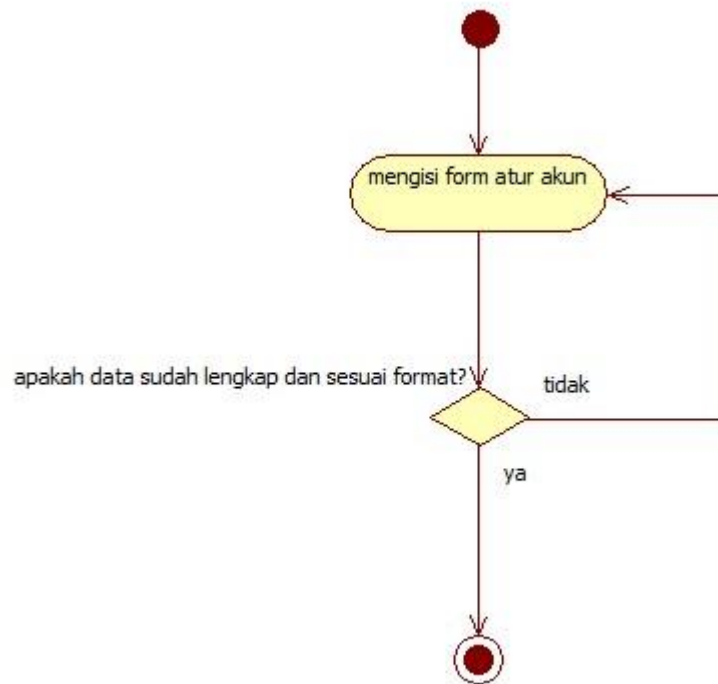
4. Activity Diagram pesanBarang



Gambar 25 Activity Diagram pesanBarang

Pada Gambar 25 dijelaskan bahwa untuk melakukan pemesanan barang, pembeli terlebih dahulu harus memilih barang dan menentukan kuantitasnya. Jika pembeli ingin memilih barang lagi dari jenis yang sama maka sistem akan mengulang proses pilih barang. Jika pembeli ingin memilih barang lagi dari jenis yang berbeda maka sistem akan menampilkan jenis barang. Pembeli dapat menghapus barang tidak jadi dipesan. Jika pembeli sudah selesai memilih barang, maka pembeli harus menentukan jenis pengantaran dan mengkonfirmasi pesanan tersebut.

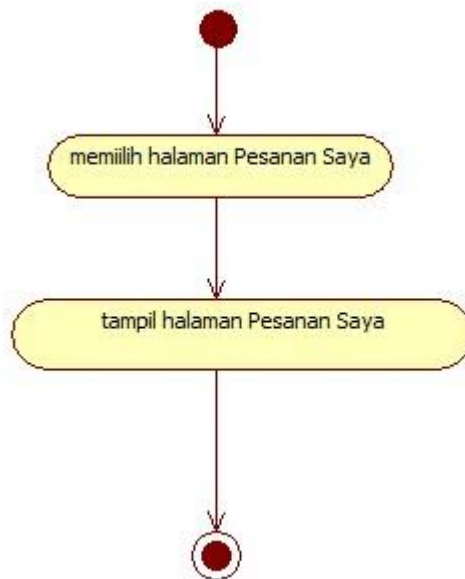
5. Activity Diagram aturAkun



Gambar 26 Activity Diagram aturAkun

Pada Gambar 26 dijelaskan bahwa pada proses atur akun pembeli harus mengisi seluruh data akun yang akan diganti. Jika data sudah lengkap dan sesuai format maka proses atur akun selesai. Jika data belum lengkap atau tidak sesuai format maka sistem akan mengulang proses atur akun.

6. Activity Diagram tampilPesananSaya



Gambar 27 Activity Diagram tampilPesananSaya

Pada Gambar 27 dijelaskan bahwa proses menampilkan pesanan saya, pembeli harus memilih salah satu data pesannya. Kemudian sistem akan menampilkan detail dari pesannya tersebut.

7. Activity Diagram tampilAturanBelanja

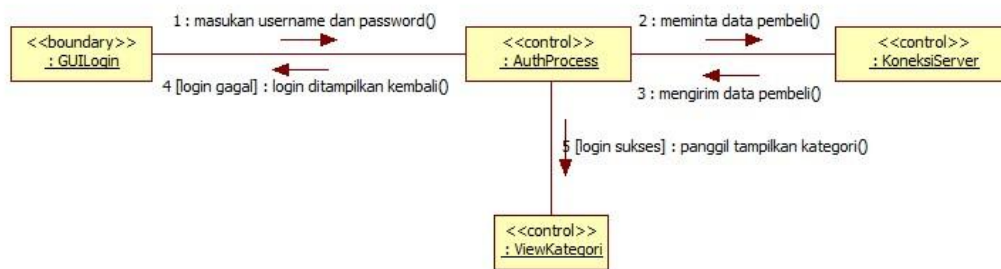


Gambar 28 Activity Diagram tampilAturanBelanja

Pada Gambar 28 dijelaskan bahwa proses menampilkan aturan belanja, pembeli harus memilih halaman aturan belanja. Kemudian sistem akan menampilkan halaman aturan belanja.

3.3.4 Communication Diagram

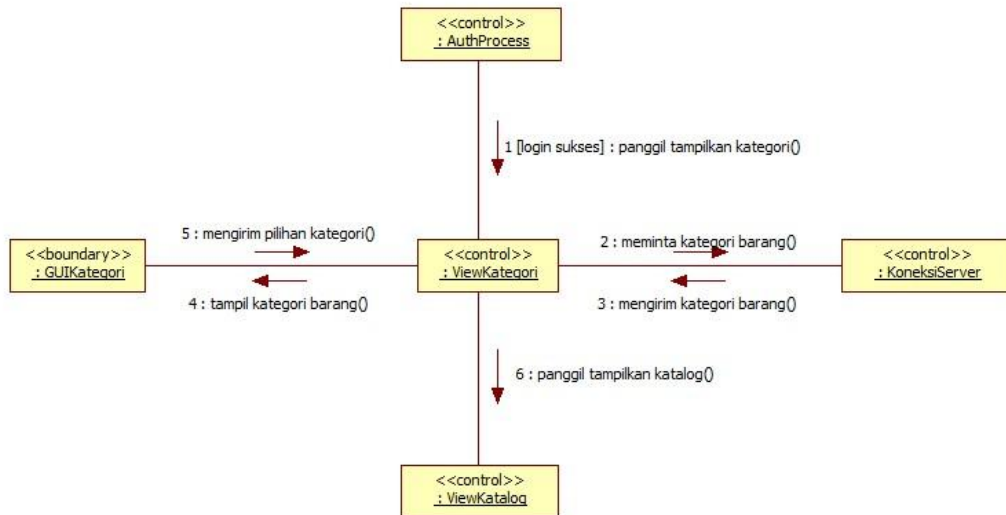
1. Communication Diagram login



Gambar 29 Communication Diagram login

Pada Gambar 29 dijelaskan bahwa proses login terdiri dari 3 komponen, yaitu boundary GUILogin yang digunakan untuk menampung masukkan berupa username dan password pembeli, control AuthProses digunakan untuk meminta data pembeli ke control KoneksiServer yang kemudian meneruskan permintaan tersebut ke server. Setelah mendapatkan data dari server, control KoneksiServer mengirimkan kembali data pembeli kepada control AuthProcess untuk kemudian dicocokkan. Jika data pembeli sesuai dengan masukkan maka control AuthProcess akan memanggil control ViewKategori. Jika data pembeli tidak sesuai dengan masukkan maka akan mengarah ke boundary GUILogin lagi.

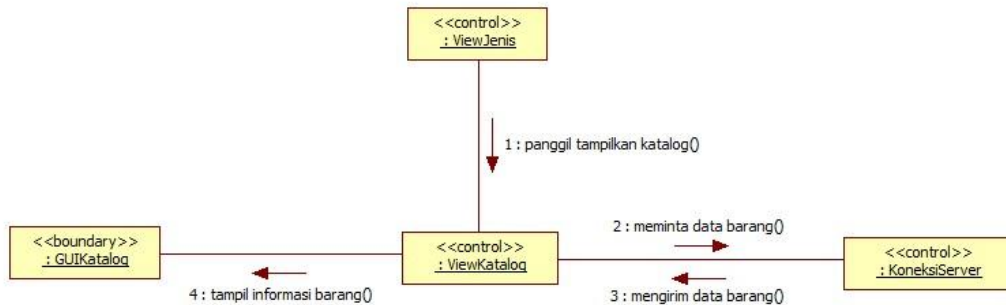
2. Communication Diagram tampilKategori



Gambar 30 Communication Diagram tampilKategori

Pada Gambar 30 dijelaskan bahwa proses tampilKategori terdiri dari 3 komponen utama yaitu boundary GUIKategori, control ViewKategori, dan control Koneksi Server. Control ViewKategori yang dipanggil oleh control AuthProcess setelah proses login berhasil dan berguna untuk meminta kategori barang kepada control KoneksiServer yang selanjutnya diteruskan ke server. Setelah mendapatkan data dari server, control KoneksiServer akan mengirimkan kategori barang ke control ViewKategori. Boundary GUIKategori berguna untuk menampilkan kategori barang dan menampung masukkan pilihan kategori barang yang selanjutnya akan dikirimkan ke control ViewKategori. Selanjutnya control ViewKategori akan memanggil control ViewKatalog.

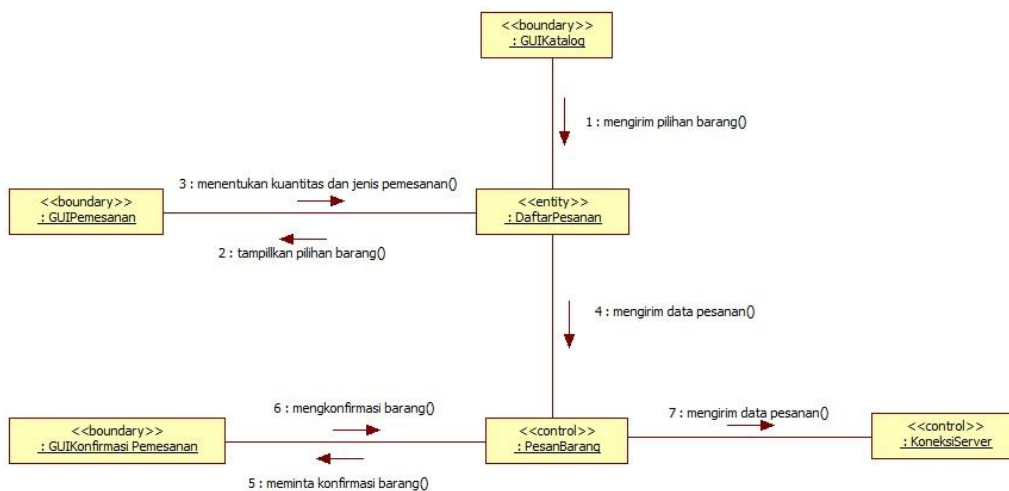
3. Communication Diagram tampilKatalog



Gambar 31 Communication Diagram tampilKatalog

Pada Gambar 31 dijelaskan bahwa proses tampilKatalog terdiri dari 3 komponen utama yaitu boundary GUIKatalog, control ViewKatalog, dan control KoneksiServer. Control ViewKatalog yang dipanggil oleh control ViewKategori berguna untuk meminta data barang ke control KoneksiServer yang kemudian diteruskan ke server. Control KoneksiServer akan mengirimkan data barang kepada control ViewKatalog. Boundary GUIKatalog berguna untuk menampilkan data barang.

4. Communication Diagram pesanBarang

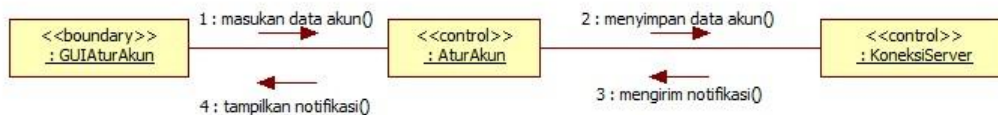


Gambar 32 Communication Diagram pesanBarang

Pada Gambar 32 dijelaskan bahwa proses pesanBarang terdiri dari 5 komponen utama yaitu boundary GUIPemesanan, entity DaftarPesanan, control PesanBarang, boundary GUIKonfirmasiPemesanan, dan control KoneksiServer.

KoneksiServer. Entity DaftarPesanan berfungsi menampung masukkan pilihan barang yang dikirimkan oleh boundary GUIKatalog. GUIPemesanan berguna untuk menampilkan pilihan barang dan menampung masukkan kuantitas dan jenis pengantaran yang kemudian akan dikirimkan kembali ke entity DaftarPesanan. Entity DaftarPesanan mengirimkan data pesanan ke control PesanBarang yang kemudian akan diteruskan ke GUIKonfirmasiPemesanan. GUIKonfirmasiPemesanan berguna untuk menampung masukkan konfirmasi pemesanan yang akan diteruskan ke control PesanBarang dan selanjutnya akan diteruskan ke server melalui control KoneksiServer.

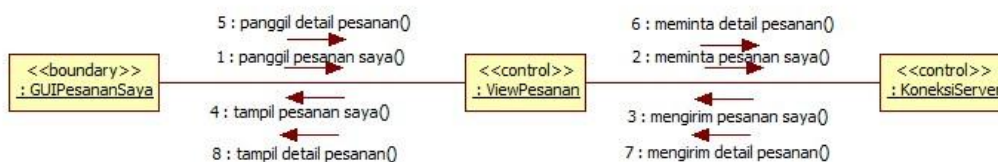
5. Communication Diagram aturAkun



Gambar 33 Communication Diagram aturAkun

Pada Gambar 33 dijelaskan bahwa proses aturAkun terdiri dari 3 komponen yaitu boundary GUIAturAkun, control AturAkun, dan control KoneksiServer. GUIAturAkun berguna untuk menampung masukkan atur akun. Control AturAkun berguna untuk mengirimkan data ke control KoneksiServer yang kemudian akan diteruskan ke server. Kemudian control KoneksiServer akan mengirimkan notifikasi ke control AturAkun dan ditampilkan oleh GUIAturAkun.

6. Communication Diagram tampilPesananSaya

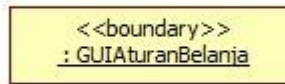


Gambar 34 Communication Diagram tampilPesananSaya

Pada Gambar 34 dijelaskan bahwa proses tampilPesananSaya terdiri dari 3 komponen yaitu boundary GUIPesananSaya, control ViewPesanan dan control KoneksiServer. Boundary GUIPesananSaya berguna untuk memanggil pesan saya yang akan diteruskan ke control KoneksiServer

oleh control ViewPesanan. Control KoneksiSever berguna untuk mengirimkan pesanan saya ke control ViewPesanan yang kemudian akan ditampilkan oleh boundary GUIPesananSaya. GUIPesananSaya akan memanggil detail pesanan ke control ViewPesanan yang akan diteruskan ke control KoneksiServer. Control KoneksiServer akan minta data ke server dan akan mengirimkannya ke control ViewPesanan yang kemudian akan ditampilkan oleh boundary GUIPesananSaya.

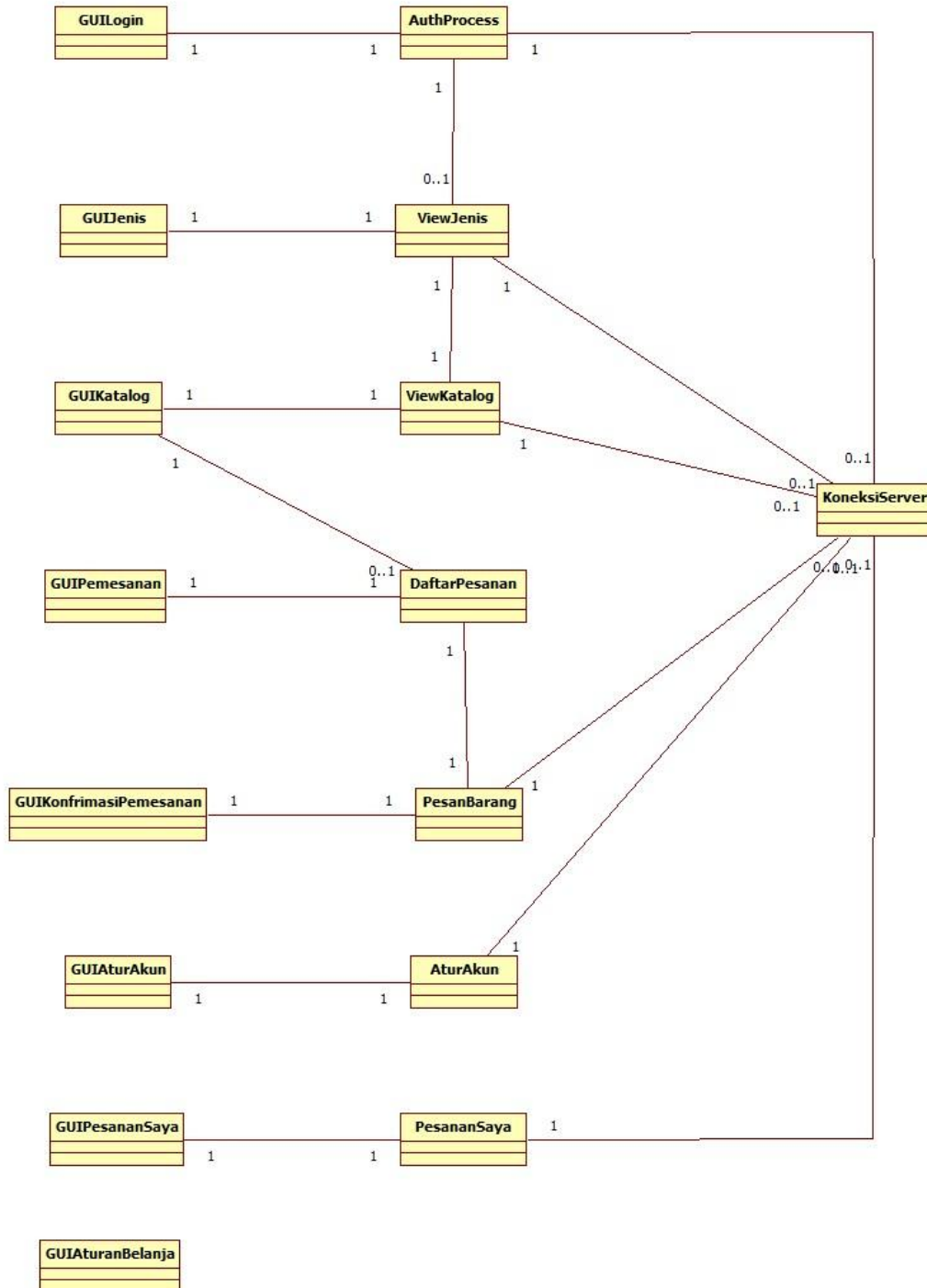
7. Communication Diagram tampilAturanBelanja



Gambar 35 Communication Diagram tampilAturanBelanja

Pada Gambar 35 dijelaskan bahwa proses tampilAturanBelanja hanya memiliki 1 komponen yaitu boundary GUIAturanBelanja yang akan menampilkan seluruh aturan belanja.

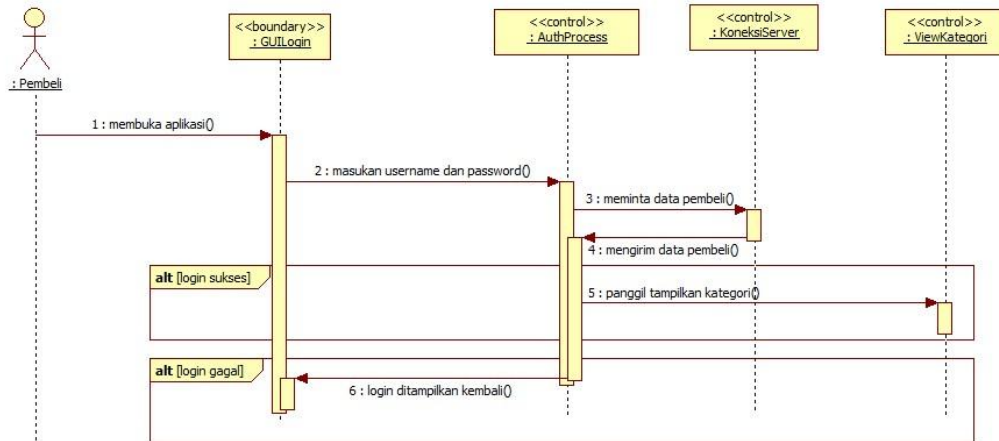
3.3.5 Class Diagram



Gambar 36 Class Diagram

3.3.6 Sequence Diagram

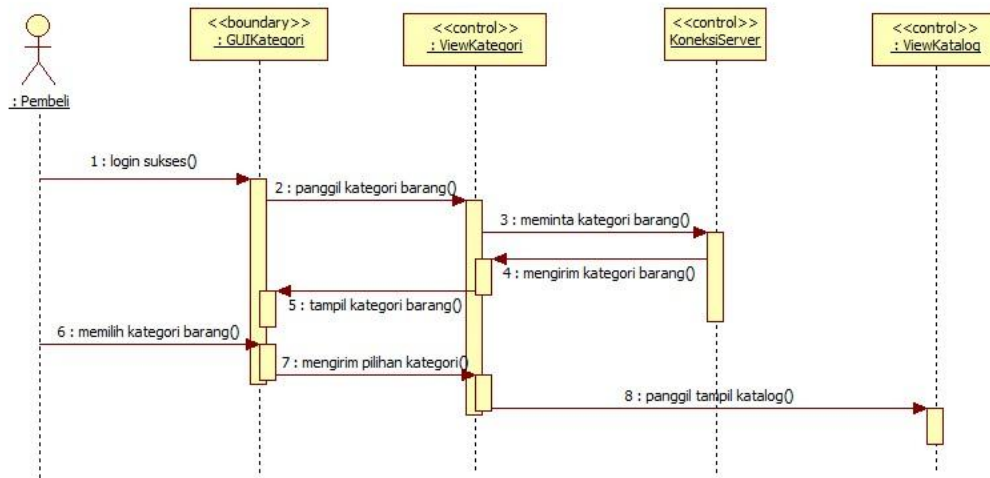
1. Sequence Diagram login



Gambar 37 Sequence Diagram login

Pada Gambar 37 dijelaskan bahwa proses login terdiri dari 3 komponen, yaitu boundary GUILogin yang digunakan untuk menampung masukkan berupa username dan password pembeli, control AuthProses digunakan untuk meminta data pembeli ke control KoneksiServer yang kemudian meneruskan permintaan tersebut ke server. Setelah mendapatkan data dari server, control KoneksiServer mengirimkan kembali data pembeli kepada control AuthProcess untuk kemudian dicocokkan. Jika data pembeli sesuai dengan masukkan maka control AuthProcess akan memanggil control ViewKategori. Jika data pembeli tidak sesuai dengan masukkan maka akan mengarah ke boundary GUILogin lagi.

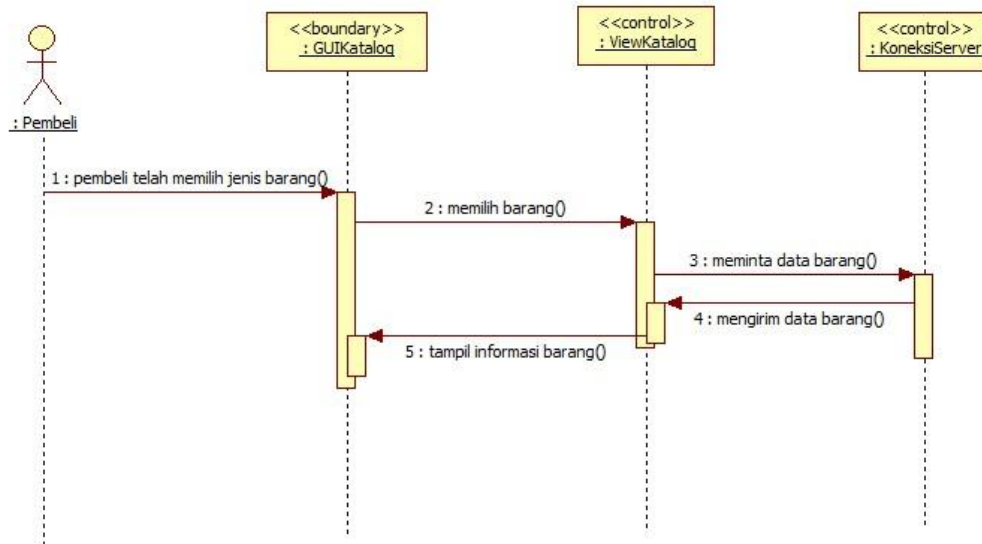
2. Sequence Diagram tampilKategori



Gambar 38 Sequence Diagram tampilKategori

Pada Gambar 38 dijelaskan bahwa proses tampilKategori terdiri dari 3 komponen utama yaitu boundary GUIKategori, control ViewKategori, dan control Koneksi Server. Control ViewKategori yang dipanggil setelah proses login berhasil dan berguna untuk meminta kategori barang kepada control KoneksiServer yang selanjutnya diteruskan ke server. Setelah mendapatkan data dari server, control KoneksiServer akan mengirimkan kategori barang ke control ViewKategori. Boundary GUIKategori berguna untuk menampilkan kategori barang dan menampung masukkan pilihan kategori barang yang selanjutnya akan dikirimkan ke control ViewKategori. Selanjutnya control ViewKategori akan memanggil control ViewKatalog.

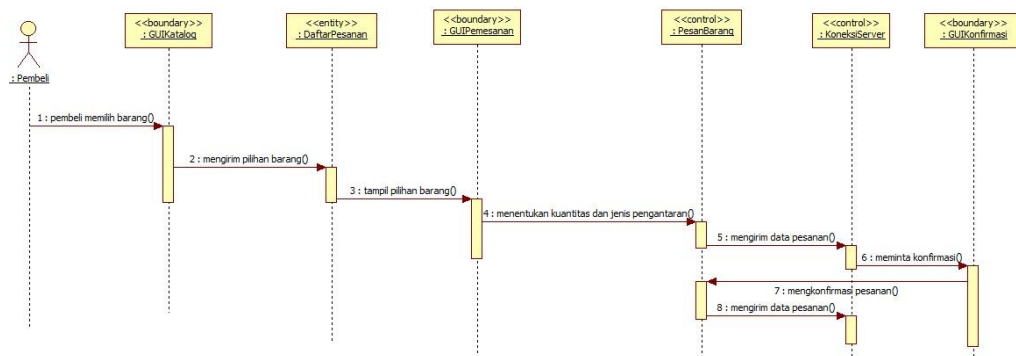
3. Sequence Diagram tampilKatalog



Gambar 39 Sequence Diagram tampilKatalog

Pada Gambar 39 dijelaskan bahwa proses tampilKatalog terdiri dari 3 komponen utama yaitu boundary GUIKatalog, control ViewKatalog, dan control KoneksiServer. Control ViewKatalog berguna untuk meminta data barang ke control KoneksiServer yang kemudian diteruskan ke server. Control KoneksiServer akan mengirimkan data barang kepada control ViewKatalog. Boundary GUIKatalog berguna untuk menampilkan data barang.

4. Sequence Diagram pesanBarang

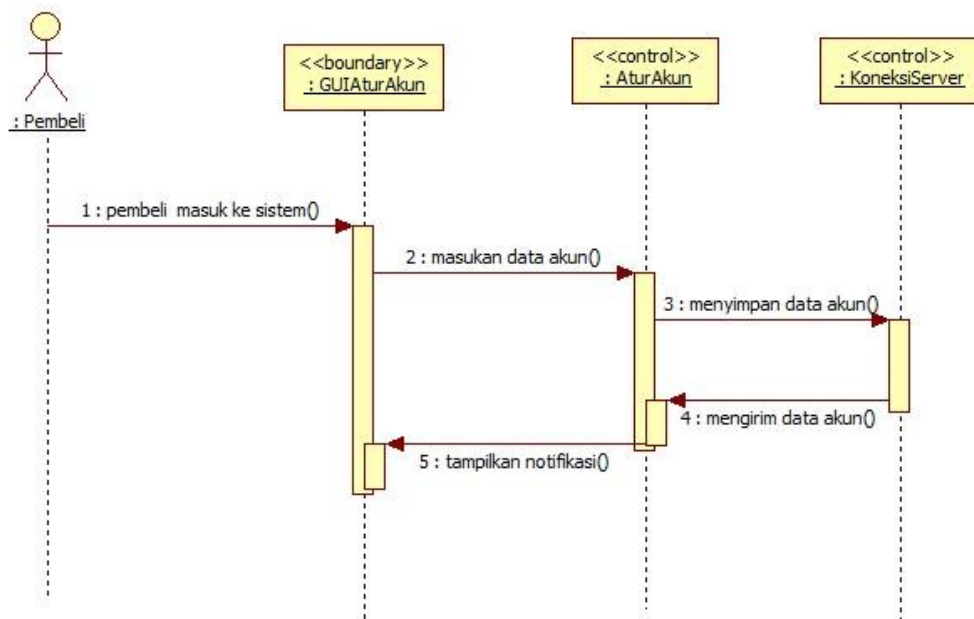


Gambar 40 Sequence Diagram pesanBarang

Pada Gambar 40 dijelaskan bahwa proses pesanBarang terdiri dari 5 komponen utama yaitu boundary GUIPemesanan, entity DaftarPesanan, control PesanBarang, control KoneksiServer, dan boundary GUIKonfirmasi.

control PesanBarang, boundary GUIKonfirmasiPemesanan, dan control KoneksiServer. Entity DaftarPesanan berfungsi menampung masukkan pilihan barang yang dikirimkan oleh boundary GUIKatalog. GUIPemesanan berguna untuk menampilkan pilihan barang dan menampung masukkan kuantitas dan jenis pengantaran yang kemudian akan dikirimkan kembali ke entity DaftarPesanan. Entity DaftarPesanan mengirimkan data pesanan ke control PesanBarang yang kemudian akan diteruskan ke GUIKonfirmasiPemesanan. GUIKonfirmasiPemesanan berguna untuk menampung masukkan konfirmasi pemesanan yang akan diteruskan ke control PesanBarang dan selanjutnya akan diteruskan ke server melalui control KoneksiServer.

5. Sequence Diagram aturAkun

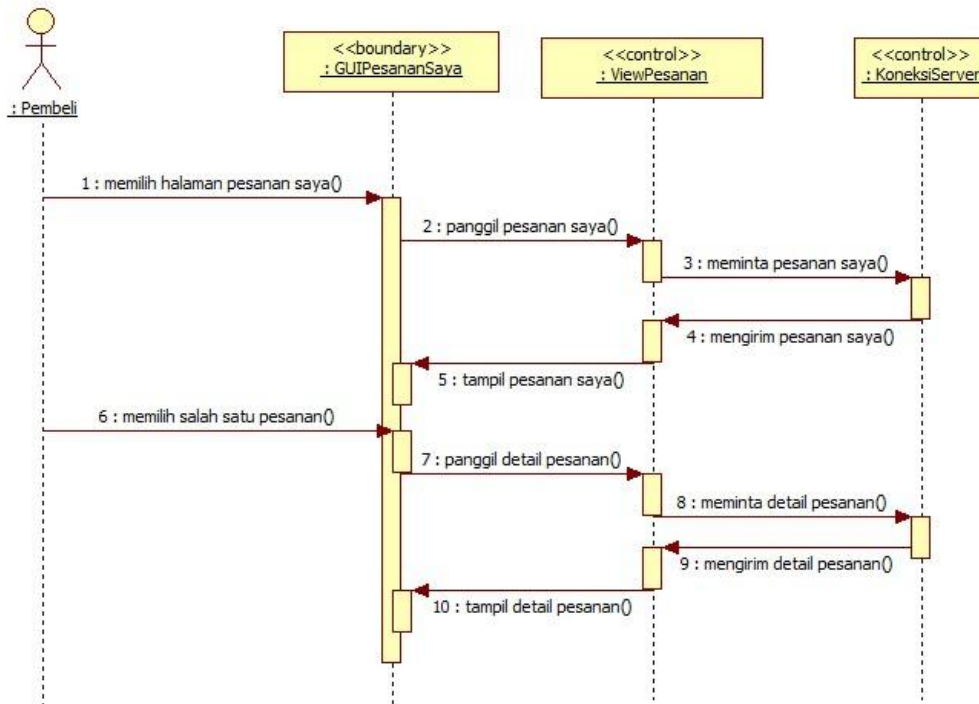


Gambar 41 Sequence Diagram aturAkun

Pada Gambar 41 dijelaskan bahwa proses aturAkun terdiri dari 3 komponen yaitu boundary GUIAturAkun, control AturAkun, dan control KoneksiServer. GUIAturAkun berguna untuk menampung masukkan atur akun. Control AturAkun berguna untuk mengirimkan data ke control KoneksiServer yang kemudian akan diteruskan ke server. Kemudian

control KoneksiServer akan mengirimkan notifikasi ke control AturAkun dan ditampilkan oleh GUIAturAkun.

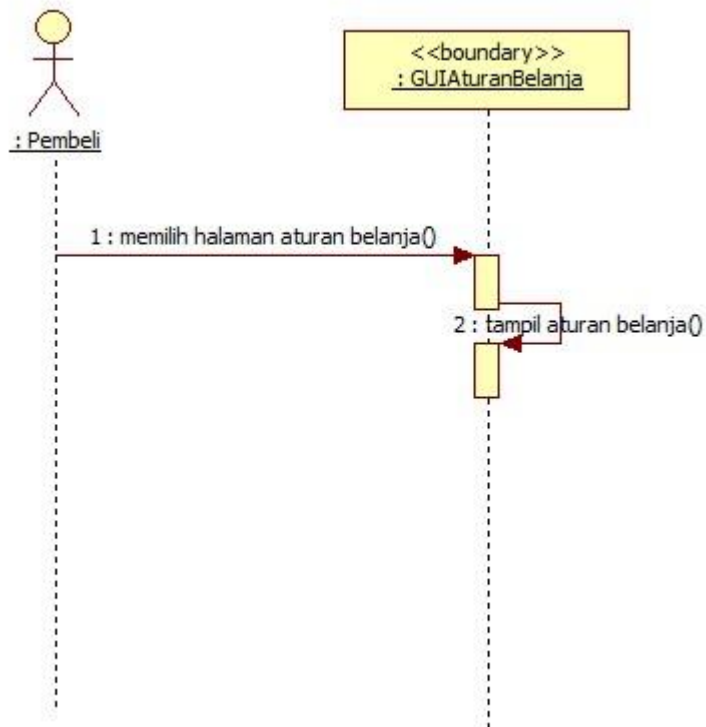
6. Sequence Diagram tampilPesananSaya



Gambar 42 Sequence Diagram tampilPesananSaya

Pada Gambar 42 dijelaskan bahwa proses tampilPesananSaya terdiri dari 3 komponen yaitu boundary GUIPesananSaya, control ViewPesanan dan control KoneksiServer. Boundary GUIPesananSaya berguna untuk memanggil pesanan saya yang akan diteruskan ke control KoneksiServer oleh control ViewPesanan. Control KoneksiServer berguna untuk mengirimkan pesanan saya ke control ViewPesanan yang kemudian akan ditampilkan oleh boundary GUIPesananSaya. GUIPesananSaya akan memanggil detail pesanan ke control ViewPesanan yang akan diteruskan ke control KoneksiServer. Control KoneksiServer akan minta data ke server dan akan mengirimkannya ke control ViewPesanan yang kemudian akan ditampilkan oleh boundary GUIPesananSaya.

7. Sequence Diagram tampilAturanBelanja

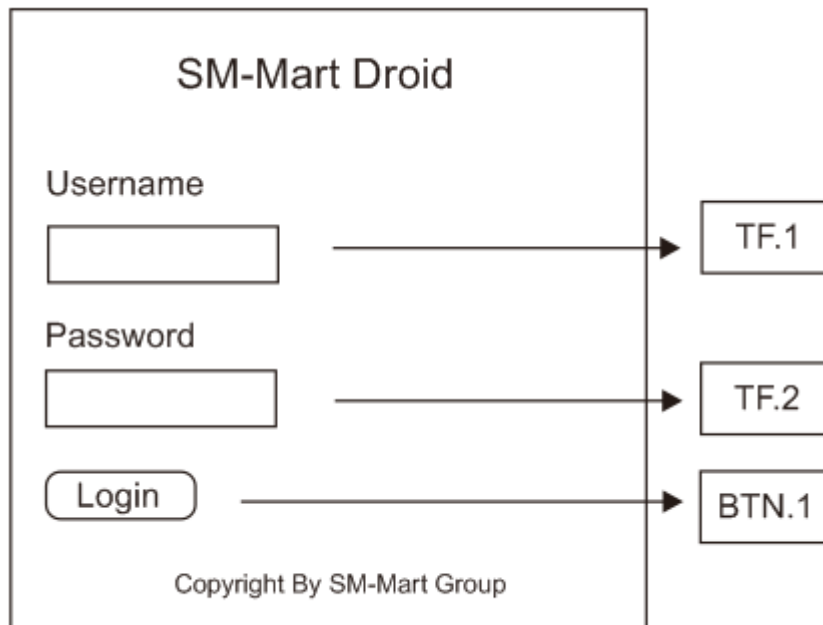


Gambar 43 Sequence Diagram tampilPesananSaya

Pada Gambar 35 dijelaskan bahwa proses tampilAturanBelanja hanya memiliki 1 komponen yaitu boundary GUIAturanBelanja yang akan menampilkan seluruh aturan belanja.

3.4 Perancangan Antarmuka

3.4.1 Perancangan Antarmuka Login



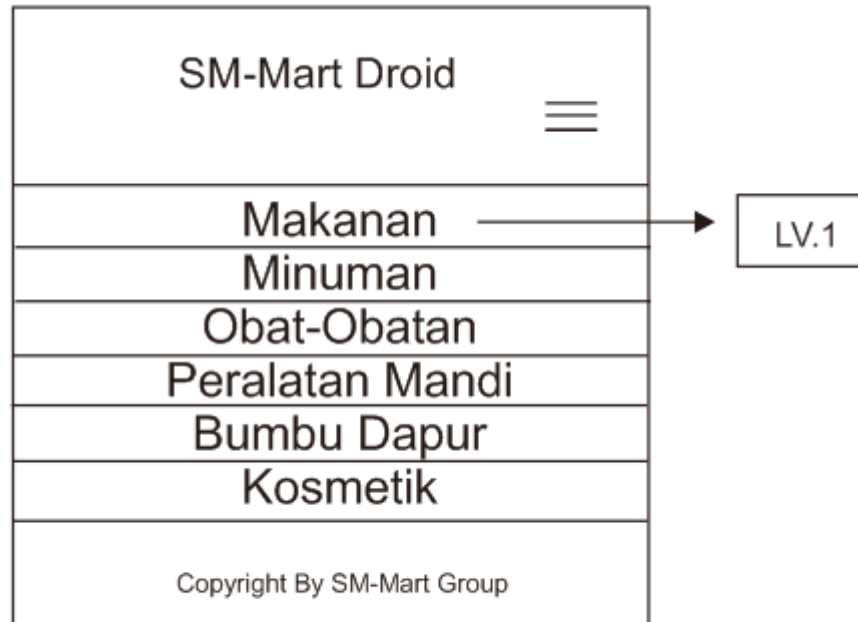
Gambar 44 Perancangan Antarmuka Login

Deskripsi perancangan antarmuka login, dijelaskan pada tabel 14.

Tabel 14 Deskripsi Perancangan Antarmuka Login

Id_objek	Jenis	Nama	Keterangan
TF.1	<i>Text Field</i>	<i>Username</i>	Diisi sesuai dengan <i>Username</i>
TF.2	<i>Text Field</i>	<i>Password</i>	Diisi sesuai dengan <i>Password</i>
BTN.1	<i>Button</i>	<i>Login</i>	Masuk ke dalam <i>website</i>

3.4.2 Perancangan Antarmuka Kategori Barang



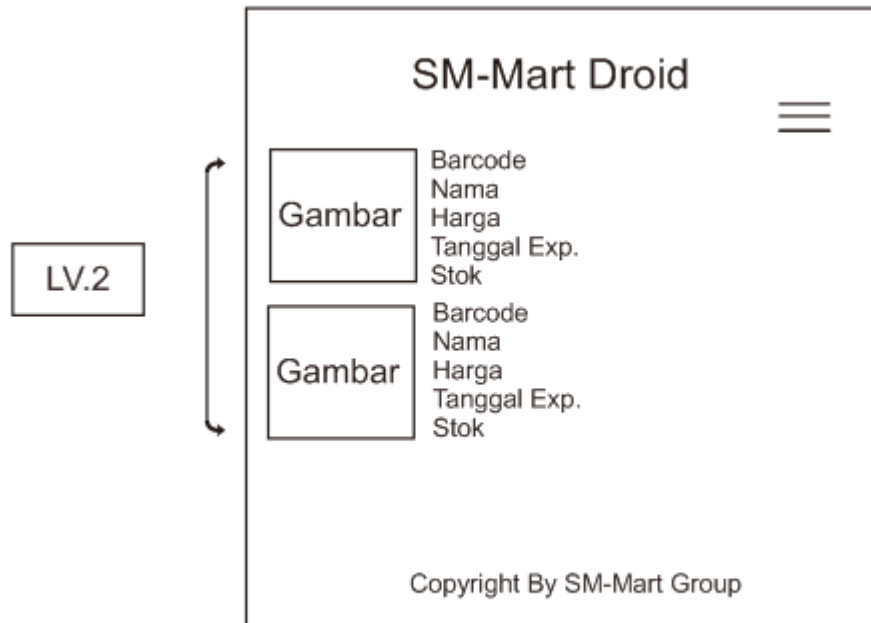
Gambar 45 Perancangan Antarmuka Kategori Barang

Deskripsi perancangan antarmuka kategori barang, dijelaskan pada tabel 15.

Tabel 15 Deskripsi Perancangan Antarmuka Kategori Barang

Id_objek	Jenis	Nama	Keterangan
LV. 1	<i>List View</i>	Daftar Kategori Barang	Menampilkan kategori barang

3.4.3 Perancangan Antarmuka Katalog Barang



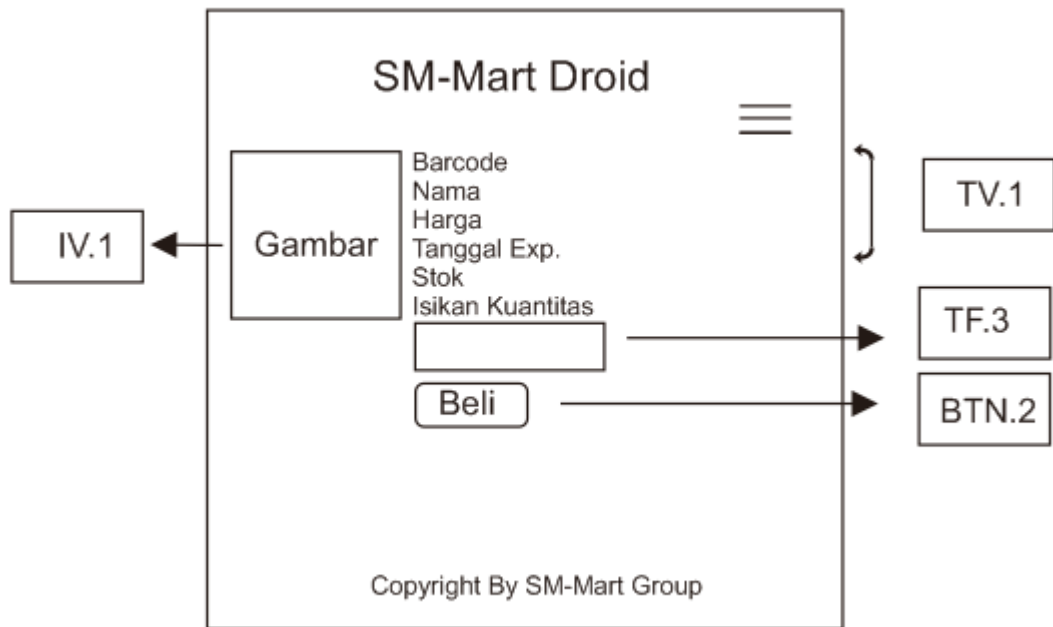
Gambar 46 Perancangan Antarmuka Katalog Barang

Deskripsi perancangan antarmuka katalog barang, dijelaskan pada tabel 16.

Tabel 16 Deskripsi Perancangan Antarmuka Katalog Barang

Id_objek	Jenis	Nama	Keterangan
LV. 2	<i>List View</i>	Daftar Barang	Menampilkan daftar katalog barang

3.4.4 Perancangan Antarmuka Pesan Barang



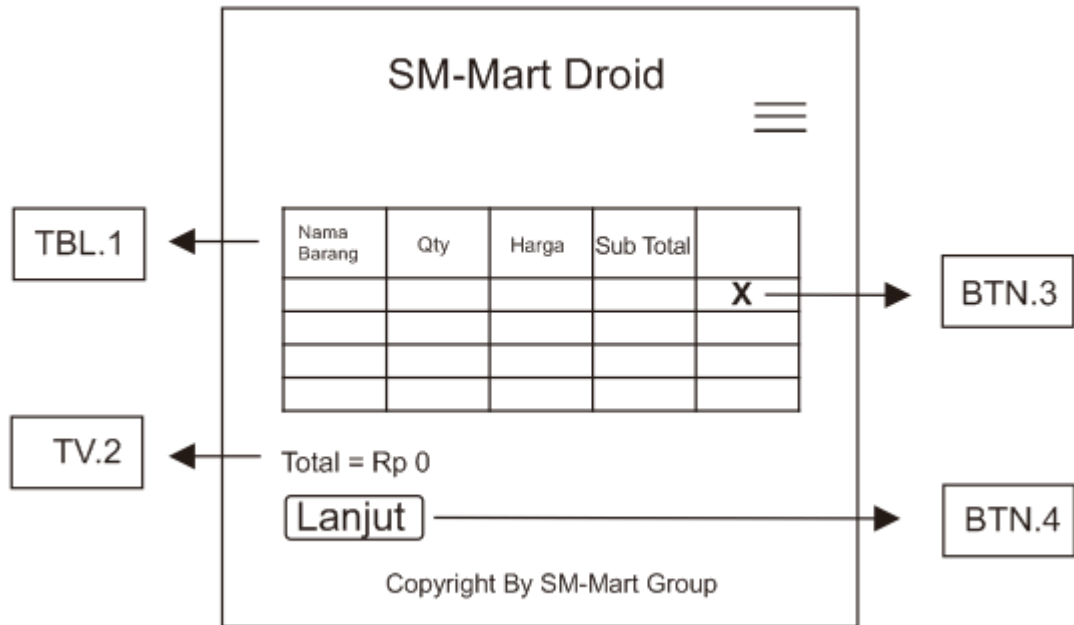
Gambar 47 Perancangan Antarmuka Pesan Barang

Deskripsi perancangan antarmuka pesan barang, dijelaskan pada tabel 17.

Tabel 17 Deskripsi Perancangan Antarmuka Pesan Barang

Id_objek	Jenis	Nama	Keterangan
IV. 1	<i>Image View</i>	Gambar Barang	Menampilkan gambar barang
TV.1	<i>Text View</i>	Detail Barang	Menampilkan detail barang
TF.3	<i>Text Field</i>	Kuantitas	Menampung masukan kuantitas
BTN. 2	<i>Button</i>	Tombol Pesan	Memesan barang

3.4.5 Perancangan Antarmuka Keranjang Belanja



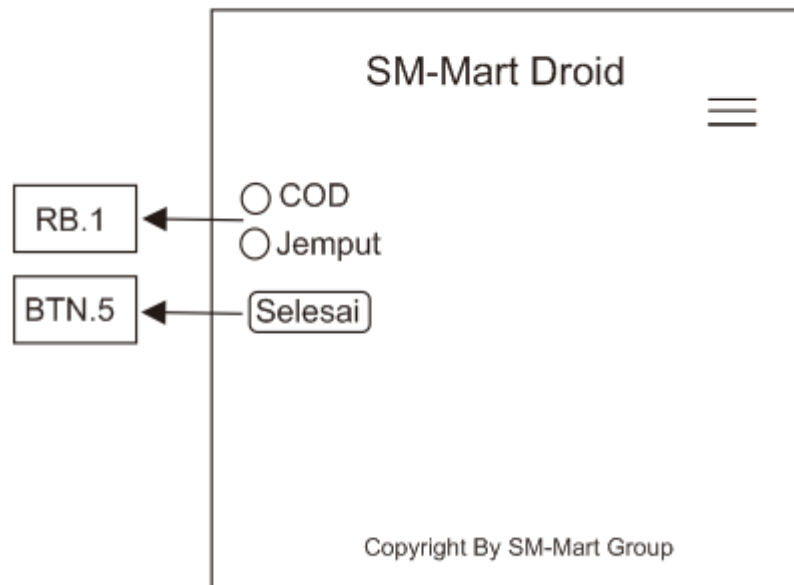
Gambar 48 Perancangan Antarmuka Keranjang belanja

Deskripsi perancangan antarmuka keranjang belanja, dijelaskan pada tabel 18.

Tabel 18 Deskripsi Perancangan Antarmuka Keranjang belanja

Id_objek	Jenis	Nama	Keterangan
TBL.1	<i>Table</i>	Keranjang belanja Sementara	Menampilkan keranjang belanja
TV.2	<i>Text View</i>	Total	Menampilkan total harga
BTN.3	<i>Button</i>	Hapus	Menghapus barang
BTN.4	<i>Button</i>	Lanjut	Mengkonfirmasi pesanan dan lanjut ke langkah selanjutnya

3.4.6 Perancangan Antarmuka Konfirmasi Jenis Pengantaran



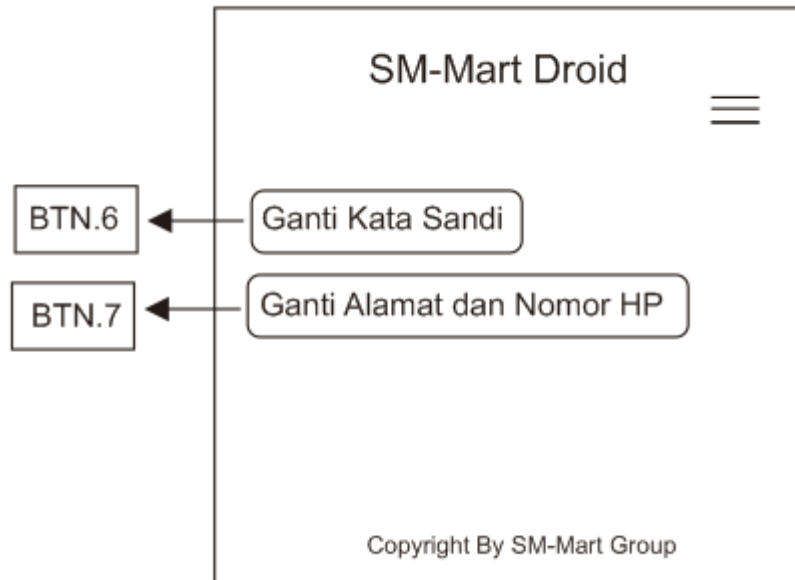
Gambar 49 Perancangan Antarmuka Konfirmasi Jenis Pengantaran

Deskripsi perancangan antarmuka Konfirmasi Jenis Pengantaran, dijelaskan pada tabel 19.

Tabel 19 Deskripsi Perancangan Antarmuka Konfirmasi Jenis Pengantaran

Id_objek	Jenis	Nama	Keterangan
RB. 1	<i>Radio Button</i>	Jenis Pengantaran	Pilihan jenis pengantaran
BTN. 5	<i>Button</i>	Selesai	Data pemesanan dikirim ke <i>server</i> dan masuk ke halaman selanjutnya

3.4.7 Perancangan Antarmuka Atur Akun



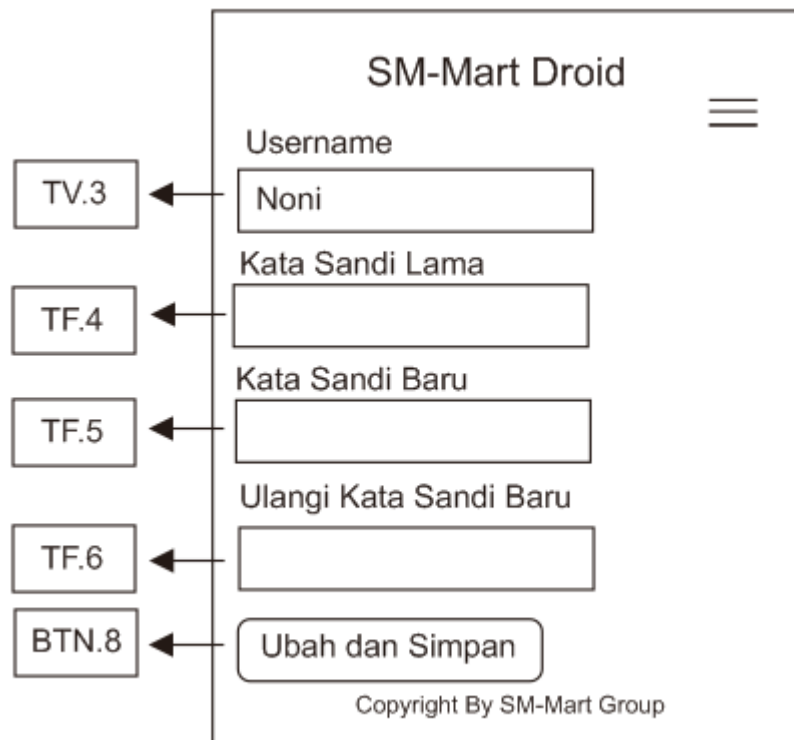
Gambar 50 Perancangan Antarmuka Atur Akun

Deskripsi perancangan antarmuka atur akun, dijelaskan pada tabel 20.

Tabel 20 Deskripsi Perancangan Antarmuka Atur Akun

Id_objek	Jenis	Nama	Keterangan
BTN. 6	<i>Button</i>	Ganti Password	Menampilkan halaman ganti password
BTN. 7	<i>Button</i>	Ganti Alamat dan Nomor HP	Menampilkan halaman ganti alamat dan nomor HP

3.4.8 Perancangan Antarmuka Ganti Kata sandi



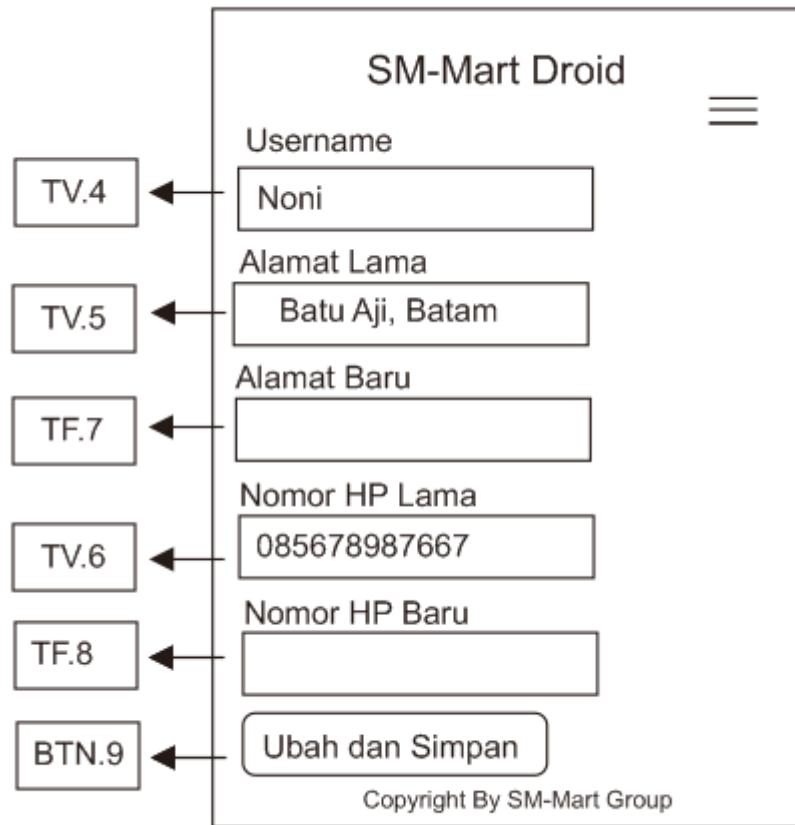
Gambar 51 Perancangan Antarmuka Ganti Kata Sandi

Deskripsi perancangan antarmuka ganti kata sandi, dijelaskan pada tabel 21.

Tabel 21 Deskripsi Perancangan Antarmuka Ganti Kata Sandi

Id_objek	Jenis	Nama	Keterangan
TV.3	<i>Text View</i>	Username	Menampilkan <i>username</i>
TF.4	<i>Text Field</i>	Kata Sandi Lama	Menampung masukan kata sandi lama
TF.5	<i>Text Field</i>	Kata Sandi Baru	Menampung masukan kata sandi baru
TF.6	<i>Text Field</i>	Ulangi Kata Sandi Baru	Menampung masukan kata sandi baru
BTN. 8	<i>Button</i>	Ubah dan Simpan	Menyimpan perubahan <i>kata sandi</i>

3.4.9 Perancangan Antarmuka Ganti Alamat dan Nomor HP



Gambar 52 Perancangan Antarmuka Ganti Alamat dan Nomor HP

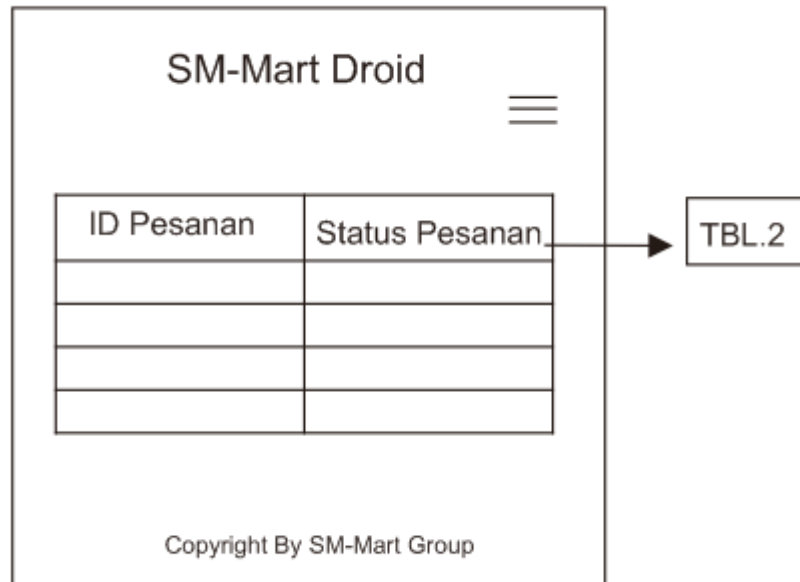
Deskripsi perancangan antarmuka ganti alamat dan nomor HP, dijelaskan pada tabel 22.

Tabel 22 Deskripsi Perancangan Antarmuka Ganti Alamat dan Nomor HP

Id_objek	Jenis	Nama	Keterangan
TV.4	<i>Text View</i>	Username	Menampilkan <i>username</i>
TV.5	<i>Text View</i>	Alamat Lama	Menampilkan alamat lama
TF.7	<i>Text Field</i>	Alamat Baru	Menampung masukan alamat baru
TV.6	<i>Text View</i>	Nomor HP Lama	Menampilkan nomor HP lama
TF.8	<i>Text Field</i>	Nomor HP Baru	Menampung masukan nomor HP baru
BTN. 9	<i>Button</i>	Ubah dan Simpan	Menyimpan

			perubahan alamat dan nomor HP
--	--	--	-------------------------------

3.4.10 Perancangan Antarmuka Pesanan Saya



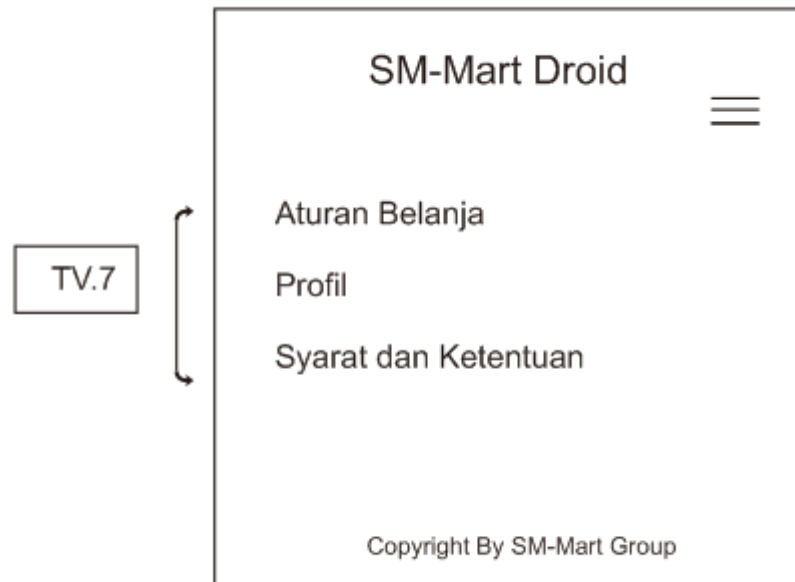
Gambar 53 Perancangan Antarmuka Pesanan Saya

Deskripsi perancangan antarmuka pesanan saya, dijelaskan pada tabel 23.

Tabel 23 Deskripsi Perancangan Antarmuka Pesanan Saya

Id_objek	Jenis	Nama	Keterangan
TBL.2	<i>Table</i>	Pesanan saya	Menampilkan data pesanan saya

3.4.11 Perancangan Antarmuka Aturan Belanja



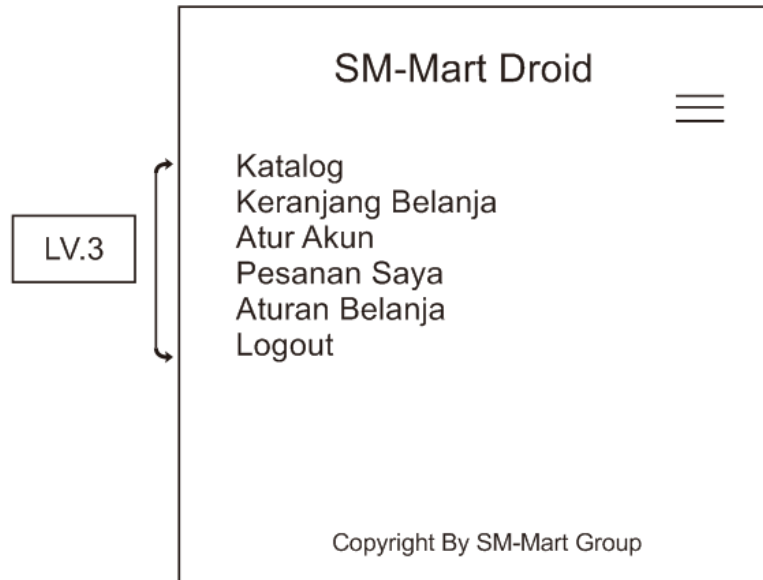
Gambar 54 Perancangan Antarmuka Aturan Belanja

Deskripsi perancangan antarmuka aturan belanja, dijelaskan pada tabel 24.

Tabel 24 Deskripsi Perancangan Antarmuka Aturan Belanja

Id_objek	Jenis	Nama	Keterangan
TV.7	<i>Text View</i>	Aturan Belanja	Menampilkan aturan belanja

3.4.12 Perancangan Antarmuka Fitur Option



Gambar 55 Perancangan Antarmuka Fitur Option

Deskripsi perancangan antarmuka fitur option, dijelaskan pada tabel 25.

Tabel 25 Deskripsi Perancangan Antarmuka Fitur Option

Id_objek	Jenis	Nama	Keterangan
LV. 3	<i>List View</i>	Fitur Option	Menampilkan fitur option

BAB IV
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Struktur Tabel Basisdata

Implementasi struktur tabel basisdata pada Aplikasi SM-Mart *Supermarket Online* Berbasis Android adalah sebagai berikut:

Nama tabel : t_pembeli

Tabel 26 Implementasi Struktur Tabel t_pembeli

Nama Field	Tipe	Panjang	Kunci	Atribut
no_ktp	Varchar	20	PK	Not Null
username	Varchar	20		Not Null
pass	Varchar	20		Not Null
gender	Enum('L', 'P')			Not Null
tgl_lahir	Date			Not Null
tempat_lahir	Varchar	50		Not Null
alamat	Varchar	50		Not Null
no_hp	Varchar	12		Not Null
nama	Varchar	50		Not Null

Nama tabel : t_pesanan

Tabel 27 Implementasi Struktur Tabel t_pesanan

Nama Field	Tipe	Panjang	Kunci	Atribut
id_pesanan	Varchar	50	PK	Not Null
username	Varchar	20	FK	Not Null
id_karyawan	Int	10	FK	Not Null
tgl_pesan	Date			Not Null
waktu_pesan	Time			Not Null
status_pesan	Enum('PESANAN DITERIMA, 'PESANAN DIPROSES', 'SELESAI DIPROSES', 'SUDAHHANTAR', 'SUDAHJEMPUT')			Not Null
jenis_pengantaran	Enum('ANTAR', 'JEMPUT')			Not Null
sudahlihat	Enum('y', 'n')			Not Null

Nama tabel : t_barang

Tabel 28 Implementasi Struktur Tabel t_barang

Nama Field	Tipe	Panjang	Kunci	Atribut
id_barang	Varchar	50	PK	Not Null
nama_barang	Varchar	50		Not Null
gambar	Varchar	50		Not Null
id_satuan	int	10	FK	Not Null
id_kategori	int	10	FK	Not null
harga_jual	Int	50		Not null
harga_beli	Int	50		Not Null

Nama tabel : t_kategori

Tabel 29 Implementasi Struktur Tabel t_kategori

Nama Field	Tipe	Panjang	Kunci	Atribut
id_kategori	Int	2	PK	Not Null
nama_kategori	Varchar	20		Not Null

Nama tabel : t_karyawan

Tabel 30 Implementasi Struktur Tabel t_karyawan

Nama Field	Tipe	Panjang	Kunci	Atribut
id_karyawan	Int	10	PK	Not null
nama_karyawan	Varchar	50		Not Null
pass_karyawan	Varchar	20		Not Null
jenis_karyawan	Enum('admin','kasir')			Not null

Nama tabel : t_detail_pesanan

Tabel 31 Implementasi Struktur Tabel t_detail_pesanan

Nama Field	Tipe	Panjang	Kunci	Atribut
id_pesanan	Int	8	FK	Not null
id_barang	Varchar	50	FK	Not Null
kuantitas	Int	10		Not Null
harga_jual	int	50	FK	

Nama tabel : t_pengantaran

Tabel 32 Implementasi Struktur Tabel t_pengantaran

Nama Field	Tipe	Panjang	Kunci	Atribut
id	Int	10	PK	Not null
nama	Varchar	20		Not Null
harga	Int	50		Not Null

Nama tabel : t_satuan

Tabel 33 Implementasi Struktur Tabel t_satuan

Nama Field	Tipe	Panjang	Kunci	Atribut
id_satuan	Int	10	PK	Not null
nama	Varchar	20		Not Null

Nama tabel : t_stok

Tabel 34 Implementasi Struktur Tabel t_stok

Nama Field	Tipe	Panjang	Kunci	Atribut
id	Int	10	PK	Not null
id_barang	Varchar	50	FK	Not Null
tgl_masuk	Date			Not Null
tgl_exp	Date			Not Null
stok_barang	Int	10		Not Null
id_kategori	Int	10	FK	Not Null

Berdasarkan perancangan, Aplikasi SM-Mart memerlukan 9 tabel basisdata, yaitu:

1. t_pembeli
2. t_pesanan
3. t_barang
4. t_kategori
5. t_karyawan
6. t_detail_pesanan
7. t_pengantaran

8. t_satuan

9. t_stok

Namun pada implementasinya, terdapat penambahan tabel yaitu tabel t_sementara. Tabel t_sementara digunakan untuk menyimpan data barang sebelum pesanan dikonfirmasi.

Nama tabel : t_sementara

Tabel 35 Implementasi Struktur Tabel t_sementara

Nama Field	Tipe	Panjang	Kunci	Atribut
id_pesanan	Varchar	50	FK	Not null
id_barang	Varchar	50	FK	Not Null
kuantitas	Int	10		Not Null
harga_jual	int	50	FK	
id	Int	10		Not Null

4.2 Implementasi Proses

4.2.1 Implementasi Proses Login

1. Implementasi Antarmuka Login

Implementasi antarmuka login, terdiri dari dua buah textfield dan 1 buah button. Untuk lebih memahami bisa lihat Gambar 56.



Gambar 56 Implementasi Antarmuka Login

Penjelasan tentang halaman login dapat dilihat di Tabel 36.

Tabel 36 Daftar Kode Sumber Halaman Login

Menu	Deskripsi	Nama File
Form Login	File untuk menangani inputan login pembeli	lpembeli.php
Login	File untuk menangani proses seleksi login pembeli	clp.php

2. Implementasi Source Code

Potongan kode yang menjelaskan cara kerja file clp.php dapat dilihat pada Tabel 37.

Tabel 37 Penjelasan Kode clp.php

Kode	Penjelasan
<code>\$nama = \$_POST['nama']; \$paswd = \$_POST['sand'];</code>	Mengambil variable dari halaman lpembeli.php
<code>\$query = "select * from t_pembeli where username='\$nama' and pass='\$paswd'";</code>	Query untuk menampilkan seluruh data pembeli yang username dan kata sandinya sama dengan masukan.
<code>if (\$r['username']==\$nama and \$r['pass']== \$paswd) { session_start(); \$_SESSION['nama'] = \$r['username']; \$_SESSION['sand'] = \$r['pass']; header('location:menu.php'); }</code>	Validasi apakah username dan kata sandi yang dimasukkan sama dengan di database. Jika iya maka membuat session untuk username dan kata sandi tersebut dan akan mengarah ke halaman menu.php
<code>elseif(\$r['username']==\$nama and \$r['pass']!= \$paswd){ echo "<script>alert('Username atau Password salah!'); window.location='lpembeli.php'</script>"; }</code>	Jika username dan kata sandi beda maka tampil pesan kesalahan dan akan mengarah ke halaman lpembeli.php

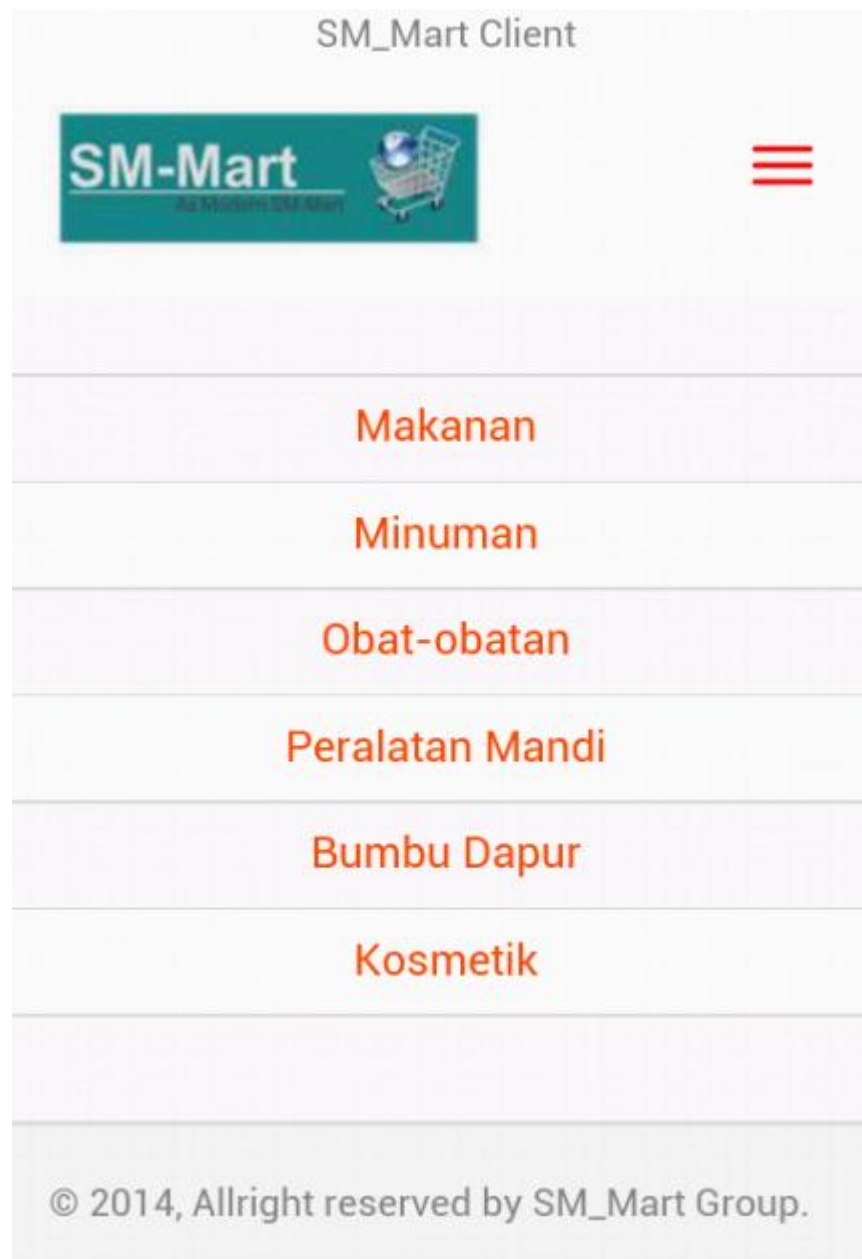
<pre>else{ echo "<script>alert('User belum terdaftar!'); window.location='lpembeli.php'</script>"; }</pre>	Jika semua kondisi tidak dipenuhi maka akan tampil pesan kesalahan dan akan mengarah ke halaman lpembeli.php
--	--

4.2.2 Implementasi Proses Tampil Kategori Barang

1. Implementasi Antarmuka Kategori Barang

Implementasi antarmuka kategori barang, terdiri dari satu listview.

Untuk lebih memahami bisa lihat Gambar 57.



Gambar 57 Implementasi Antarmuka Kategori Barang

Penjelasan tentang halaman kategori barang dapat dilihat di Tabel 38.

Tabel 38 Daftar Kode Sumber Halaman Kategori Barang

Menu	Deskripsi	Nama File
Kategori Barang	File untuk menampilkan kategori barang	menu.php

2. Implementasi Source Code

Potongan kode yang menjelaskan cara kerja file menu.php dapat dilihat pada Tabel 39.

Tabel 39 Daftar Kode Sumber Halaman Kategori Barang

Kode	Penjelasan
<pre>\$query = "select * from t_kategori";</pre>	Query menampilkan seluruh data kategori
<pre>while(\$data_bro = mysql_fetch_array(\$result)){ \$nama_barang = \$data_bro['nama_kategori']; \$id_barang = \$data_bro['id_kategori'];</pre>	Kondisi untuk menampilkan seluruh data selama data tersebut ada masih di database

4.2.3 Implementasi Proses Tampil Katalog

1. Implementasi Antarmuka Katalog

Implementasi antarmuka katalog, terdiri dari satu listview. Untuk lebih memahami bisa lihat Gambar 58.



Gambar 58 Implementasi Antarmuka Katalog

Penjelasan tentang halaman katalog dapat dilihat di Tabel 40.

Tabel 40 Daftar Kode Sumber Halaman Katalog Barang

Menu	Deskripsi	Nama File
Katalog Barang	File untuk menampilkan katalog barang	list.php

2. Implementasi Source Code

Potongan kode yang menjelaskan cara kerja file list.php dapat dilihat pada Tabel 41.

Tabel 41 Penjelasan Kode list.php

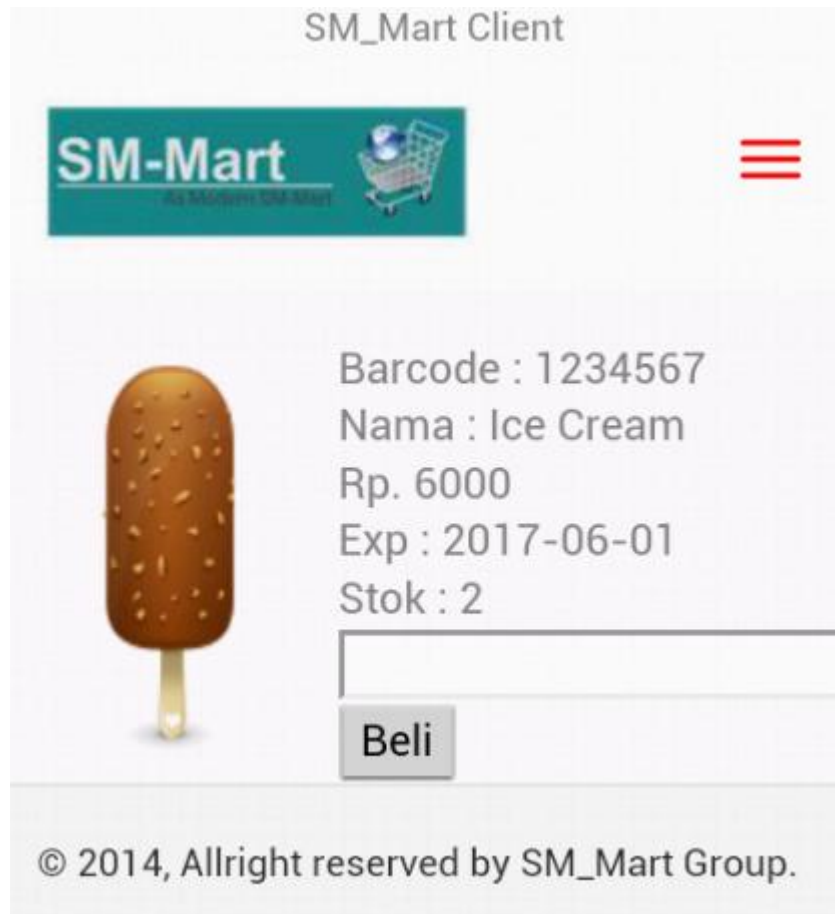
Kode	Penjelasan
<code>\$sql1 =mysql_query("select b.id_barang, b.id_kategori,</code>	Query untuk menampilkan barang

<pre>s.id_kategori, s.id_barang, SUM(s.stok_barang) stok, s.tgl_exp, b.nama_barang, b.harga_jual, b.gambar, s.tgl_masuk from t_barang b, t_stok s where b.id_kategori=s.id_kategori and b.id_barang=s.id_barang and b.id_kategori='\$nis' group by b.id_barang");</pre>	<p>sesuai dengan kategori yang dipilih.</p>
<pre>while(\$data=mysql_fetch_assoc(\$sql1)){ \$barcode = \$data['id_barang']; \$nama = \$data['nama_barang']; \$h_jual = \$data['harga_jual']; \$gambar = \$data['gambar']; \$tglexp = \$data['tgl_exp']; \$stok = \$data['stok'];</pre>	<p>Kondisi untuk menampilkan seluruh data selama data tersebut masih ada di database</p>

4.2.4 Implementasi Proses Pesan Barang

1. Implementasi Antarmuka Pesan Barang

Implementasi antarmuka detail barang, terdiri dari satu imageview, lima textview, satu textfield dan satu button. Untuk lebih memahami bisa lihat Gambar 59.



Gambar 59 Implementasi Antarmuka Detail Barang

Penjelasan tentang halaman detail barang dapat dilihat di Tabel 42.

Tabel 42 Daftar Kode Sumber Halaman Detail Barang

Menu	Deskripsi	Nama File
Detail Barang	File untuk menampilkan detail barang	beli.php proses_fifo.php
Inputan kuantitas	File untuk menangani	
Beli		

Implementasi antarmuka keranjang belanja, terdiri dari satu tabel, satu textview dan satu button. Untuk lebih memahami bisa lihat Gambar 60.



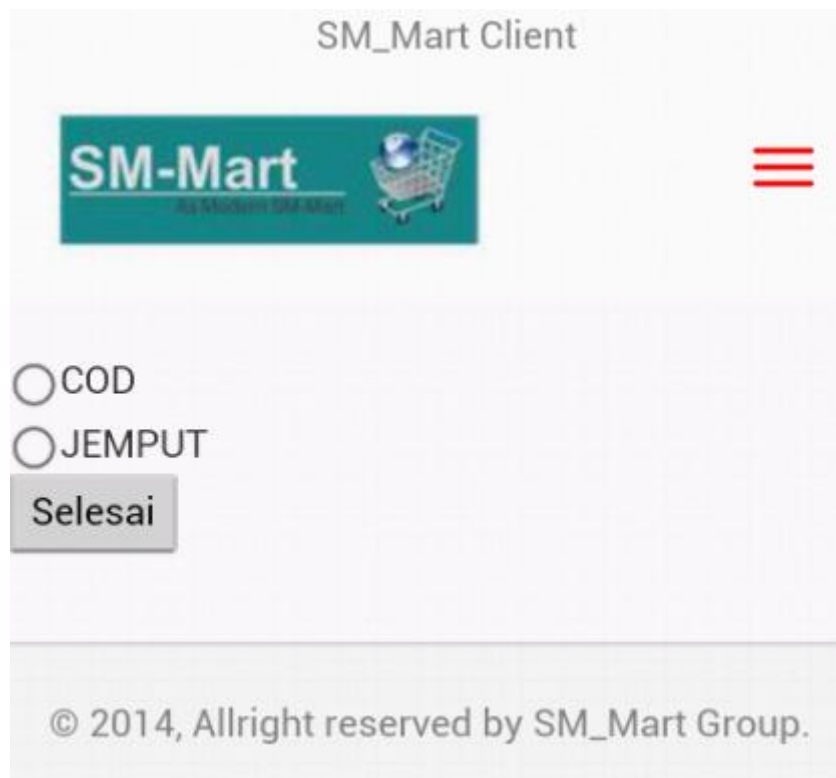
Gambar 60 Implementasi Antarmuka Keranjang Belanja

Penjelasan tentang halaman keranjang belanja dapat dilihat di Tabel 43.

Tabel 43 Daftar Kode Sumber Halaman Keranjang Belanja

Menu	Deskripsi	Nama File
Keranjang Belanja	File untuk menampilkan keranjang belanja	Wishlist.php

Implementasi antarmuka jenis pengantaran, terdiri dari dua radio button dan satu button. Untuk lebih memahami bisa lihat Gambar 61.



Gambar 61 Implementasi Antarmuka Jenis Pengantaran

Penjelasan tentang halaman jenis pengantaran dapat dilihat di Tabel 44.

Tabel 44 Daftar Kode Sumber Halaman Jenis Pengantaran

Menu	Deskripsi	Nama File
Pilih Pengantaran	File untuk menampilkan form pilih jenis pengantaran	Pilih.php

2. Implementasi Source Code

Potongan kode yang menjelaskan cara kerja file proses_fifo.php dapat dilihat pada Tabel 45.

Tabel 45 Penjelasan Kode proses_fifo.php

Kode	Penjelasan
<code>\$stok=mysql_query("select sum(t_stok.stok_barang) as stoksemua from t_stok where id_barang='\$barcode'");</code>	Query untuk mengambil penjumlahan stok berdasarkan barcode masukkan.
<code>while(\$fet=mysql_fetch_array(\$stok)){ \$ss=\$fet['stoksemua'];</code>	Mengambil data dari query \$stok
<code>if(\$ss >= \$pesanan)</code>	Validasi apakah data hasil query \$stok lebih besar sama dengan data

	kuantitas masukan
<pre>\$sql = "SELECT b.id_barang, b.harga_jual, s.id, s.id_barang, s.tgl_masuk, s.stok_barang from t_barang b, t_stok s where b.id_barang=s.id_barang and b.id_barang = '\$barcode' and s.stok_barang!='0' ORDER BY tgl_masuk";</pre>	Query untuk mengambil data barang dari t_barang dan t_stok berdasarkan tanggal masuk sesuai barcode masukkan
<pre>if(\$pesanan != 0)</pre>	Validasi pesanan tidak sama dengan 0
<pre>if(\$stock <= \$pesanan) \$stockawal=\$stock; \$pesanan -= \$stock; \$sisahabis=\$pesanan; \$stock-=\$stock;</pre>	Validasi jika \$stock lebih kecil sama dengan \$pesanan maka melakukan perhitungan
<pre>\$insert1= mysql_query("insert into t_sementara(id,id_pesanan,id_barang,kua ntitas,harga_jual) values ('\$id','\$id_pesanan','\$id_barang','\$sto ckawal','\$harga')");</pre>	Query untuk memasukkan data ke t_sementara
<pre>\$sql = "UPDATE t_stok set stok_barang='\$stock' where id='\$id'";</pre>	Query untuk mengupdate stok barang
<pre>Else { \$pesanan1 = \$pesanan; \$stock -= \$pesanan; \$pesanan -= \$pesanan;</pre>	Kondisi jika pesanan masih ada maka lakukan perintah pengurangan stok
<pre>\$insert2= mysql_query("insert into t_sementara (id,id_pesanan,id_barang,kuantitas,harg a_jual) values ('\$id','\$id_pesanan','\$id_barang','\$pes anan1','\$harga')");</pre>	Query untuk memasukkan data ke t_sementara
<pre>\$sql = "UPDATE t_stok SET stok_barang = '\$stock' WHERE `id` ='\$id'";</pre>	Query untuk mengupdate stok barang
<pre>Else { \$pesanan2=\$pesanan; \$stock -= \$pesanan; \$pesanan -= \$pesanan;</pre>	Jika kondisi diatas tidak terpenuhi maka jalankan perhitungan
<pre>\$insert2= mysql_query("insert into t_sementara (id,id_pesanan,id_barang,kuantitas,harg a_jual) values ('\$id','\$id_pesanan','\$id_barang','\$pes anan2','\$harga')");</pre>	Query untuk memasukkan data ke t_sementara
<pre>\$update=mysql_query("update t_stok set stok_barang='\$stock' where id='\$id'"); header("location:wishlist.php");</pre>	Query untuk mengupdate stok barang dan akan mengarah ke halaman wishlist.php
<pre>if(\$pesanan == 0){</pre>	Kondisi jika \$pesanan

<pre>break; }</pre>	nol
---------------------	-----

Potongan kode yang menjelaskan cara kerja file wishlist.php dapat dilihat pada Tabel 45.

Tabel 46 Penjelasan Kode wishlist.php

Kode	Penjelasan
<pre>\$sql1 = mysql_query("select t_sementara.id,t_barang.id_barang, t_barang.nama_barang, t_sementara.id_barang, sum(t_sementara.kuantitas) as qty,t_sementara.harga_jual,t_sementara .id_pesanan from t_barang, t_sementara where t_barang.id_barang=t_sementara.id_barang and t_sementara.id_pesanan='\$next_idtrx' group by t_barang.id_barang");</pre>	Query untuk menampilkan seluruh data di t_sementara
<pre>if(\$cek != null){ while(\$data = mysql_fetch_array(\$sql1)){ \$id = \$data['id_barang']; \$nama = \$data['nama_barang']; \$hjual= \$data['harga_jual']; \$kuantitas= \$data['qty']; \$total = \$hjual * \$kuantitas; \$Total=0; \$Total += \$total;</pre>	Untuk mengecek jumlah data di database dan menghitung subtotal.

Potongan kode yang menjelaskan cara kerja file pilih.php dapat dilihat pada Tabel 46.

Tabel 47 Penjelasan Kode pilih.php

Kode	Penjelasan
<pre><?php \$query=mysql_query("select * from t_pengantaran"); ?></pre>	Query untuk mengambil seluruh data di t_pengantaran

4.2.5 Implementasi Proses Atur Akun

1. Implementasi Antarmuka Atur Akun

Implementasi antarmuka atur akun terdiri dari dua button. Untuk lebih memahami bisa lihat Gambar 62.



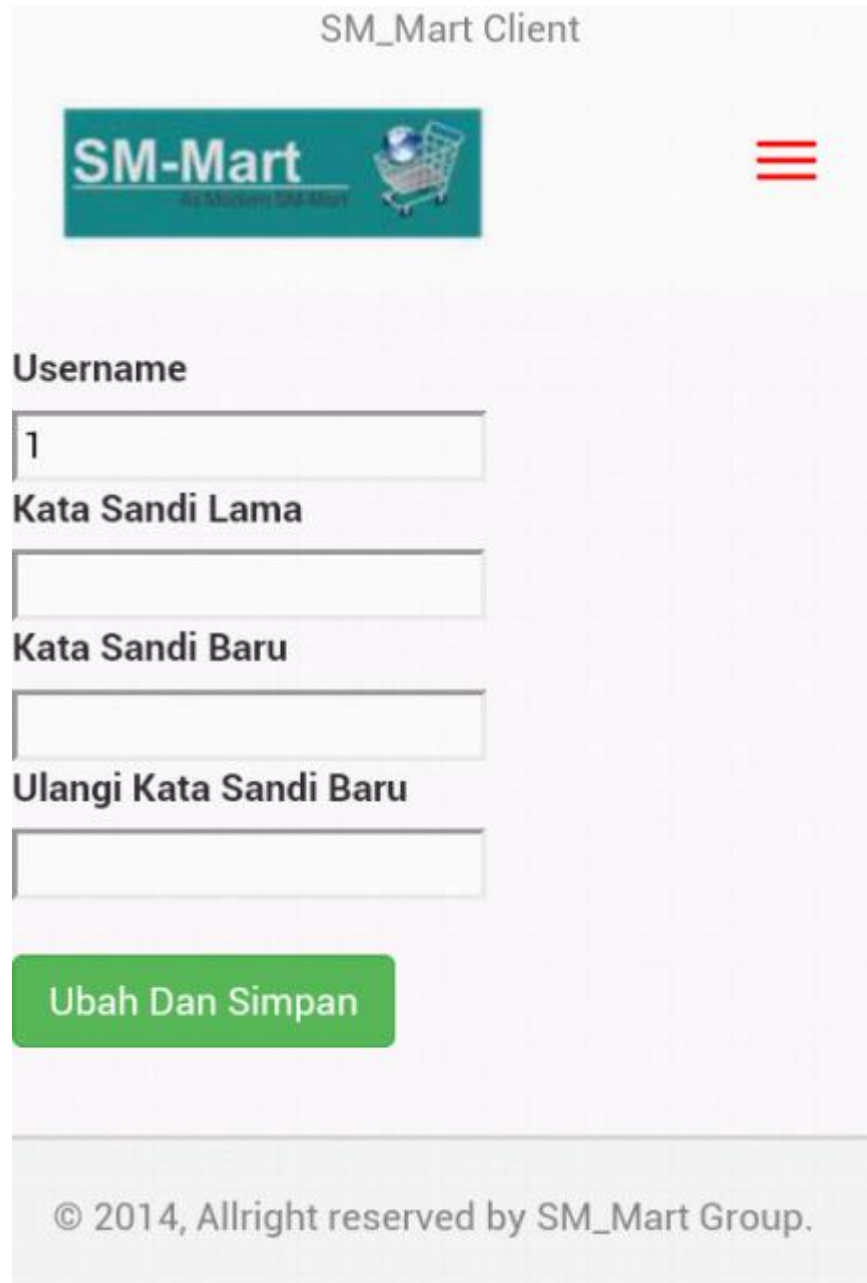
Gambar 62 Implementasi Antarmuka Atur Akun

Penjelasan tentang halaman atur akun dapat dilihat di Tabel 48.

Tabel 48 Daftar Kode Sumber Halaman Atur Akun

Menu	Deskripsi	Nama File
Atur Akun	File untuk menangani proses atur akun	gks.php
Ganti Kata Sandi	File untuk menampilkan halaman ganti kata sandi	edit_password.php
Ganti alamat dan nomor HP	File untuk menampilkan halaman ganti alamat dan nomor HP	gan.php

Implementasi antarmuka ganti kata sandi terdiri dari satu textview, tiga textfield dan satu button. Untuk lebih memahami bisa lihat Gambar 63.



Gambar 63 Implementasi Antarmuka Ganti Kata Sandi

Penjelasan tentang halaman ganti kata sandi dapat dilihat di Tabel 49.

Tabel 49 Daftar Kode Sumber Halaman Ganti Kata Sandi

Menu	Deskripsi	Nama File
Ganti Kata Sandi	File untuk menangani proses ganti kata sandi	proses_ganti_password.php

Implementasi antarmuka ganti alamat dan nomor hp terdiri dari tiga textview, dua textfield dan satu button. Untuk lebih memahami bisa lihat Gambar 64.

The screenshot shows the 'SM_Mart Client' app interface. At the top, there is a title 'SM_Mart Client' and a logo for 'SM-Mart' with a shopping cart icon. Below the logo is a red hamburger menu icon. The main content area contains several text views and text fields: 'Username' with a field containing '1', 'Alamat Lama' with a field containing 'Batam', 'Alamat Baru' with an empty field, 'Nomor Telepon Genggam Lama' with a field containing '085765464468', and 'Nomor Telepon Genggam Baru' with an empty field. At the bottom of the form is a green button labeled 'Ubah Dan Simpan'. The footer contains the text '© 2014, Allright reserved by SM_Mart Group.'

Gambar 64 Implementasi Antarmuka Ganti Alamat dan Nomor HP
Penjelasan tentang halaman ganti alamat dan nomor HP dapat dilihat di Tabel 50.

Tabel 50 Daftar Kode Sumber Halaman Ganti Alamat dan Nomor HP

Menu	Deskripsi	Nama File
Ganti alamat dan nomor HP	File untuk menangani proses ganti alamat dan nomor HP	proses_ganti_gan.php

2. Implementasi Source Code

Potongan kode yang menjelaskan cara kerja file proses_ganti_password.php dapat dilihat pada Tabel 51.

Tabel 51 Penjelasan Kode proses_ganti_password.php

Kode	Penjelasan
<code>\$user=\$_POST['nis']; \$new=\$_POST['katasandi2'];</code>	Mengambil variable dari halaman edit_password.php
<code>if(\$user != null)</code>	Mengecek apakah username ada
<code>\$query=mysql_query("update t_pembeli set pass='\$new' where username='\$user'"); header("location:gks.php?pesan=1&isi=Password Anda Berhasil Diganti");</code>	Query untuk mengupdate kata sandi dan mengarahkan ke gks.php dan menampilkan pesan sukses
<code>else { header("location:gks.php?pesan=2&isi=Password Gagal Diganti karena ".mysql_error()); }</code>	Jika kondisi tidak terpenuhi maka akan mengarah ke halaman gks.php dan tampil pesan kesalahan

Potongan kode yang menjelaskan cara kerja file proses_ganti_gan.php dapat dilihat pada Tabel 52.

Tabel 52 Penjelasan Kode proses_ganti_gan.php

Kode	Penjelasan
<code>\$user=\$_POST['nis']; \$ab=\$_POST['ab']; \$tb=\$_POST['tb'];</code>	Mengambil variable dari halaman gan.php
<code>if(\$user != null)</code>	Mengecek apakah username ada
<code>{ \$query=mysql_query("update t_pembeli set alamat='\$ab', no_hp='\$tb' where username='\$user'"); header("location:gks.php?pesan=1&isi=Alamat dan Nomor Telepon Anda Berhasil Diganti"); }</code>	Query untuk mengupdate alamat dan nomor hp dan akan mengarahkan ke halaman gks.php dan muncul pesan sukses.
<code>else {</code>	Jika kondisi tidak



<pre>header("location:gks.php?pesan=1&isi=Alamat dan Nomor Telepon Gagal Diganti karena ".mysql_error()); }</pre>	terpenuhi maka akan mengarah ke halaman gks.php dan tampil pesan kesalahan
---	--

4.2.6 Implementasi Proses Tampil Pesanan Saya

1. Implementasi Antarmuka Pesanan Saya

Implementasi antarmuka pesanan saya terdiri dari satu tabel. Untuk lebih memahami bisa lihat Gambar 65.

SM_Mart Client

ID Pesanan	Status
Konoha20140707000002	DITERIMA
Konoha20140707000001	DITERIMA
Konoha20140704000001	DITERIMA
Konoha20140701000001	SELESAI
Konoha20140630000004	DIPROSES
Konoha20140630000003	DITERIMA
Konoha20140630000002	DITERIMA

Gambar 65 Implementasi Antarmuka Pesanan Saya

Penjelasan tentang halaman pesanan saya dapat dilihat di Tabel 53.

Tabel 53 Daftar Kode Sumber Halaman Pesanan Saya

Menu	Deskripsi	Nama File
Pesanan Saya	File untuk menampilkan pesanan saya	pesanansaya.php

2. Implementasi Source Code

Potongan kode yang menjelaskan cara kerja file pesanansaya.php dapat dilihat pada Tabel 54.

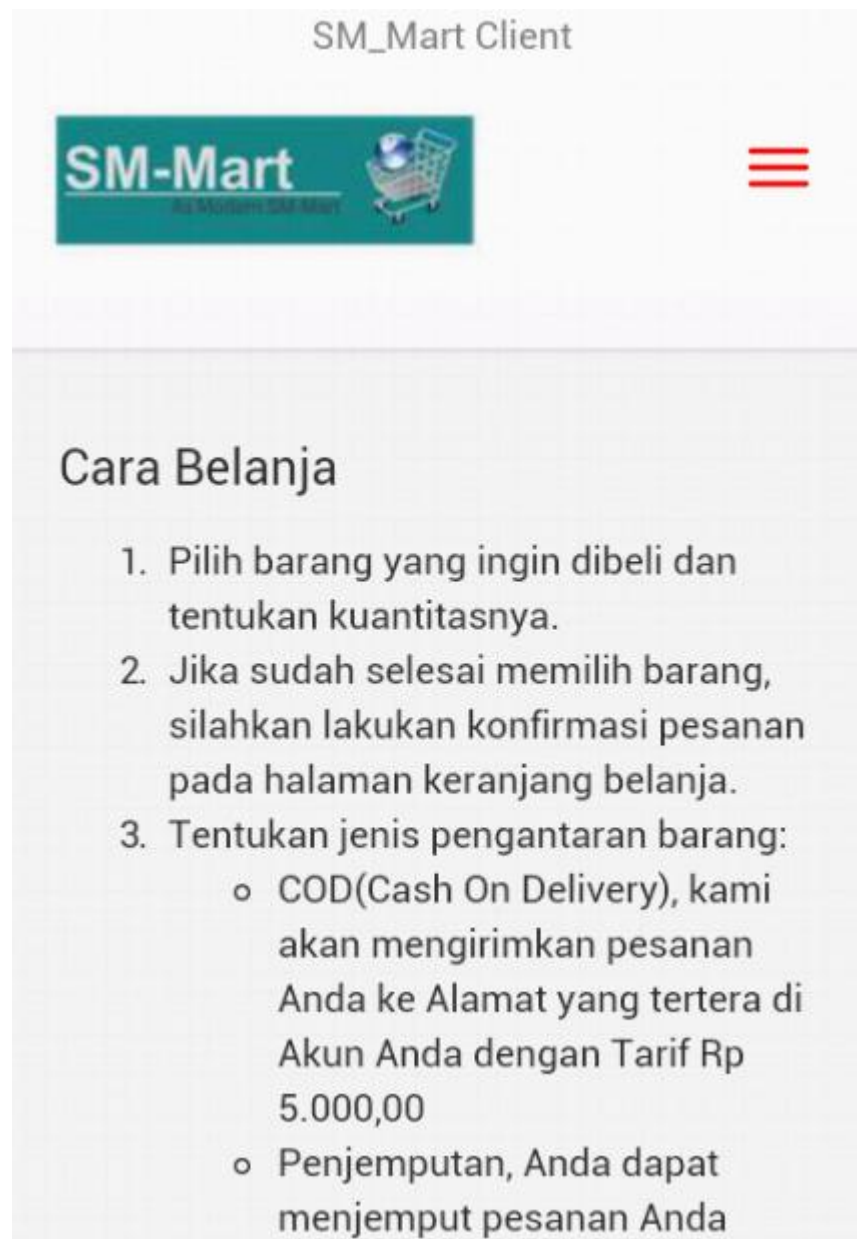
Tabel 54 Penjelasan Kode pesanansaya.php

Kode	Penjelasan
<code>\$user=\$_SESSION['nama'];</code>	Mengambil session berdasarkan username
<code>\$query = "select * from t_pesanan where username='\$user' order by id_pesanan DESC";</code>	Query untuk mengambil seluruh data dari t_pesanan berdasarkan id_pesanan

4.2.7 Implementasi Proses Tampil Aturan Belanja

1. Implementasi Antarmuka Aturan Belanja

Implementasi antarmuka pesanan saya terdiri dari satu textview. Untuk lebih memahami bisa lihat Gambar 66.



Gambar 66 Implementasi Antarmuka Aturan Belanja

Penjelasan tentang halaman aturan belanja dapat dilihat di Tabel 55.

Tabel 55 Daftar Kode Sumber Halaman Aturam Belanja

Menu	Deskripsi	Nama File
Aturan Belanja	File untuk menampilkan halaman aturan belanja	cara_belanja.php

2. Implementasi Source Code

Potongan kode yang menjelaskan cara kerja file cara_belanja.php dapat dilihat pada Tabel 56.

Tabel 56 Penjelasan Kode cara_belanja.php

Kode	Penjelasan
<pre> <div class='well' > <p> <h4>Cara Belanja</h4> Pilih barang yang ingin dibeli dan tentukan kuantitasnya. Jika sudah selesai memilih barang, silahkan lakukan konfirmasi pesanan pada halaman keranjang belanja. Tentukan jenis pengantaran barang: COD(Cash On Delivery), kami akan mengirimkan pesanan Anda ke Alamat yang tertera di Akun Anda dengan Tarif Rp 5.000,00 Penjemputan, Anda dapat menjemput pesanan Anda langsung ke Supermarket kami Pembayaran hanya dapat dilakukan secara tunai dan langsung pada saat serah-terima barang Anda dapat melihat status pesanan Anda pada halaman Pesanan Saya <h4>Akun</h4> Anda dapat mengubah data Akun Anda berupa password, alamat dan no.HP pada halaman Atur Akun. Jika Anda lupa password Anda, silahkan hubungi karyawan Supermarket Kami. <h4>Syarat & Ketentuan</h4> Pesanan yang sudah di konfirmasi tidak dapat dibatalkan lagi. Jika terjadi penipuan pesanan, maka kami akan mem-blacklist pelanggan dan melaporkan ke pihak yang berwajib </p> </div> </pre>	<p>Untuk menampilkan aturan belanja.</p>

4.3 Pengujian

Nama Penguji : Nur Zahрати Janah, M. Sc
 Tanggal Pengujian : 7 Juli 2014
 Tempat Pengujian : Ruang 702, Politeknik Negeri Batam

No	Nama Proses	Skenario Uji	Data Uji	Target	Validasi
1	Login	1. Pembeli memasukkan username dan password 2. Pembeli mengklik tombol "Login"	Username : "shintia" Password : "shintia1995"	Login berhasil dan masuk ke halaman "Kategori Barang"	
			Username : "shintia" Password : "shintia"	Login gagal dan muncul pesan kesalahan	
2	Pilih Kategori	1. Pembeli memilih salah satu kategori barang	Kategori : "Makanan"	Muncul katalog Makanan	
3	Pilih Barang	1. Pembeli memilih salah satu barang	Nama Barang: "Coklat"	Muncul detail barang dengan nama "Coklat"	
4	Pesan Barang	1. Pembeli memilih salah satu barang 2. Pembeli memasukkan kuantitas barang 3. Pembeli mengklik tombol "Beli" 4. Pembeli memilih halaman "Keranjang Belanja" 5. Pembeli mengklik tombol	Nama Barang: "Coklat" Kuantitas: "1"	Pesanan masuk ke server dan muncul di halaman "Keranjang Belanja"	
			Nama Barang: "Coklat" Kuantitas: tidak diisi	Muncul pesan kesalahan	

		<p>“Lanjut”</p> <p>6. Pembeli menentukan jenis pengantaran</p> <p>7. Pembeli mengklik tombol “Selesai”</p>	<p>Nama Barang: “Coklat”</p> <p>Kuantitas:” a”</p>	Muncul pesan kesalahan	
			<p>Nama Barang: “Coklat”</p> <p>Kuantitas: “1000”</p>	Muncul pesan kesalahan	
5	Atur Akun	<p>1. Pembeli memilih halaman “Atur Akun”</p> <p>2. Pembeli mengklik tombol “Ganti Password”</p> <p>3. Pembeli mengisi form ganti password</p> <p>4. Pembeli mengklik tombol “Ubah dan Simpan”</p>	<p>Username : “shintia”</p> <p>Password Lama : “shintia1995”</p> <p>Password Baru : “shintia95”</p> <p>Ulangi Password Baru : “shintia95”</p>	Proses Ganti Password berhasil.	
			<p>Username : “shintia”</p> <p>Password Lama : “shintia995”</p> <p>Password Baru : “shintia95”</p> <p>Ulangi Password Baru : “shintia95”</p>	Muncul pesan kesalahan	
			<p>Username : shintia</p> <p>Password Lama : “shintia1995”</p> <p>Password Baru : “shintia95”</p> <p>Ulangi Password Baru : “shintia9”</p>	Muncul pesan kesalahan	
			<p>Username : “shintia”</p>	Muncul pesan	

			Password Lama : “shintia1995” Password Baru : “shintia95” Ulangi Password Baru : tidak diisi	kesalahan	
6		<ol style="list-style-type: none"> 1. Pembeli memilih halaman “Atur Akun” 2. Pembeli mengklik tombol “Ganti Alamat dan Nomor HP” 3. Pembeli mengisi form ganti alamat dan no. hp 4. Pembeli mengklik tombol “Ubah dan Simpan” 	Alamat : “Anggrek Sari Blok A No. 7 Batam Centre” Nomor HP :”085765476589”	Proses Ganti Alamat dan Nomor HP berhasil.	
			Alamat :” Anggrek Sari Blok A No. 7 Batam Centre” Nomor HP :tidak diisi	Muncul pesan kesalahan	
			Alamat : “Anggrek Sari Blok A No. 7 Batam Centre” Nomor HP :”abc”	Muncul pesan kesalahan	
7	Lihat Pesanan Saya	<ol style="list-style-type: none"> 1. Pembeli memilih halaman “Pesanan Saya” 2. Pembeli memilih salah satu pesanannya 		Muncul halaman “Pesanan Saya” dan detail pesanannya	
8	Lihat Aturan Belanja	<ol style="list-style-type: none"> 1. Pembeli memilih halaman “Aturan Belanja 		Muncul halaman “Aturan Belanja”	

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan Tahapan Analisis, Perancangan dan Implementasi pada SM-Mart Client Aplikasi Berbasis SupermarkrtAndroid, maka dapat disimpulkan bahwa:

1. Aplikasi dapat menampilkan katalog yang diambil dari server.
2. Aplikasi dapat menangani proses pemesanan barang.
3. Aplikasi dapat menangani proses atur akun.
4. Aplikasi dapat menampilkan status pesanan.

5.2 Saran

Sebagai langkah pengembangan aplikasi ini pada masa yang akan datang, maka saran yang dapat diberikan adalah:

1. Untuk pengembang selanjutnya diharapkan menangani sekuritas sistem
2. Sebaiknya juga menangani proses pembayaran menggunakan sistem banking
3. Sebaiknya menambahkan fitur notifikasi jika terjadi perubahan status pesanan.
4. Tampilan sebaiknya dibuat lebih menarik.

DAFTAR PUSTAKA

- Fowler, Martin. "UML Distilled, 3th Ed., A Brief Guide to the Standard Object Modeling Language". Pearson Education, Inc, 2004.
- Henderi, Syafei. "Diktat Kuliah UML STMIK AMIKOM". Bandung: STMIK AMIKOM, 2007.
- Rainer & Cegielski, "Introduction To Information Systems-Supporting and Transforming Business". United States of America: Jhon Wiley & Sons, Inc, 2011.
- Safaat, Nazruddin."Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android". Bandung:Penerbit Informatika, 2012.
- Tasmawati, Mila."Aplikasi Konversi Reguler Grammer Menjadi Ekspresi Reguler Dengan Menggunakan Bahasa Pemrograman Java". Jakarta, 2008.
- Tim Penerjemah Penerbit Andi. "UML Distilled 3th Ed., Panduan Singkat Bahasa Pemodelan Objek Standar/Martin Fowler". Yogyakarta: Andi, 2005
- Tim Wahana Komputer. " Pengembangan Web di Linux dengan Apache, MySQL, dan PHP(LAMP)". Jakarta:Penerbit Salemba Infotek, 2005.

Peraturan Perundang-undangan

Permendag RI Nomor:53/M-DAG/PER/12/2008 BAB IV