BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis

Analisis bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang dibangun. Analisis tersebut meliputi analisis masalah, analisis kebutuhan data, analisis kebutuhan fungsional, dan analisis sistem.

3.1.1 Analisis Masalah

Permasalahan pada penelitian ini adalah masih banyak perusahaan yang tidak memanfaatkan efisiensi dari teknologi yang sudah tersedia pada saat ini. Perusahaan masih melakukan cara manual dalam menginformasikan produknya, baik pengumuman yang ditempel di tempat tertentu, menyebarkan brosur-brosur kepada setiap konsumen atau dengan menyewa *space* iklan yang memakan biaya cukup mahal yang menyebabkan penyebaran informasi yang kurang luas sehingga orang-orang kesulitan untuk mengakses informasi tersebut.

Serta bagaimana mengimplementasikan *M-Commerce* dengan android sebagai solusi permaslahan tersebut.

3.1.2 Analisis Sistem Yang Sedang Berjalan

Analisis sistem yang berjalan berisi tentang pemaparan prosedur-prosedur yang sedang berjalan saat ini. Analisis ini dimaksudkan agar perangkat lunak yang dibangun tidak keluar dari cakupan sistem yang ada. Adapun analisis dari sistem

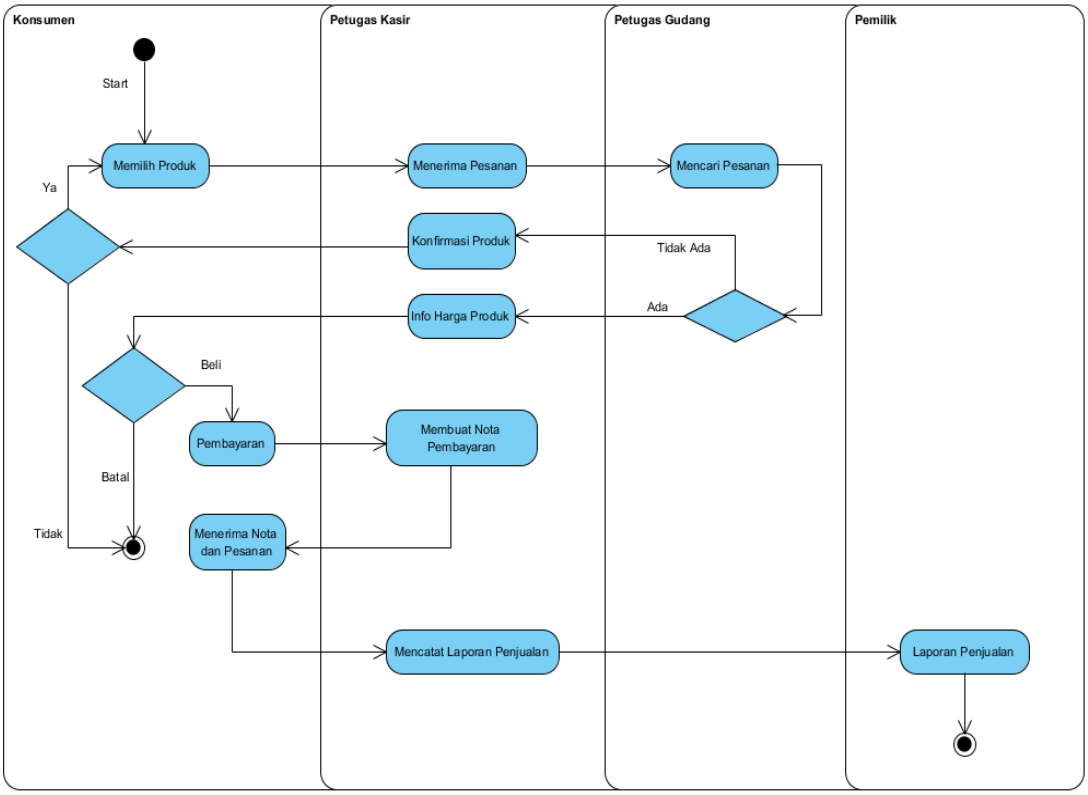
yang sedang berjalan saat ini terdiri dari analisis transaksi pada toko blackshoes dan analisis transaksi pada website.

3.1.2.1 Analisis Penjualan Pada Toko Blackshoes

Berdasakan hasil wawancara dengan pihak Blackshoes, prosedur transakasi penjualan yang sedang berjalan di toko Blackshoes adalah sebagai berikut :

1. Konsumen memilih data produk yang akan dibelinya. Setelah konsumen mendapatkan data produk yang akan dibelinya, konsumen memberikan data produk tersebut ke petugas kasir.
2. Petugas kasir menerima data produk yang akan dibeli oleh konsumen.
3. Petugas kasir memberi data produk tersebut kepada petugas gudang untuk melihat ketersediaan barang.
4. Petugas gudang mencari barang yang sesuai dengan data produk yang diberikan oleh petugas kasir. Jika barang ada petugas gudang akan memberikan barang tersebut kepada petugas toko. Jika barang tidak ada, maka petugas gudang akan memberikan konfirmasi kepada petugas kasir bahwa stok barang sudah habis lalu petugas kasir memberitahu konsumen bahwa barang tidak tersedia.
5. Petugas kasir memberi tahu kepada konsumen berapa total harga dari produk yang dibelinya.
6. Konsumen memberikan sejumlah uang ke petugas kasir sesuai dengan total harga dari produk yang dibelinya.

7. Petugas kasir akan mencetak nota pembayaran dan diberikan konsumen beserta produk yang dibeli.
8. Petugas kasir lalu mencatat laporan penjualan.
9. Pemilik menerima hasil pencatatan laporan penjualan

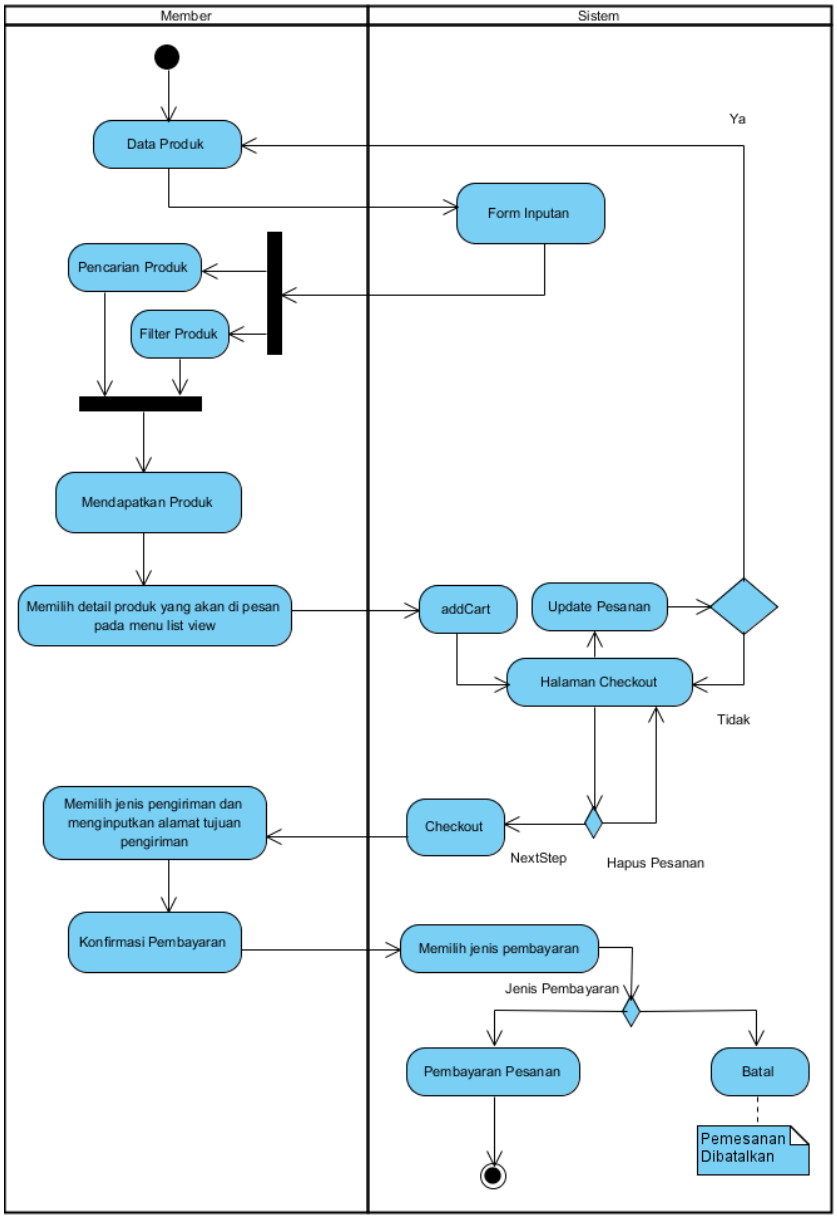


Gambar III.1 Sistem Yang Sedang Berjalan Pada Toko Blackshoes

3.1.2.2 Analisis Penjualan Pada *Website* Blackshoes

Berdasakan hasil wawancara dengan pihak yang membangun *website e-commerce* pada toko blackshoes, prosedur transaksi penjualan yang sedang berjalan di toko Blackshoes adalah sebagai berikut :

1. Member memilih data produk yang akan dibelinya.
2. Member mengisi form inputan untuk mendapatkan produk dengan menggunakan fitur yang telah disediakan, seperti filter produk dan pencarian produk.
3. Setelah mendapatkan produk yang diinginkan, member dapat melihat detail dari produk dari produk yang akan dipesan.
4. Setelah mendapatkan detail produk dari produk yang akan dipesan, member bisa memasukan produk yang akan dipesan ke dalam keranjang belanja.
5. Pada halaman *checkout*, member dapat melakukan update pesanan. Setelah selesai melakukan proses pemesanan member dapat melakukan proses *checkout*.
6. Pada halaman *shipping*, member diharuskan untuk memilih jenis pengiriman barang dan menginputkan kota tujuan pengiriman.
7. Setelah selesai mengiputkan kota tujuan pengiriman, member dapat melanjutkan ke proses konfirmasi pembayaran. Pada halaman konfirmasi pembayaran ditampilakan total pembayaran yang harus dibayar oleh member.
8. Setelah dikonfirmasi total pembayaran yang harus dibayar, member dapat melakukan pembayaran sesuai dengan jenis pembayaran yang telah disediakan oleh sistem.



Gambar III.2 Sistem Yang Sedang Berjalan Pada Website

3.1.3 Evaluasi Sistem Yang Sedang Berjalan

Berdasakan pada analisis dan hasil wawancara terhadap apa yang dibutuhkan dalam membangun aplikasi *m-commerce* ini maka dapat dievaluasi hal-hal seperti pada tabel berikut :

Tabel III.1 Evaluasi sistem yang sedang berjalan pada toko blackshoe

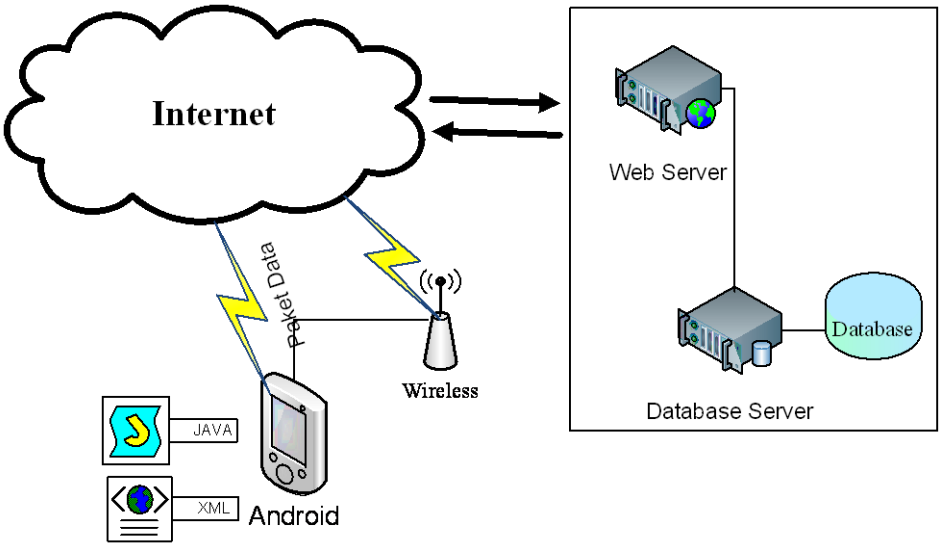
No	Permasalahan	Bagian/pihak
1	Proses pencarian dan pemilihan produk dilakukan di tempat mengharuskan konsumen datang ke tempat tujuan untuk mendapatkan produk.	Konsumen
2	Proses pembayaran yang dilakukan di tempat mengharuskan konsumen datang ke tempat tujuan untuk melakukan pembayaran.	Konsumen
3	Tidak adanya media penybaran informasi bagi toko Blackshoes.	Toko
4	Dengan sistem yang berjalan sekarang, pemberian informasi kepada komsumen dirasakan kurang efektif sehingga konsumen belum tentu mendapatkan informasi secara lengkap.	Konsumen

Tabel III.2 Evaluasi sistem yang sedang berjalan pada website

No	Permasalahan	Bagian/pihak
1	Proses transaksi pada website membutuhkan bandwidth yang besar.	Konsumen
2	Tidak adanya media penybaran informasi via mobile bagi toko Blackshoes.	Toko

3.1.4 Solusi Yang Ditawarkan

Berdasarkan hasil evaluasi sistem yang berjalan, solusi yang tawarkan adalah pembangunan aplikasi *m-commerce* di toko Blackshoes. Dalam sistem ini, konsumen dapat melihat informasi produk dan dapat melakukan pembelian secara online. Dengan sistem ini juga dapat menyelesaikan permasalahan dalam proses penyebarluasan informasi dan dapat memperluas pemasaran produk. Gambaran mengenai solusi yang akan ditawarkan dapat dilihat pada gambar III.3.



Gambar III.3 Solusi Yang Ditawarkan

3.1.5 Analisis Kebutuhan Data

Sistem yang akan dibangun merupakan sistem untuk membantu konsumen dalam mempermudah proses pembelian produk secara online melalui platform android. Dan juga untuk menginformasikan promo barang kepada konsumen yang membutuhkannya. Untuk proses pengolahan data dapat dilakukan di *website e-commerce*.

Http Connection digunakan sebagai penghubung antara aplikasi android ke *database* pada *web server*, sehingga *client* pada android dapat melakukan proses pemanggilan method atau fungsi untuk mendapatkan data produk yang ada di toko. Untuk pembangunan sistem ini, dibutuhkan data berupa informasi produk pada website yang di kelola oleh admin pada toko tersebut.

3.1.6 Analisis Fitur M-Commerce

Sistem yang dibangun memiliki fitur yang dirancang untuk meningkatkan penjualan, berikut ini adalah rincian fitur yang terdapat dalam *M-Commerce* Blackshoes yang dapat ditemukan dalam aplikasi.

1. Filter dan Pencarian Produk
2. Cek status pesanan
3. Notifikasi Produk Baru
4. Retur Produk
5. Sharing Produk
6. Map Direction

3.1.7 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Kebutuhan non fungsional yang dibutuhkan untuk membangun sistem ini terdiri dari tiga hal, yaitu : Kebutuhan pengguna, kebutuhan perangkat keras dan kebutuhan perangkat lunak.

3.1.7.1 Analisis Kebutuhan Pengguna

Sistem yang akan dibangun digunakan oleh dua jenis pengguna, yaitu member dan *user* biasa. Member dapat melakukan pencarian produk, pembelian dan pembayaran produk. Sedangkan *user* biasa hanya dapat melihat produk dan detail produk saja. Karakteristik pengguna di toko Blackshoes dapat dilihat pada tabel III.3.

Tabel III.3 Karakteristik Pengguna

Pengguna	Hak akses	Tingkat keterampilan
Member	Pembelian dan pembayaran produk	Mampu mengoperasikan smartphone android dan terkoneksi dengan internet.
Pengunjung	Terbatas	Mampu mengoperasikan smartphone android dan terkoneksi dengan internet.

3.1.7.2 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam membangun sistem ini adalah :

1. Processor Intel Core 2 Duo 3.0 GHz
2. Kapasitas RAM 4 GB
3. VGA *Card* 1GB DDR3
4. Kapasitas Hardisk 1 TB
5. Monitor
6. *Lan Card* 10/100 Mbps
7. Koneksi internet 1024 Kbps
8. *Keyboard*
9. *Mouse*

Spesifikasi kebutuhan minimal perangkat keras yang digunakan untuk membangun sistem ini adalah :

1. Processor dengan kecepatan minimal 2 GHz
2. RAM minimal 512 MB
3. VGA *Card* minimal 64 MB
4. Kapasitas Harddisk minimal 40 GB
5. Monitor
6. *Lan Card 10/100Mbps*
7. Koneksi internet minimal 64 kbps
8. *Keyboard*
9. *Mouse*

Spesifikasi kebutuhan perangkat keras (*hardware*) pada mobile android yang digunakan untuk menjalankan aplikasi mobile commerce ini adalah :

1. ARM Cortex A9 1GHz dual-core processor
2. Kapasitas RAM 512 MB
3. Memori internal 8 GB
4. GPU ULP GeForce
5. HSDPA, 7.2 Mbps; HSUPA, 5.76 Mbps
6. Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi hotspot
7. Ukuran layar 4.0 inches dengan resolusi 480 x 800 pixels
8. Standard battery, Li-Ion 1500 mAh

Spesifikasi kebutuhan minimal perangkat keras (*hardware*) pada mobile android yang digunakan untuk menjalankan aplikasi mobile commerce ini adalah :

1. Processor 600 MHz
2. Kapasitas RAM 256 MB
3. Memori internal 158 MB
4. HSDPA, 3.6 Mbps
5. Ukuran layar 3,3 inch dengan resolusi HVGA (240x320)
6. Baterai Li-Ion 1350 mAh

3.1.7.3 Analisis Kebutuhan Perangkat Lunak

Perangakat lunak yang diperlukan untuk membangun sistem ini adalah :

1. Java Development Kit versi 5 atau 6 , untuk kompilasi kode – kode program.
2. Java Runtime Edition, sebagai *platform* untuk menjalankan sistem.

3. Eclipse Galileo, sebagai perangkat lunak yang digunakan untuk membangun aplikasi *client* pada android.
4. Android SDK, untuk pengembang aplikasi android.
5. Android Development Tool, sebagai plugin android pada eclipse.
6. Android Virtual Device, sebagai emulator untuk menjalankan sistem android.
7. Sistem Operasi Windows 7 Ultimate
8. Android OS, v2.1 (*Eclair*)

3.1.7.4 Analisis Library

Library yang digunakan dalam membangun sistem ini adalah :

Tabel III.4 Library yang Digunakan

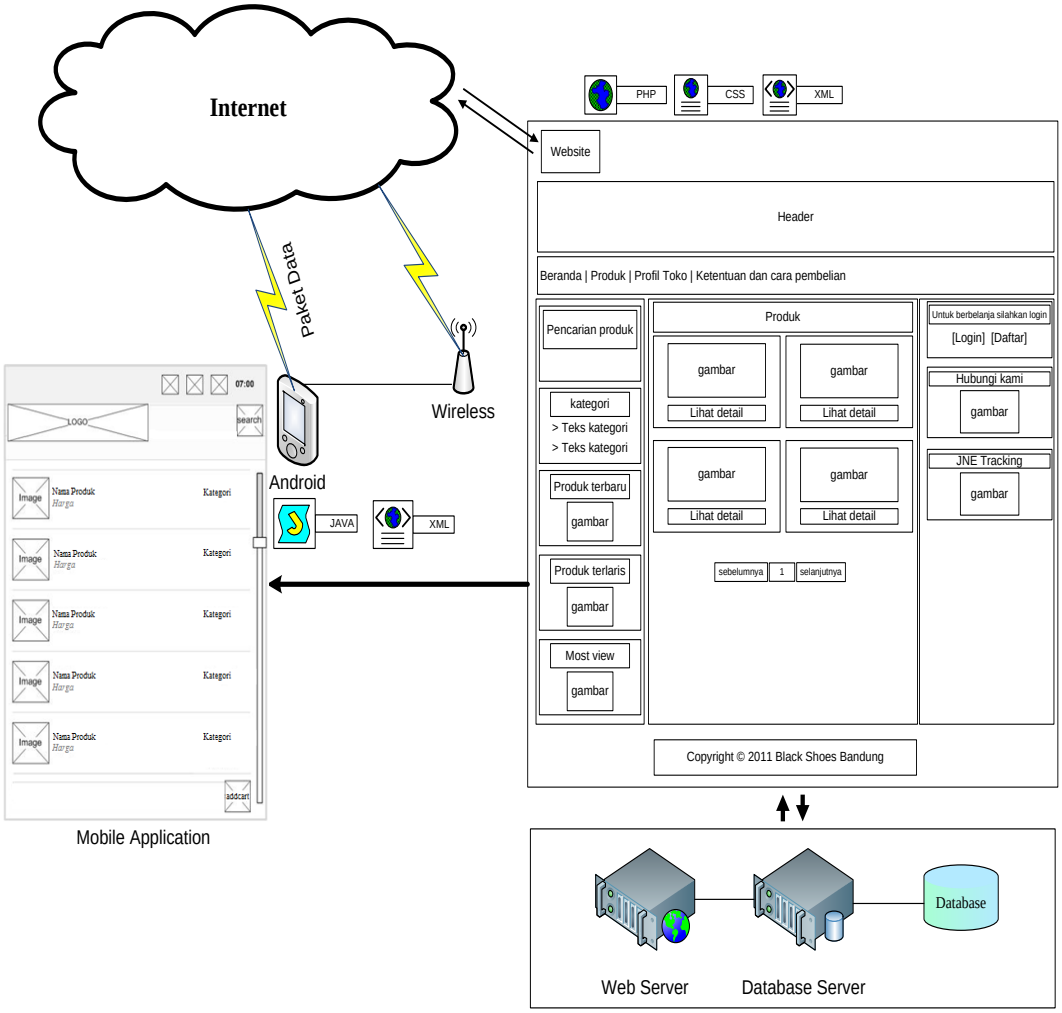
Library	Deskripsi
android.jar	*Default library* yang disediakan oleh android
maps.jar	*Library* yang digunakan untuk memanggil *maps* pada android
PayPal_MPL.jar	*Library* yang disediakan oleh paypal untuk *mobile developer* untuk mendapatkan hak akses pembayaran via *paypal*

3.1.8 Analisis Sistem

3.1.8.1 Arsitektur Sistem

Aplikasi utama pada *m-commerce* ini merupakan aplikasi *mobile* yang dapat digunakan oleh user melalui internet. *User* yang menggunakan aplikasi utama ini dapat dibedakan menjadi dua jenis, yaitu : pengunjung dan member. *Native user* diberikan hak akses hanya dapat mengakses fungsi-fungsi seperti pencarian produk, filter produk, dan pemilihan produk. Sedangkan yang membedakan antara pengunjung dan member, dimana member dapat melakukan pencarian produk, filter produk, pemilihan produk pemesanan produk dan pembayaran produk.

Selain aplikasi utama, *m-commerce* juga mempunyai dukungan *web server* sebagai pusat pengolahan data dalam bentuk *client – server*. Dengan menggunakan akses internet *client* dapat terhubung ke *database* pada *web server* dengan menggunakan *http Connection*. Pada penelitian ini program *client* akan dibuat pada platform android. Gambaran mengenai arsitektur sisitem yang akan dibangun dapat dilihat pada gambar III.4.



Gambar III.4 Arsitektur Sistem M-Commerce

3.1.8.2 Analisis Kebutuhan Fungsional

Dalam membangaun aplikasi *m-commerce* di toko blackshoes ini digunakan metode perancangan berorientasi objek. Pada dasarnya perancangan objek memberikan kemudahan dalam hal perawataan atai *maintenance* dan pengembangan sistem lebih lanjut.

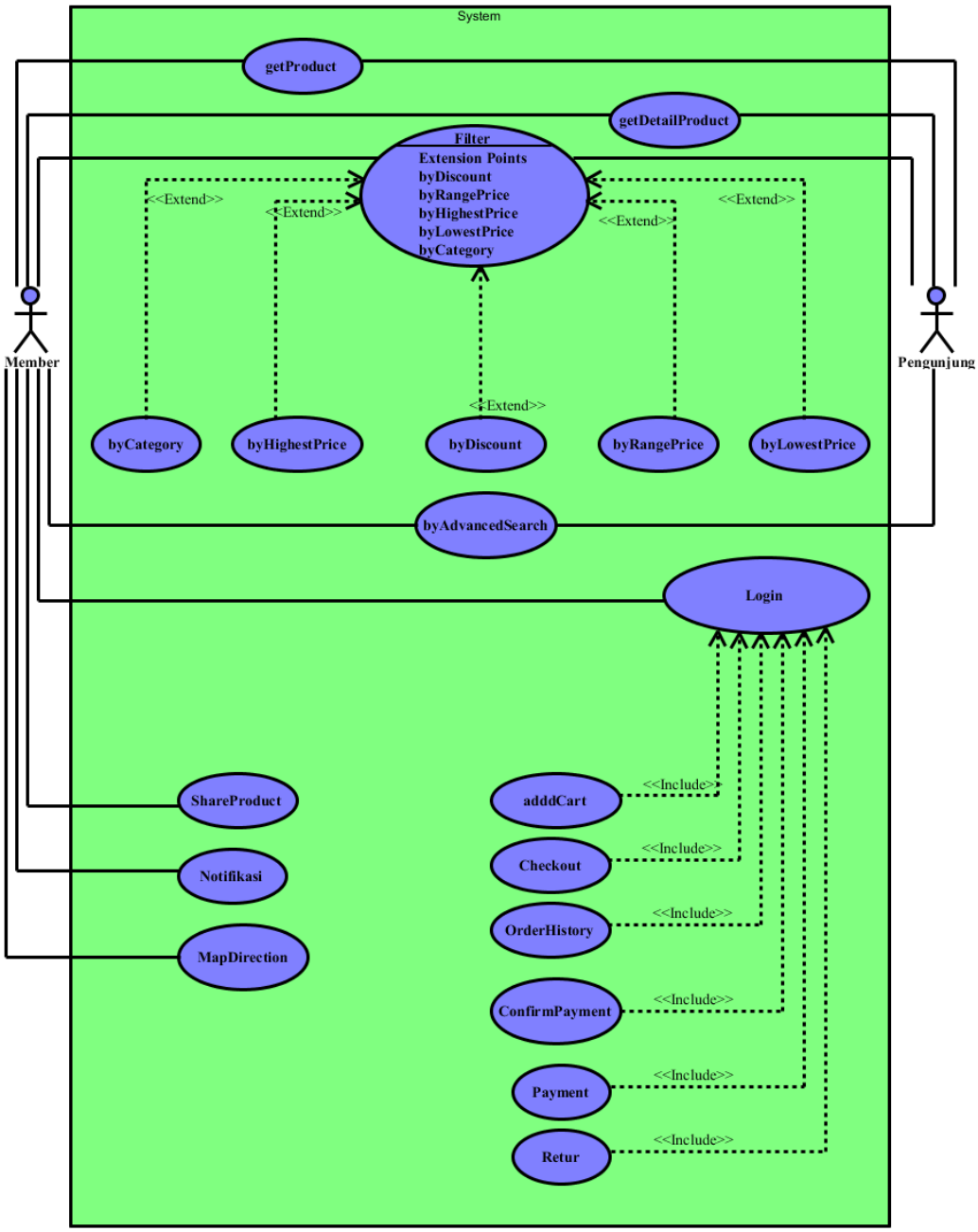
Selain itu, perancangan sistem aplikasi ini menggunakan metode perancangan berorientasi objek dengan menggunakan UML (*Unified Modeling Language*) untuk menentikan, visualisasi, konstruksi, dan mendokumentasikan informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan sistem.

Berdasarkan analisis yang telah dilakukan dalam pembuatan aplikasi ini, maka dapat dilihat apa yang menjadi (*input*), keluaran (*output*), metode yang digunakan, dan antar muka sistem yang dibuat, sehingga sistem yang dibuat sesuai yang diharapkan. Hasil perancangan berupa gambaran umum mengenai aplikasi secara keseluruhan yang akan menjelaskan proses implementasi dari sistem.

3.1.8.3 Analisis Sistem Pada Aplikasi M-Commerce

1. **U**

se Case Diagram



Gambar III.5 Use Case Diagram Sistem Pada Aplikasi M-Commerce

a. Identifikasi Aktor

Pada tabel III.5 berisi daftar aktor dan deskripsi peran meraka masing-masing dalam aplikasi ini.

Tabel III.5 Definisi Aktor pada aplikasi android.

NO	Aktor	Deskripsi
A-01	Pengunjung	Merupakan pengguna dari aplikasi *M-Commerce* pada android yang dibatasi hak aksesnya.
A-02	Member	Merupakan pengguna dari aplikasi *M-Commerce* yang dapat melakukan pemesanan pada android.

b. Identifikasi Use Case

Berdasarkan use case diagram di atas maka di buatlah skenario dari tiap proses yang ada pada use case diagram tersebut. Pada tabel III.6 menggambarkan skenario *login* pada use case *login*.

Tabel III.6 Skenario use case login.

Identifikasi	
Nomor	UC-AN-01
Nama Use Case	*Login*
Deskripsi	Proses autentifikasi user untuk masuk pada sistem
Aktor	Member
Skenario Utama (UC-AN-01-S01)	
Kondisi awal	Koneksi ke *database* berhasil dan tampil halaman *login*
Aksi aktor	**Respon Sistem**
1. Membuka halaman utama	2. Menampilkan form *login*
3. Memasukan *username* dan	4. Mencocokan *username* dan *password* yang

password	ada pada *database*
	5. Log in berhasil
Skenario Alternatif (UC-AN-01-S01) *Login* namun terdapat pesan error	
	5a. Menampilkan pesan kesalahan, missal karena *username* dan *password* tidak ada pada *database* atau *username* dan *password* tidak cocok
Kondisi akhir	Halaman member berhasil ditampilkan di android

Pada tabel III.7 menggambarkan skenario memilih produk pada use case *getProduct*.

Tabel III.7 Skenario Use case getProduct

Identifikasi	
Nomor	UC-AN-02
Nama Use Case	*getProduct*
Deskripsi	Proses untuk memilih data produk yang ada
Aktor	Pengunjung, Member
Skenario Utama(UC-AN-02-S01)	
Kondisi awal	Koneksi ke *web server* berhasil
Aksi aktor User	**Respon Sistem**
1. Menjalankan menu utama	2. Memanggil fungsi *getProduct*
	3. Mengambil *return value* dari *web server* dalam bentuk JSON
	4. Melakukan parsing data
	5. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-02-S02) *getProduct* namun terdapat pesan error	
	2a. Menampilkan pesan kesalahan, misalnya karena tidak ada koneksi internet atau tidak dapat terhubung ke *server*
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.8 menggambarkan skenario memilih produk pada use case *getDetailProduct*.

Tabel III.8 Skenario use case getDetailProduct.

Identifikasi	
Nomor	UC-AN-03
Nama Use Case	*getDetailProduct*
Deskripsi	Proses untuk mengambil data produk berdasarkan id produk
Aktor	Pengunjung, Member
Skenario Utama (UC-AN-03-S01)	
Kondisi awal	Koneksi ke *web server* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih produk pada *listView*	2. Menggambil parameter id produk
	3. Memanggil fungsi *getProductById*
	4. Mengambil *return value* dari *web server* dalam bentuk JSON
	5. Melakukan parsing data
	6. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-03-S02*) getProductByID* namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena tidak ada koneksi internet atau tidak dapat terhubung ke *server*
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.9 menggambarkan skenario filter data produk berdasarkan kategori pada use case *byCategory*.

Tabel III.9 Skenario use case byCategory

Identifikasi	
Nomor	UC-AN-04
Nama Use Case	*byCategory*
Deskripsi	Proses untuk memfilter data produk yang akan di tampilkan berdasarkan kategori produk
Aktor	Pengunjung, Member
Skenario Utama(UC-AN-04-S01)	
Kondisi awal	Koneksi ke *web server* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu filter produk berdasarkan kategori	2. Menampilkan form inputan
3. Menginputkan nama produk berdasarkan kategori	4. Memanggil fungsi *byCategory*
	5. Mengambil *return value* dari *web server* dalam bentuk JSON
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif(UC-AN-04-S02) *byCategory* namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena input dari user salah(skenario kembali ke langkah 3)
Skenario Alternatif(UC-AN-04-S03) *byCategory* namun data tidak ditemukan	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada

Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.10 menggambarkan skenario filter data produk berdasarkan produk baru pada use case *byRangePrice*.

Tabel III.10 Skenario use case byRangePrice

Identifikasi	
Nomor	UC-AN-05
Nama Use Case	*byRangePrice*
Deskripsi	Proses untuk memfilter data produk yang akan di tampilkan berdaskan *range* harga
Aktor	Pengunjung, Member
Skenario Utama (UC-AN-05-S01)	
Kondisi awal	Koneksi ke *web server* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu filter produk berdasarkan *range* harga	2. Menampilkan form filter produk
3. Memilih produk baru yang akan ditampilkan	4. Memanggil fungsi *byRangePrice*
	5. Mengambil *return value* dari *web server* dalam bentuk JSON
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-05-S02) *byRangePrice* namun terdapat pesan error	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.11 menggambarkan skenario filter data produk berdasarkan diskon tertentu pada use case *byDiscount*.

Tabel III.11 Skenario use case byDiscount

Identifikasi	
Nomor	UC-AN-06
Nama Use Case	*byDiscount*
Deskripsi	Proses untuk menfilter data produk yang akan di tampilkan berdaskan diskon
Aktor	Pengunjung, Member
Skenario Utama(UC-AN-06-S01)	
Kondisi awal	Koneksi ke *web server* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu filter produk berdasarkan diskon	2. Menampilkan form inputan
3. Menginputkan diskon produk.	4. Memanggil fungsi *byDiscount*
	5. Mengambil *return value* dari *web server* dalam bentuk JSON
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif(UC-AN-06-S02) *byDiscount* namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena input dari user salah(skenario kembali ke langkah 3)
Skenario Alternatif(UC-AN-06-S03) *byDiscount* namun data tidak ditemukan	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.12 menggambarkan skenario filter data produk berdasarkan harga tertinggi produk pada use case *byHigestPrice*.

Tabel III.12 Skenario use case byHigestPrice

Identifikasi	
Nomor	UC-AN-07
Nama Use Case	*byHigestPrice*
Deskripsi	Proses untuk memfilter data produk yang akan di tampilkan berdaskan harga tertinggi
Aktor	Pengunjung, Member
Skenario Utama (UC-AN-07-S01)	
Kondisi awal	Koneksi ke *web server* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu filter produk berdasarkan harga tertinggi.	2. Menampilkan filter produk
3. Memilih harga tertinggi yang akan ditampilkan	4. Memanggil fungsi *byHigestPrice*
	5. Mengambil *return value* dari *web server* dalam bentuk JSON
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-07-S02) *byHigestPrice* namun terdapat pesan error	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.13 menggambarkan skenario filter data produk berdasarkan harga terendah produk pada use case *byLowestPrice*.

Tabel III.13 Skenario use case byLowestPrice

Identifikasi	
Nomor	UC-AN-08
Nama Use Case	*byLowestPrice*
Deskripsi	Proses untuk memfilter data produk yang akan di tampilkan berdaskan harga terendah
Aktor	Pengunjung, Member
Skenario Utama (UC-AN-08-S01)	
Kondisi awal	Koneksi ke *web server* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu filter produk berdasarkan harga terendah.	2. Menampilkan form filter produk
3. Memilih harga terendah yang akan ditampilkan	4. Memanggil fungsi *byLowestPrice*
	5. Mengambil *return value* dari *web server* dalam bentuk JSON
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-08-S02) *byLowestPrice* namun terdapat pesan error	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.14 menggambarkan skenario pencarian data produk berdasarkan pencarian advance tertentu pada use case *byAdvancedSearch*.

Tabel III.14 Skenario use case byAdvancedSearch

Identifikasi	
Nomor	UC-AN-9
Nama Use Case	*byAdvancedSearch*
Deskripsi	Proses untuk pencarian data produk yang akan di tampilkan berdaskan pencarian advance.
Aktor	Pengunjung, Member
Skenario Utama(UC-AN-9-S01)	
Kondisi awal	Koneksi ke *web server* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu pencarian produk berdasarkan pencarian advance.	2. Menampilkan form inputan
3. Menginputkan diskon produk.	4. Memanggil fungsi by*AdvancedSearch*
	5. Mengambil *return value* dari *web server* dalam bentuk JSON
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif(UC-AN-9-S02) *byAdvancedSearch* namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena input dari user salah(skenario kembali ke langkah 3)
Skenario Alternatif(UC-AN-9-S03) *byAdvancedSearch* namun data tidak ditemukan	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada.

Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel III.15 menggambarkan skenario untuk *addCart* produk pada usecase *addCart*.

Tabel III.15 Skenario use case addCart

Identifikasi	
Nomor	UC-AN-10
Nama Use Case	*addCart*
Deskripsi	Proses untuk memasukan produk kedalam keranjang belanja
Aktor	Member
Skenario Utama(UC-AN-10-S01)	
Kondisi awal	Berada pada halaman *Detail Product*
Aksi aktor	**Respon Sistem**
1. Memilih menu *addCart*	2. Menyimpan pesanan kedalam keranjang belanja
Kondisi akhir	Data pesanan berhasil di simpan ke dalam keranjang belanja

Pada tabel III.16 menggambarkan skenario untuk *Checkout* produk pada usecase *Checkout*.

Tabel III.16 Skenario use case Checkout

Identifikasi	
Nomor	UC-AN-11
Nama Use Case	*Checkout*
Deskripsi	Proses pemeriksaan produk yang dipesan dan proses pengiriman produk ke alamat tujuan pengiriman
Aktor	Member
Skenario Utama(UC-AN-11-S01)	
Kondisi awal	Berada pada halaman utama

Aksi aktor	**Respon Sistem**
1. Memilih produk 3. Memilih menu *addCart*	2. Menampilkan detail produk 4. Menampilkan halaman *Checkout*
5. Update pesanan	
6. Memilih *Checkout*	7. Proses *Checkout*
8. Menginpukan alamat tujuan pengiriman	9. Proses *Shipping*
Skenario Alternatif(UC-AN-11-S03) *Checkout* pada proses update pesanan	
	5.a kembali ke proses pemilihan produk yang akan di tambahkan ke dalam keranjang belanjaan
Kondisi akhir	Data berhasil dipesan

Pada table III.17 menggambarkan skenario untuk *ConfirmPayment* produk pada usecase *ConfirmPayment*.

Tabel III.17 Skenario use case ConfirmPayment

Identifikasi	
Nomor	UC-AN-12
Nama Use Case	*ConfirmPayment*
Deskripsi	Proses untuk konfirmasi pembayaran
Aktor	Member
Skenario Utama(UC-AN-12-S01)	
Kondisi awal	Berada pada halaman *ConfirmPayment*
Aksi aktor	**Respon Sistem**
1. Konfirmasi total pembayaran	2. Memproses konfirmasi pembayaran
Kondisi akhir	Pesanan berhasil dikonfirmasi

Pada tabel III.18 menggambarkan skenario untuk *Payment* produk pada usecase *Payment*.

Tabel III.18 Skenario use case Payment

Identifikasi	
Nomor	UC-AN-13
Nama Use Case	*Payment*
Deskripsi	Proses pembayaran produk
Aktor	Member
Skenario Utama(UC-AN-13-S01)	
Kondisi awal	Berada pada halaman *Payment*
Aksi aktor	**Respon Sistem**
1. Melakukan pembayaran produk	2. Pembayaran via *paypal*
	3. Menampilkan *login paypal*
4. *Login paypal*	
5. *Submit payment*	6. *Payment process*
Kondisi akhir	Pesanan berhasil dibayar

Pada tabel III.19 menggambarkan skenario untuk *OrderHistory* produk pada usecase *OrderHistory*.

Tabel III.19 Skenario use case OrderHistory

Identifikasi	
Nomor	UC-AN-14
Nama Use Case	*OrderHistory*
Deskripsi	Proses untuk melihat semua transaksi yang telah dilakukan oleh member
Aktor	Member

Skenario Utama(UC-AN-14-S01)	
Kondisi awal	Berada pada halaman utama
Aksi aktor	**Respon Sistem**
1. Memilih menu *Purchase List* pada android	2. Menampilkan halaman *Purchase List* android
3. Memilih *order history*	4. Menampilkan *order history*
Kondisi akhir	History pemesanan berhasil ditampilkan

Pada tabel III.20 menggambarkan skenario untuk retur produk pada usecase Retur.

Tabel III.20 Skenario use case Retur

Identifikasi	
Nomor	UC-AN-15
Nama Use Case	Retur
Deskripsi	Proses pengembalian barang dari customer
Aktor	Member
Skenario Utama(UC-AN-15-S01)	
Kondisi awal	Berada pada *OrderHistory*
Aksi aktor	**Respon Sistem**
1. Memilih menu Retur pada android	2. Menampilkan halaman Retur android
	3. Menampilkan Retur
Kondisi akhir	Retur Produk berhasil ditampilkan

Pada tabel III.21 menggambarkan skenario untuk *share* produk melalui *facebook* pada use case *shareProduk.*

Tabel III.21 Skenario use case shareProduct

Identifikasi	
Nomor	UC-AN-16
Nama Use Case	*shareProduct*
Deskripsi	Proses untuk membagikan produk melalui facebook
Aktor	Pengunjung, Member
Skenario Utama(UC-AN-16-S01)	
Kondisi awal	Berada pada halaman utama
Aksi aktor	**Respon Sistem**
1. Memilih menu *share*	2. Mengambil data produk berupa nama,tgl,dan lokasi
	3. Login *facebook*
	4. Menampilkan pesan produk berhasil di *posting*
Kondisi akhir	Data produk berhasil di posting di facebook

Pada tabel III.22 menggambarkan skenario untuk Notifikasi produk pada use case Notifikasi.

Tabel III.22 Skenario use case Notifikasi

Identifikasi	
Nomor	UC-AN-17
Nama Use Case	Notifikasi
Deskripsi	Proses pemberitahuan informasi ter-*update*
Aktor	Pengunjung, Member
Skenario Utama(UC-AN-17-S01)	
Kondisi awal	*Mobile* android terhubung dengan jaringan internet

Aksi aktor	**Respon Sistem**
	5. Mengirimkan notifikasi produk
6. Membuka notifikasi	7. Menampilkan produk ter-*update*
Kondisi akhir	Data produk berhasil di tampilkan

Pada tabel III.23 menggambarkan skenario untuk *Map Dirirection* produk pada use case *MapDirection*.

Tabel III.23 Skenario use case MapDirection

Identifikasi	
Nomor	UC-AN-18
Nama Use Case	*MapDirection*
Deskripsi	Proses pemberitahuan informasi menuju toko balckshoes
Aktor	Pengunjung, Member
Skenario Utama(UC-AN-18-S01)	
Kondisi awal	*Mobile* android terhubung dengan jaringan internet
Aksi aktor	**Respon Sistem**
1. Memilih MapDirection	2. Mengambil lokasi pengguna dengan bantuan GPS
	3. Menunjukan lokasi jalan menuju toko
Kondisi akhir	Data produk berhasil di tampilkan

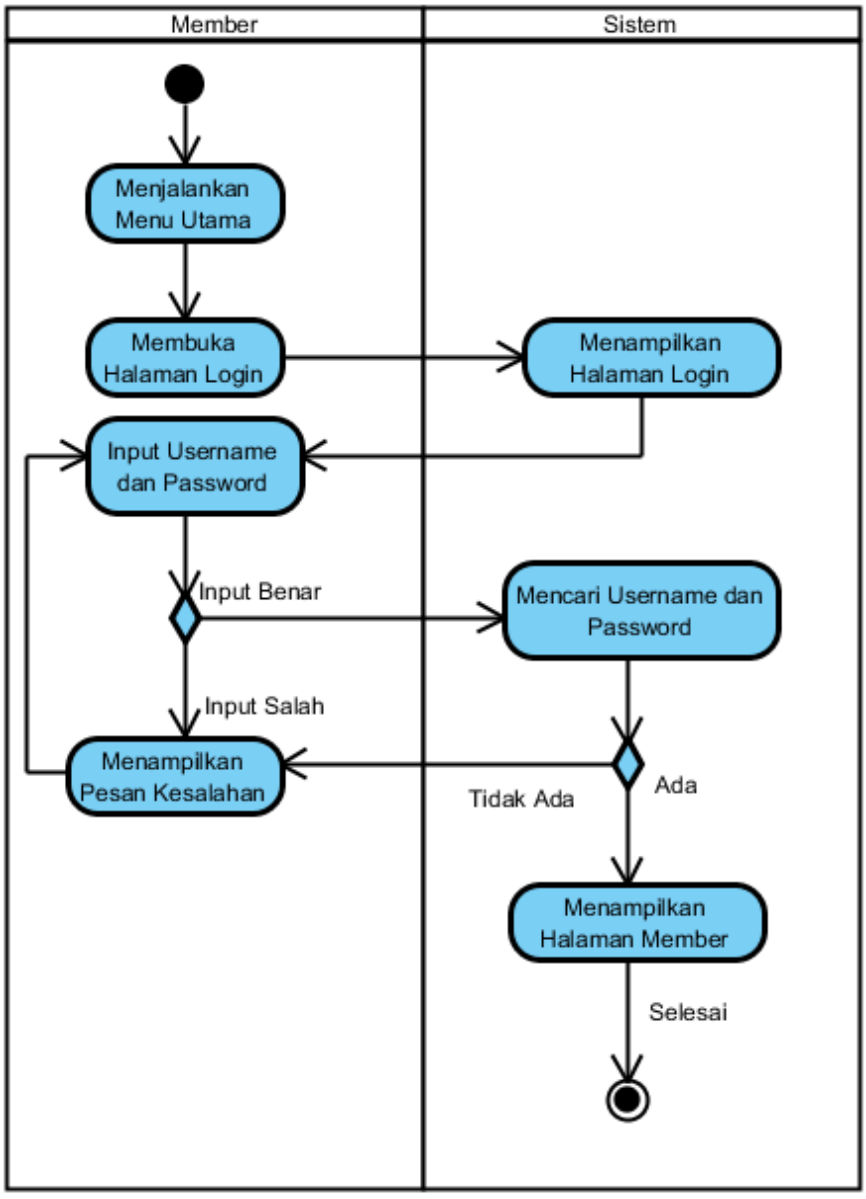
2. **Activity Diagram**

a. Activity Diagram *login*

Aktifitas *login* merupakan aktifitas yang dilakukan member untuk mandapatkan akses untuk melakukan proses pembelian produk. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Membuka halaman member, pada aktifitas ini *client* membuka halaman untuk mengakses halaman member.
2. Menampilkan *form login*, pada aktifitas ini sistem menampilkan *form login* yang harus di isi oleh *member*.
3. Mengisi *form login*, pada aktifitasi ini *member* mengisi *form* yang berupa *email* dan *password*.
4. Mencari *username* dan *password*, pada aktifitas ini sistem mencocokan *username* dan *password* yang *member* inputkan dengan *username* dan *password* yang ada pada *database*.
5. Menampilkan pesan kesalahan, aktifitas ini terjadi jika *username* dan *password* yang *member* inputkan tidak ada pada *database*.
6. Menampilkan halaman member, aktifitas ini terjadi jika username dan password yang member inputkan ada pada database lalu sistem menampilkan halaman member, dimana member dapat melakukan pemesanan dan pembelian produk.

Seluruh aktifitas *login* terlihat pada *activity diagram* gambar III.6.



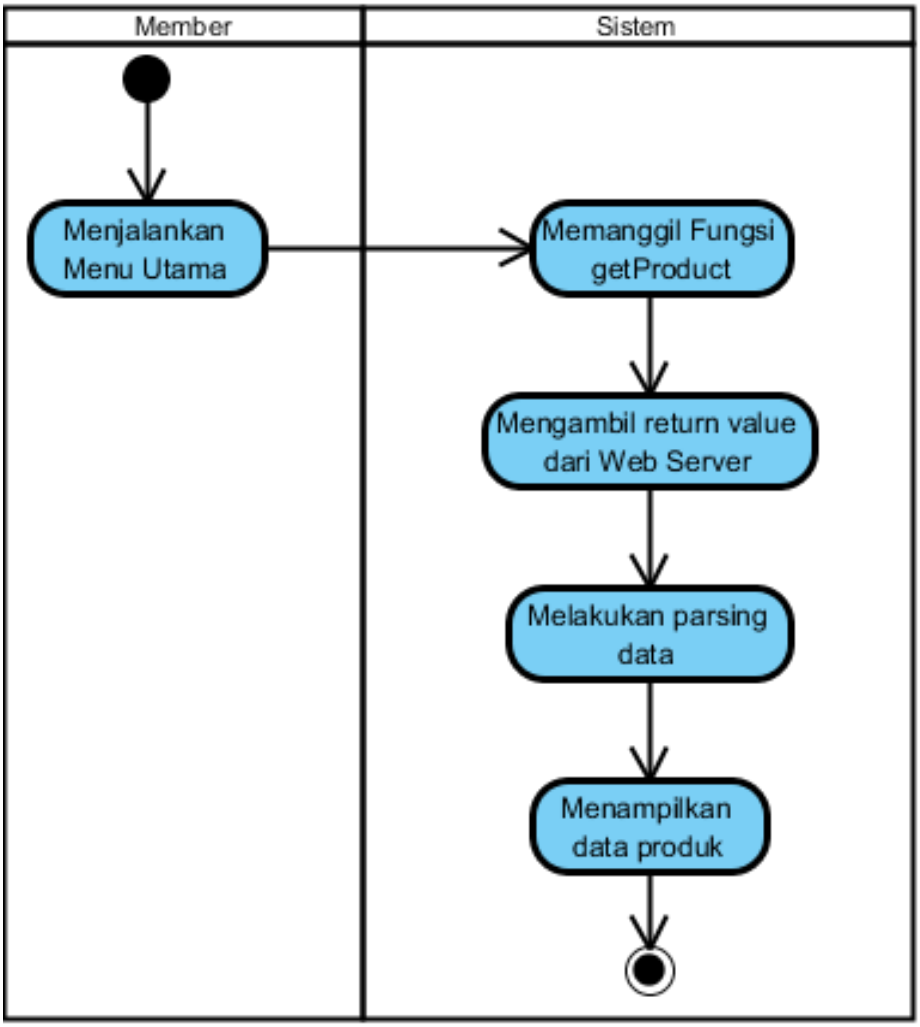
Gambar III.6 Activity Diagram Login

b. Activity Diagram *GetProduct*

Aktifitas *getProduct* merupakan aktifitas yang dilakukan oleh member untuk mangambil data produk yang ada pada *database* dengan *web server*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Menjalankan menu utama, pada aktifitas ini user menjalankan menu utama yang ada pada sistem.
2. Memanggil fungsi *getProduct*, pada aktifitas ini sistem memanggil fungsi *getProduct* yang ada pada *web server*.
3. Mengambil *return value* dari *web server*, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh *web server* berupa hasil dari eksekusi *method* pada *web server*.
4. Melakukan *parsing* data, pada aktifitas ini sistem memparsing respon dari *web server* agar data yang diterima dapat dibaca.
5. Menampilkan data produk, pada aktifitas ini sistem menampilkan hasil *parsing* kedalam bentuk *list view*.

Seluruh aktifitas *getProduct* terlihat pada *activity diagram* gambar III.7.



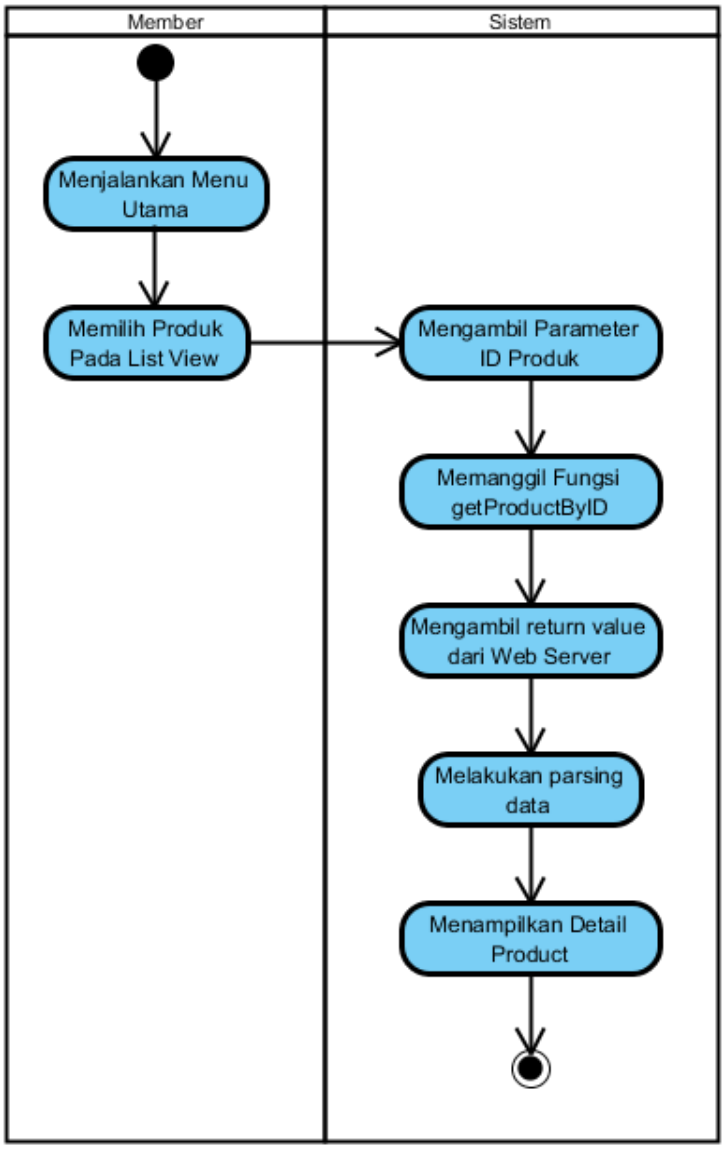
Gambar III.7 Activity Diagram GetProduct

c. ActivityDiagram *GetDetailProduct*

Aktifitas *getDetailProduct* merupakan aktifitas yang dilakukan oleh member untuk mangambil detail dari data produk yang ada pada *web server* berdasarkan id produk. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Memilih produk pada *listview*, pada aktifitas ini user memilih produk untuk menampilkan detail dari produk yang dipilih..
2. Mengambil parameter id produk, pada aktifitas ini sistem mengambil id produk, dimana id produk didapat ketika user memilih produk sebelumnya pada *list view*.
3. Memanggil fungsi *getProductByID*, pada aktifitas ini sistem memanggil fungsi *getProductByID* yang ada pada *web server*.
4. Mengambil *return value* dari *web server*, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh *web server* berupa hasil dari eksekusi *method* pada *web server*.
5. Melakukan *parsing* data, pada aktifitas ini sistem memparsing respon dari *web server* agar data yang diterima dapat dibaca.
6. Menampilkan data Produk, pada aktifitas ini sistem menampilkan hasil parsing.

Seluruh aktifitas *getDetailProduct* terlihat pada *activity diagram* gambar III.8.



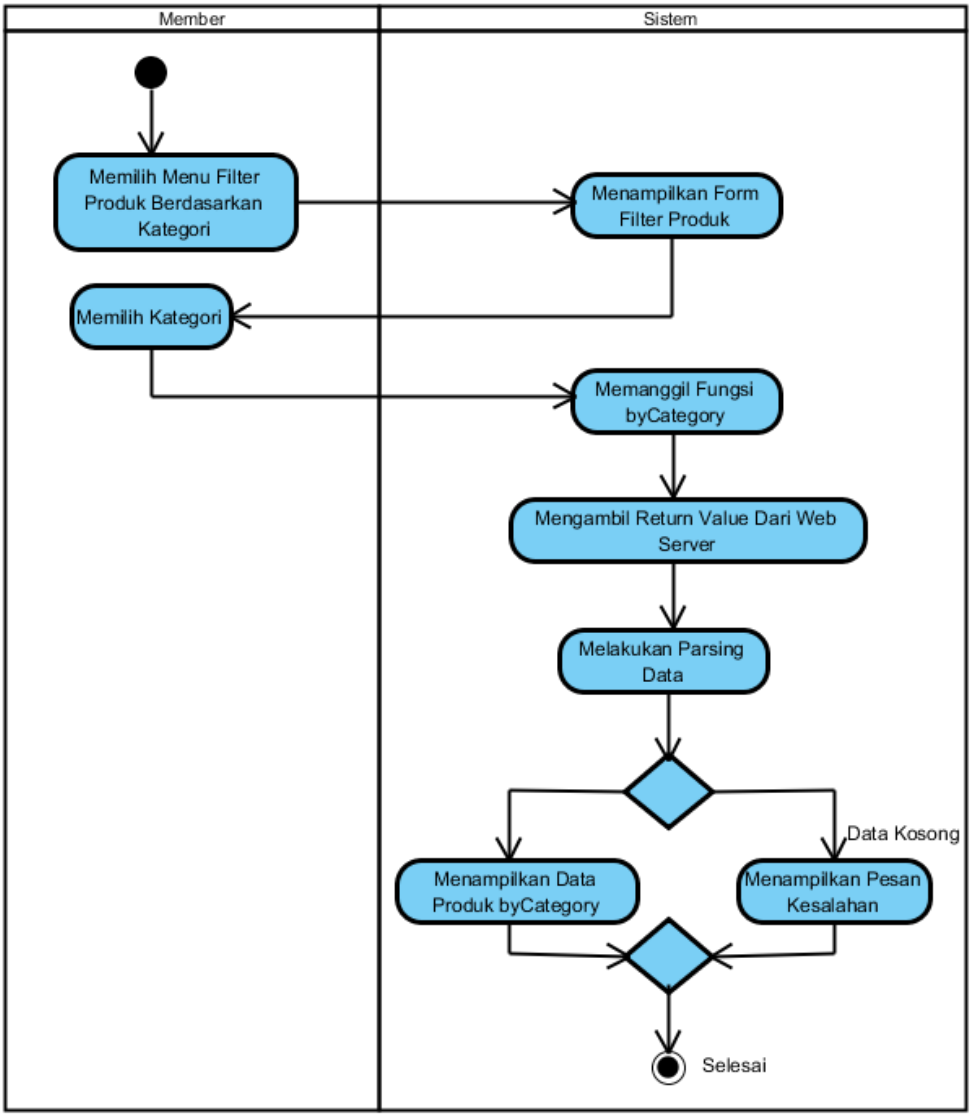
Gambar III.8 Activity Diagram GetDetailProduct

d. Activity Diagram *GetProductByCategory*

Aktifitas get*ProductByCategory* merupakan aktifitas yang dilakukan oleh member untuk memfilter data produk berdasarkan kategori produk yang ada pada *web server*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Memilih menu filter produk berdasarkan kategori, pada aktifitas ini member memilih menu untuk memfilter data berdasarkan kategori produk.
2. Memilih kategori, pada aktifitas ini user memilih kategori produk untuk memfilter data produk.
3. memanggil fungsi *byCategory*, pada aktifitas ini sistem memanggil fungsi *byCategory* yang ada pada *web server*.
4. Mengambil *return value* dari *web server*, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh *web server* berupa hasil dari eksekusi *method* pada *web server*.
5. Melakukan *parsing* data, pada aktifitas ini sistem memparsing respon dari *web server* agar data yang diterima dapat dibaca.
6. Menampilkan data produk, pada aktifitas ini sistem menampilkan hasil *parsing* kedalam bentuk *list view* jika data hasil *parsing* tidak kosong.
7. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas get*ProductByCategory* terlihat pada *activity diagram* gambar III.9.



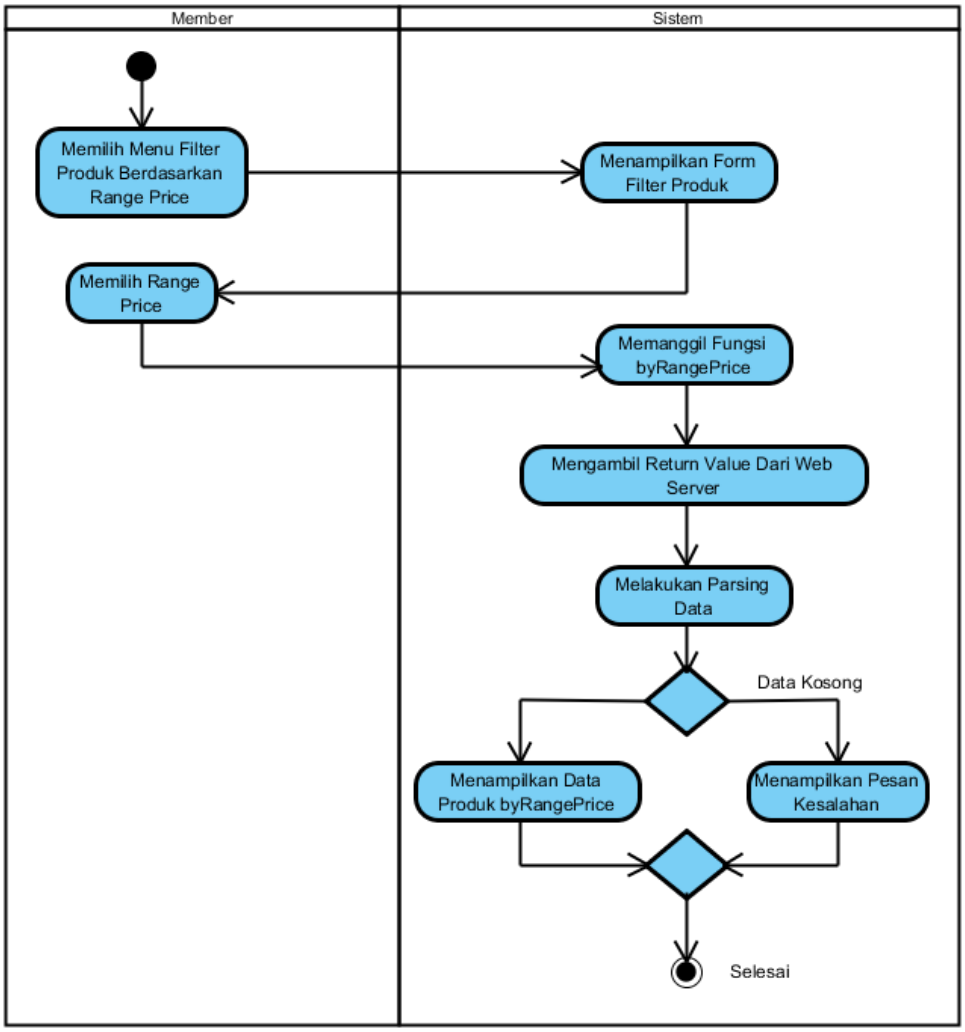
Gambar III.9 Activity Diagram GetProductByCategory

e. Activity Diagram *GetProductByRangePrice*

Aktifitas get*ProductByRangePricet* merupakan aktifitas yang dilakukan oleh member untuk memfilter data produk berdasarkan *range* harga yang ada pada *web server*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Memilih menu filter produk berdasarkan *range price*, pada aktifitas ini user memilih menu untuk memfilter data berdasarkan *range* harga.
2. Mnginputkan *range price*, pada aktifitas ini user menginputkan jumlah *range price* untuk memfilter data produk.
3. memanggil fungsi *byRangePrice*, pada aktifitas ini sistem memanggil fungsi *byRangePrice* yang ada pada *web server*.
4. Mengambil *return value* dari *web server*, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh *web server* berupa hasil dari eksekusi *method* pada *web server*.
5. Melakukan *parsing* data, pada aktifitas ini sistem memparsing respon dari *web server* agar data yang diterima dapat dibaca.
6. Menampilkan data produk, pada aktifitas ini sistem menampilkan hasil *parsing* kedalam bentuk *list view* jika data hasil parsing tidak kosong.
7. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas get*ProductByRangePricet* terlihat pada *activity diagram* gambar III.10.



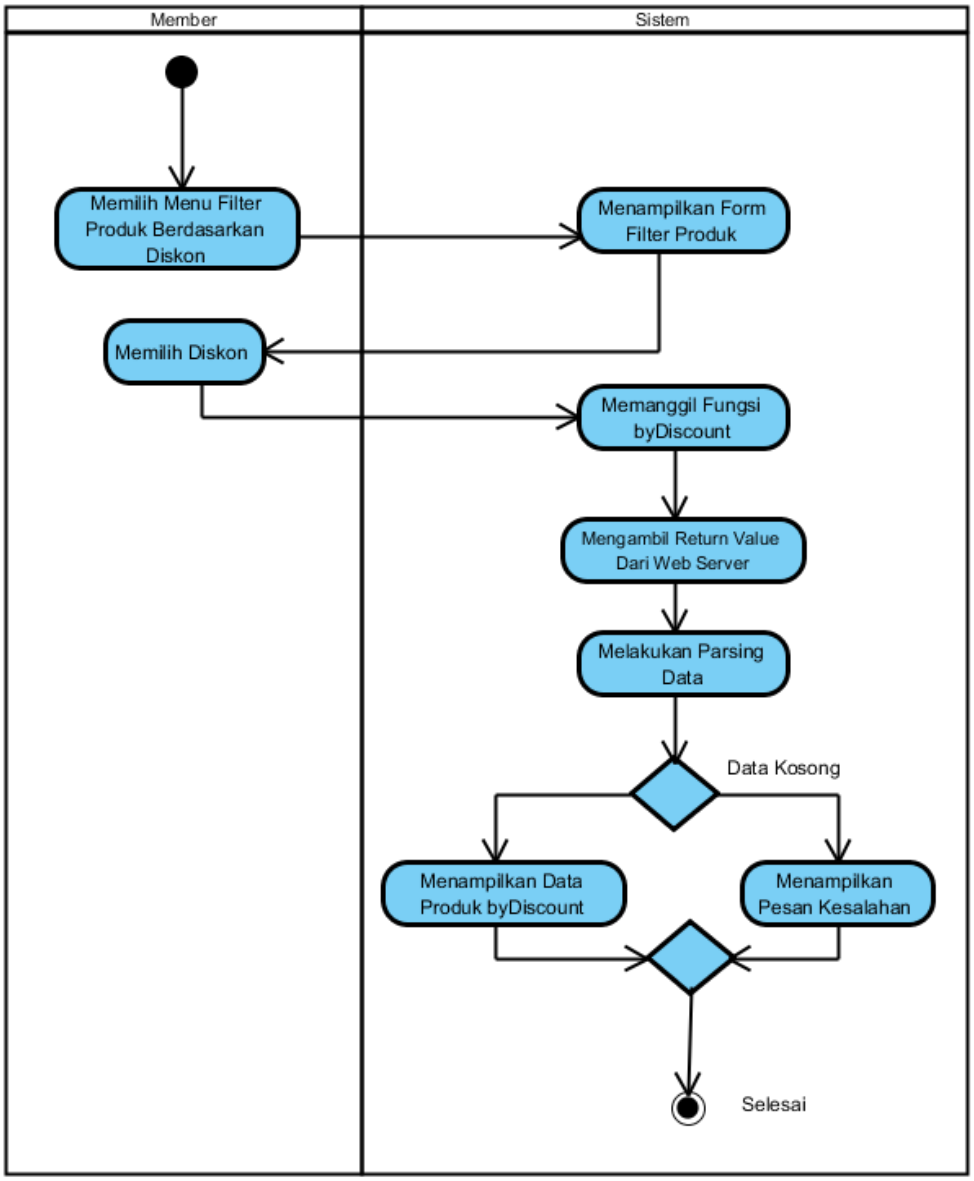
Gambar III.10 Activity Diagram GetProductByRangePrice

f. Activity Diagram *GetProductByDiscount*

Aktifitas get*ProductByDiscount* merupakan aktifitas yang dilakukan oleh member untuk memfilter data produk berdasarkan produk baru yang ada pada *web server*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Memilih menu filter produk berdasarkan diskon, pada aktifitas ini user memilih menu untuk memfilter data berdasarkan diskon produk.
2. Memilih diskon, pada aktifitas ini user memilih diskon produk untuk memfilter data produk.
3. memanggil fungsi *byDiscount*, pada aktifitas ini sistem memanggil fungsi *byDiscount* yang ada pada *web server*.
4. Mengambil *return value* dari *web server*, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh *web server* berupa hasil dari eksekusi *method* pada *web server*.
5. Melakukan *parsing* data, pada aktifitas ini sistem memparsing respon dari *web server* agar data yang diterima dapat dibaca.
6. Menampilkan data produk, pada aktifitas ini sistem menampilkan hasil *parsing* kedalam bentuk *list view* jika data hasil parsing tidak kosong.
7. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas get*ProductByDiscount* terlihat pada *activity diagram* gambar III.11.



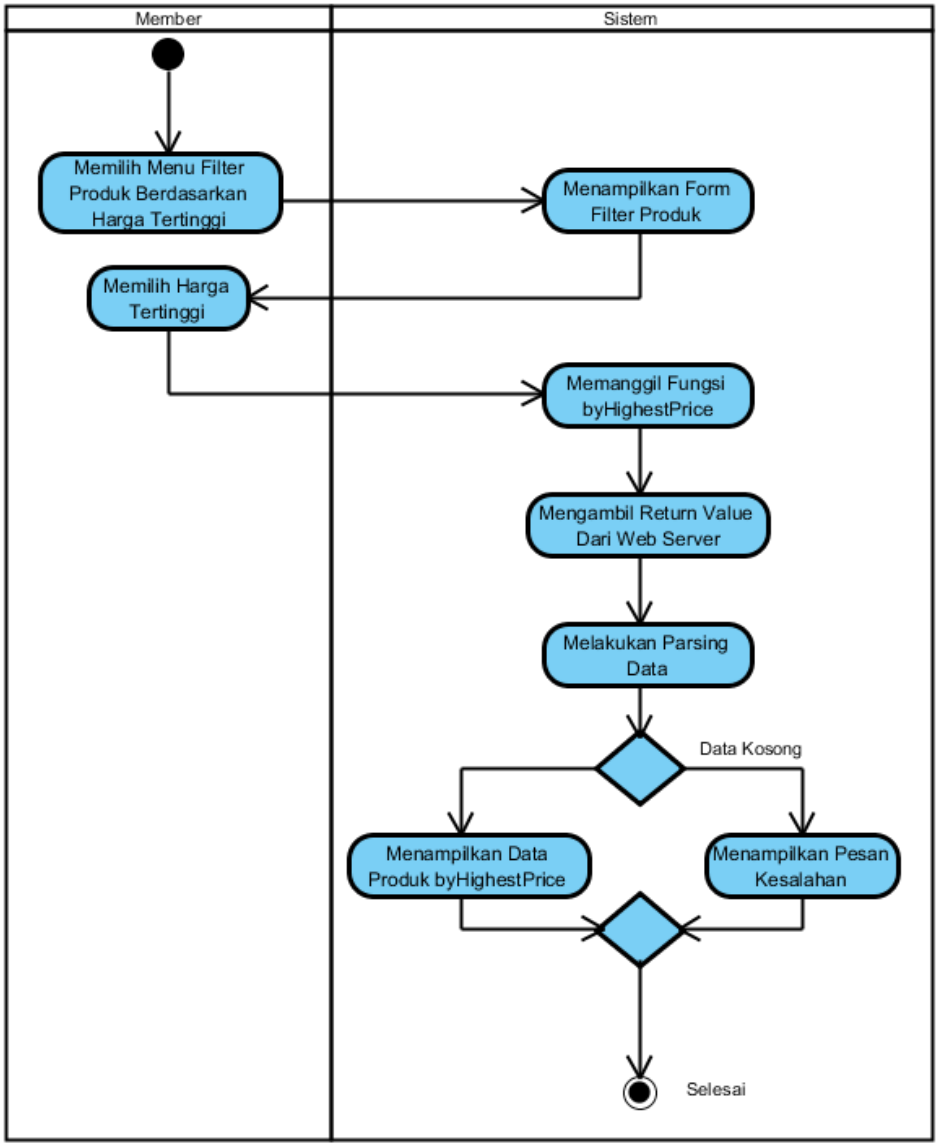
Gambar III.11 Activity Diagram GetProductByDiscount

g. Activity Diagram *GetProductByHigestPrice*

Aktifitas get*ProductByHighestPrice* merupakan aktifitas yang dilakukan oleh member untuk memfilter data produk berdasarkan harga tertinggi yang ada pada *web server*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Memilih menu filter produk berdasarkan harga tertinggi, pada aktifitas ini user memilih menu untuk memfilter data berdasarkan harga tertinggi produk.
2. Memilih harga tertinggi, pada aktifitas ini user memilih harga tertinggi produk untuk memfilter data produk.
3. memanggil fungsi *byHighestPrice*, pada aktifitas ini sistem memanggil fungsi *byHighestPrice* yang ada pada *web server*.
4. Mengambil *return value* dari *web server*, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh *web server* berupa hasil dari eksekusi *method* pada *web server*.
5. Melakukan *parsing* data, pada aktifitas ini sistem memparsing respon dari *web server* agar data yang diterima dapat dibaca.
6. Menampilkan data produk, pada aktifitas ini sistem menampilkan hasil *parsing* kedalam bentuk *list view* jika data hasil parsing tidak kosong.
7. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas get*ProductByHighestPrice* terlihat pada *activity diagram* gambar III.12.



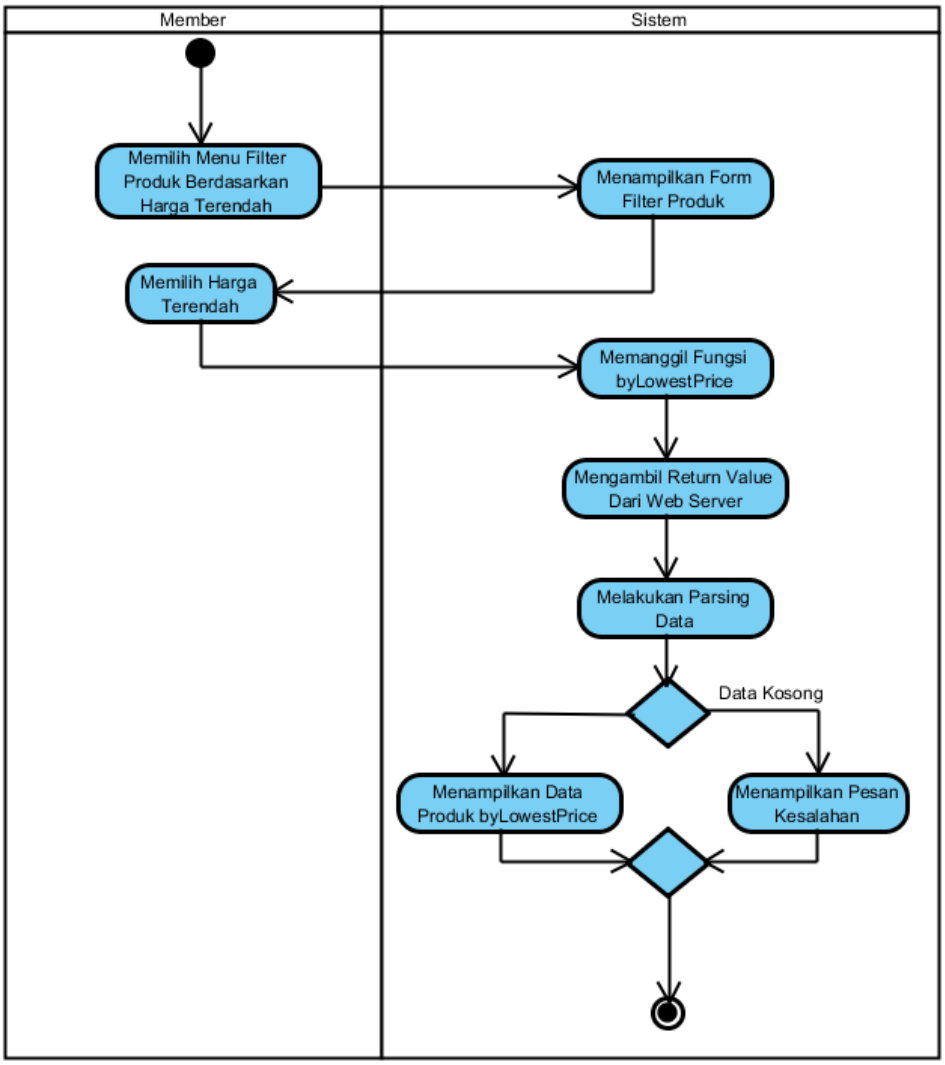
Gambar III.12 Activity Diagram GetProductByHighestPrice

h. Activity Diagram *GetProductByLowestPrice*

Aktifitas get*ProductByLowestPrice* merupakan aktifitas yang dilakukan oleh member untuk memfilter data produk berdasarkan harga terendah yang ada pada *web server*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Memilih menu filter produk berdasarkan harga terendah, pada aktifitas ini user memilih menu untuk memfilter data berdasarkan harga terendah produk.
2. Memilih produk terendah, pada aktifitas ini user memilih harga terendah produk untuk memfilter data produk.
3. memanggil fungsi *byLowestPrice*, pada aktifitas ini sistem memanggil fungsi *byLowestPrice* yang ada pada *web server*.
4. Mengambil *return value* dari *web server*, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh *web server* berupa hasil dari eksekusi *method* pada *web server*.
5. Melakukan *parsing* data, pada aktifitas ini sistem memparsing respon dari *web server* agar data yang diterima dapat dibaca.
6. Menampilkan data produk, pada aktifitas ini sistem menampilkan hasil *parsing* kedalam bentuk *list view* jika data hasil parsing tidak kosong.
7. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas get*ProductByLowestPrice* terlihat pada *activity diagram* gambar III.13.



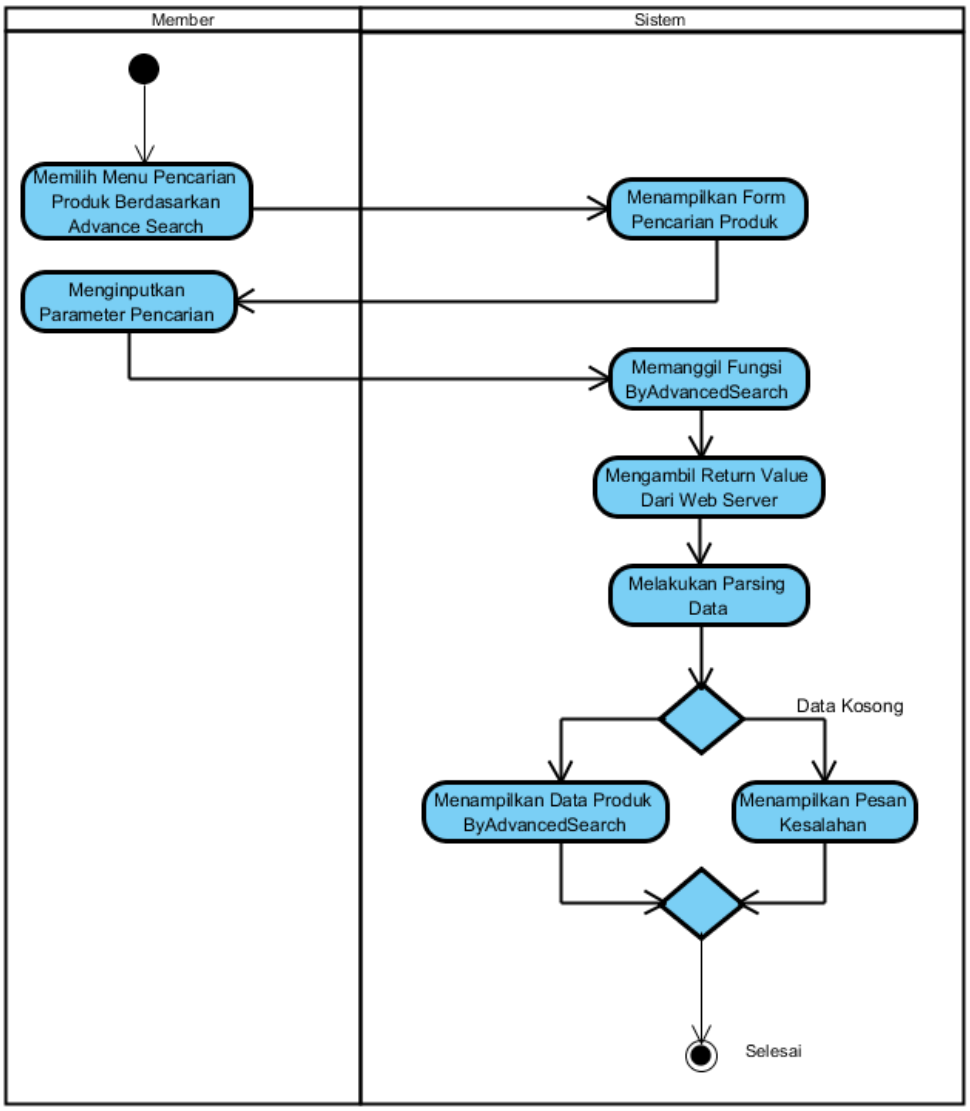
Gambar III.13 Activity Diagram GetProductByLowestPrice

i. Activity Diagram *SearchByAdvanced*

Aktifitas *SearchProductByAdvancedSearch* merupakan aktifitas yang dilakukan oleh member untuk mencari data produk berdasarkan pencarian advance yang ada pada *web server*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Memilih menu pencarian produk berdasarkan pencarian advance, pada aktifitas ini user memilih menu untuk pencarian data berdasarkan pencarian advanace produk.
2. Memilih pencarian advance, pada aktifitas ini user menginputkan pencarian advance produk untuk mencari data produk.
3. memanggil fungsi *byAdvancedSearch*, pada aktifitas ini sistem memanggil fungsi *byAdvancedSearch* yang ada pada *web server*.
4. Mengambil *return value* dari *web server*, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh *web server* berupa hasil dari eksekusi *method* pada *web server*.
5. Melakukan *parsing* data, pada aktifitas ini sistem memparsing respon dari *web server* agar data yang diterima dapat dibaca.
6. Menampilkan data produk, pada aktifitas ini sistem menampilkan hasil *parsing* kedalam bentuk *list view* jika data hasil parsing tidak kosong.
7. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas *SearchProductByAdvancedSearch* terlihat pada *activity diagram* gambar III.16.



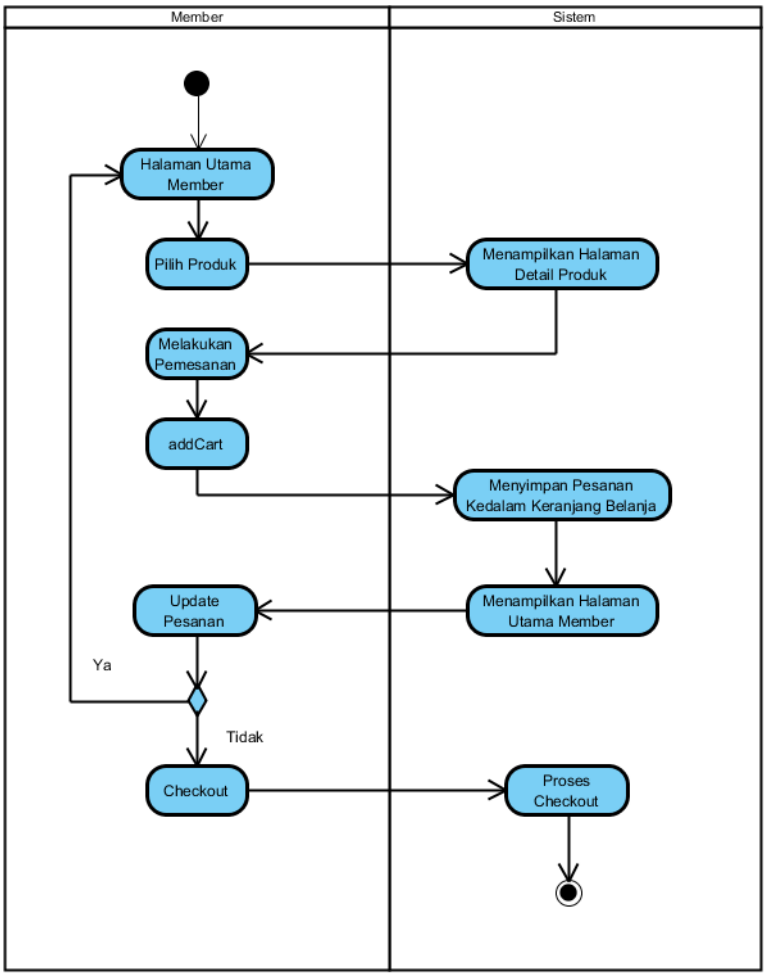
Gambar III.14 Activity Diagram SearchProductByAdvanceSearch

j. Activity Diagram *AddCart*

Aktifitas *addCart* merupakan aktifitas yang dilakukan oleh member untuk melakukan proses memasukan pesanan kedalam keranjang belanja. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Memilih menu *addCart*, pada aktifitas ini user memilih menu *addCart* yang ada pada sistem.

Seluruh aktifitas *addCart* terlihat pada *activity diagram* gambar III.17.



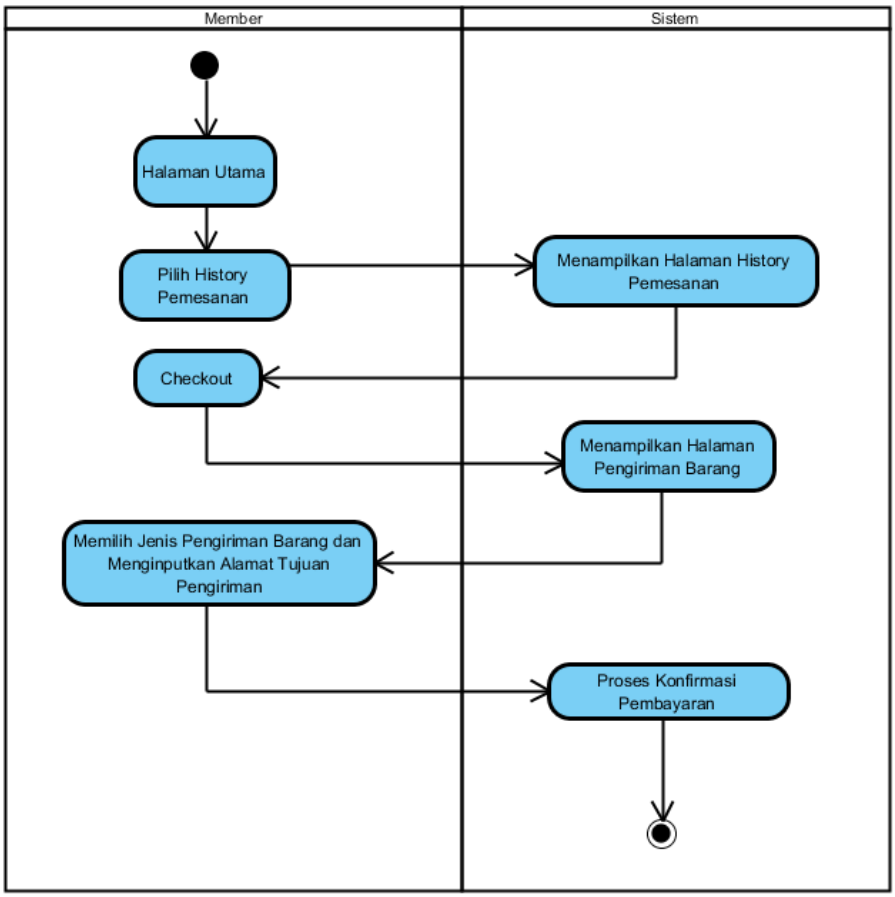
Gambar III.15 Activity Diagram AddCart

k. Activity Diagram *Checkout*

Aktifitas *Checkout* merupakan aktifitas yang dilakukan oleh member untuk melakukan proses pemeriksaan produk yang dipesan. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Menjalankan menu utama, pada aktifitas ini user menjalankan menu utama yang ada pada sistem.
2. Memilih *Product*, pada aktifitas ini user memilih produk yang akan dipesan.
3. Memilih *DetailProduct*, pada aktifitas ini user memilih detail produk yang akan dipesan.
4. Memilih menu *addCart*, pada aktifitas ini user memilih menu *addCart* yang ada pada sistem.
5. Memilih menu *UpdateOrder*, pada aktifitas ini user memilih menu *UpdateOrder* yang ada pada sistem.
6. Memilih menu *Checkout*, pada aktifitas ini user memilih menu *Checkout* yang ada pada sistem.

Seluruh aktifitas *Checkout* terlihat pada *activity diagram* gambar III.18.



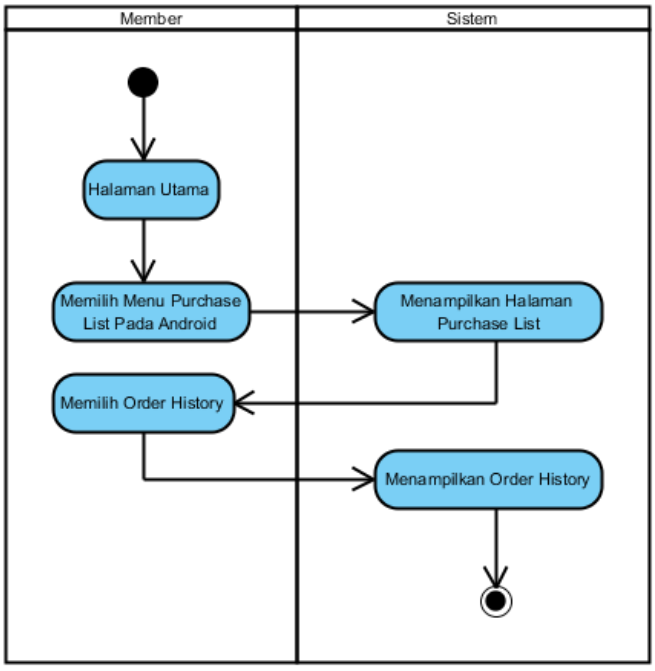
Gambar III.16 Activity Diagram Checkout

l. Activity Diagram *OrderHistory*

Aktifitas *OrderHistory* merupakan aktifitas yang dilakukan oleh member untuk melihat semua transaksi yang telah dilakukan oleh member. Dalam aktifitas ini terdapat beberapa aktifitas yaitu:

1. Menjalankan menu utama, pada aktifitas ini member menjalankan menu utama yang ada pada sistem.
2. Memilih tampilan menu, pada aktifitas ini member memilih tampilan menu pada *mobile* android.
3. *OrderHistory*, pada aktifitas ini member datpat melihat semua transaksi yang telah dilakukan.

Seluruh aktifitas *OrderHistory* terlihat pada *activity diagram* gambar III.22.



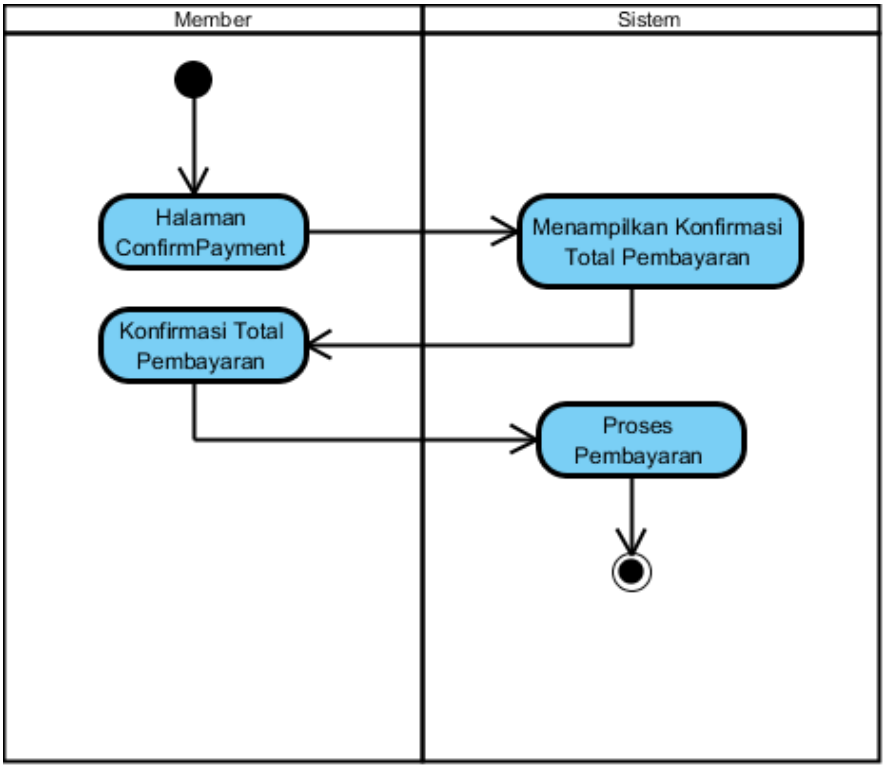
Gambar III.17 Activity Diagram OrderHistory

m. Activity Diagram *ConfirmPayment*

Aktifitas *ConfirmPayment* merupakan aktifitas yang dilakukan oleh member untuk melakukan proses konfirmasi pembayaran. Dalam aktifitas ini terdapat beberapa aktifitas yaitu:

1. *ConfirmPayment,* pada aktifitas ini user mengkonfirmasi total pembayaran yang harus dibayar.

Seluruh aktifitas *ConfirmPayment* terlihat pada *activity diagram* gambar III.20.



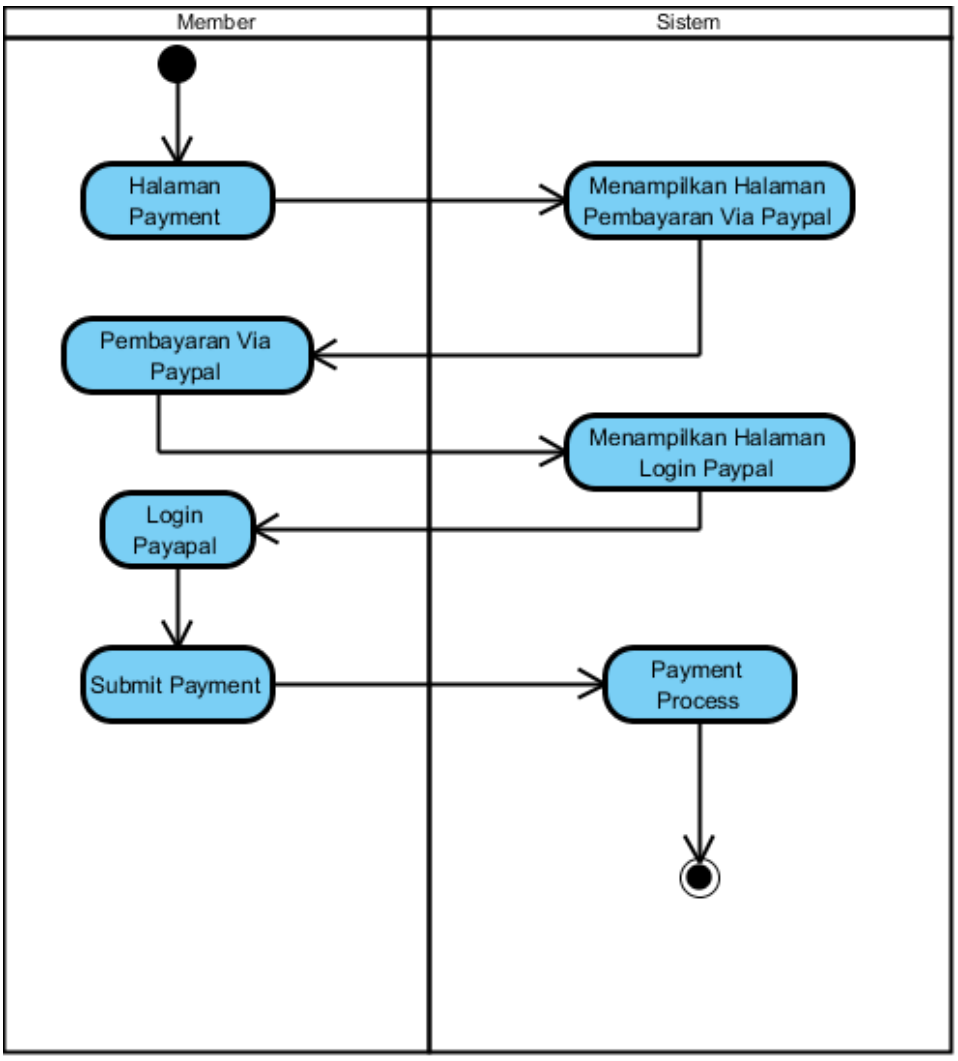
Gambar III.18 Activity Diagram ConfirmPayment

n. Activity Diagram *Paypal Payment*

Aktifitas *Paypal Payment* merupakan aktifitas yang dilakukan oleh member untuk melakukan proses pembayaran produk. Dalam aktifitas ini terdapat beberapa aktifitas yaitu:

1. *Paypal Payment*, pada aktifitas ini user melakukan proses pembayaran produk yang dipesan menggunakan *paypal*.

Seluruh aktifitas *Paypal Payment* terlihat pada *activity diagram* gambar III.21.



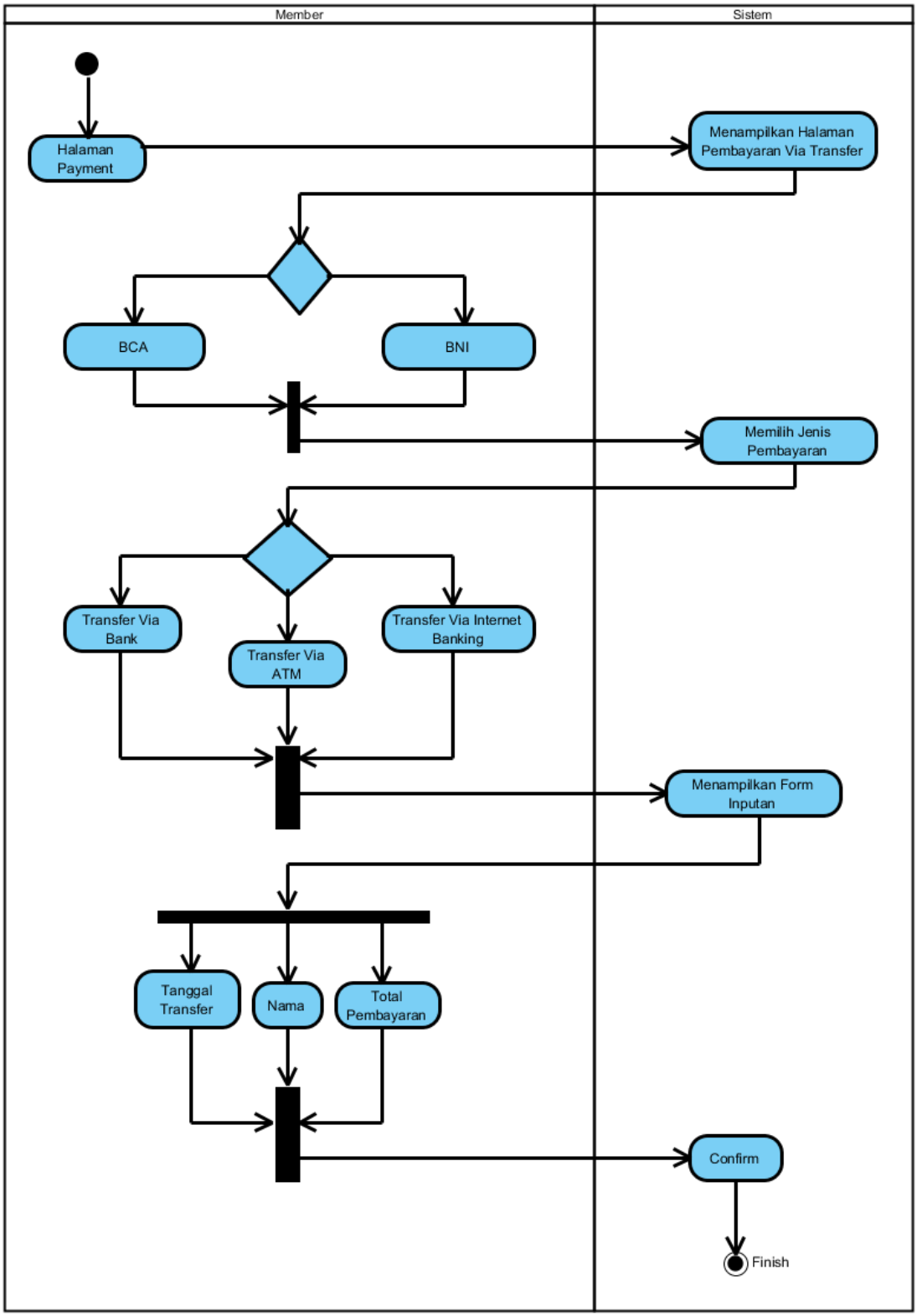
Gambar III.19 Activity Diagram Paypal Payment

o. Activity Diagram Transfer *Payment*

Aktifitas Transfer *Payment* merupakan aktifitas yang dilakukan oleh member untuk melakukan proses transfer pembayaran produk. Dalam aktifitas ini terdapat beberapa aktifitas yaitu:

1. Transfer *Payment*, pada aktifitas ini user melakukan proses transfer pembayaran produk yang dipesan.

Seluruh aktifitas Transfer *Payment* terlihat pada *activity diagram* gambar III.21.



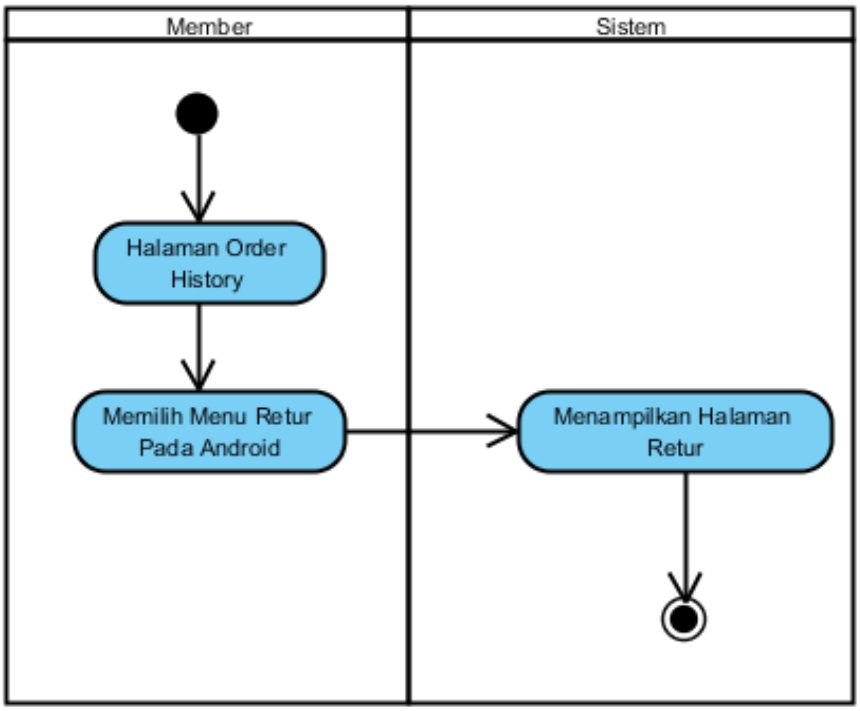
Gambar III.20 Activity Diagram Transfer Payment

p. Activity Diagram Retur

Aktifitas Retur merupakan aktifitas yang dilakukan oleh member untuk pengembalian barang karena hal tertentu. Dalam aktifitas ini terdapat beberapa aktifitas yaitu:

1. Menjalankan menu utama, pada aktifitas ini member menjalankan menu utama yang ada pada sistem.
2. Memilih tampilan menu, pada aktifitas ini member memilih tampilan menu pada *mobile* android.
3. *OrderHistory,* pada aktifitas ini member datpat melihat semua transaksi yang telah dilakukan.

Seluruh aktifitas *OrderHistory* terlihat pada *activity diagram* gambar III.23.

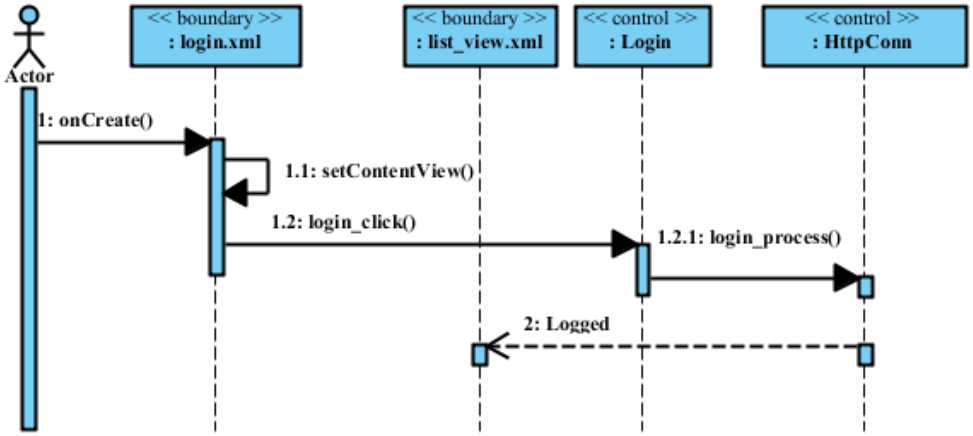


Gambar III.21 Activity Diagram Retur

3. Sequence Diagram

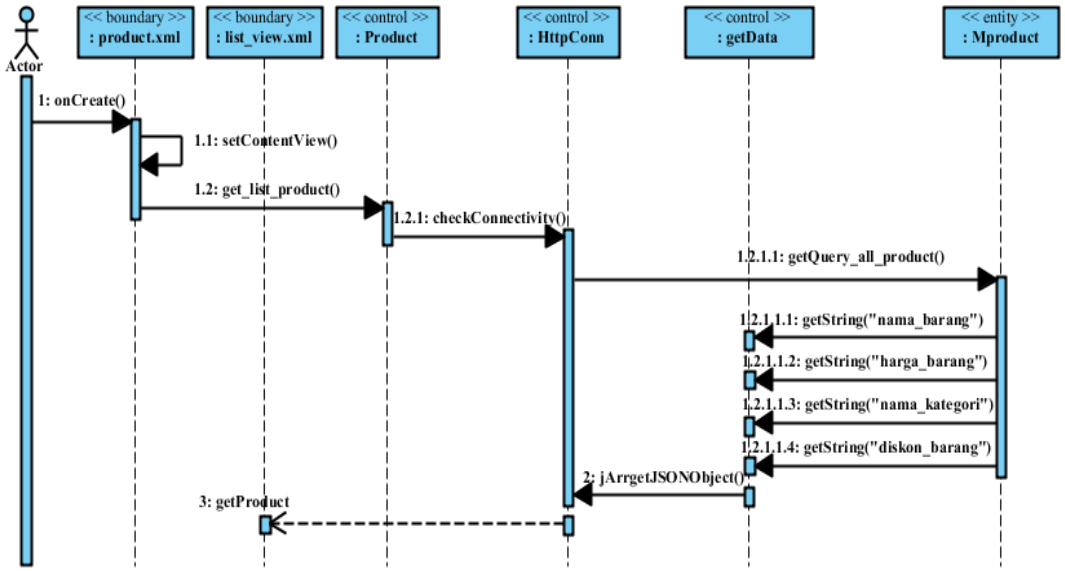
Diagram Sequence merupakan bentuk dari *interaction diagram*, menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *Message* yang di gambarkan terhadap waktu. Masing – masing objek termasuk *actor* memiliki line vertikal, *message* digambarkan garis berpanah dari satu objek ke objek lainnya, dan activation bar menunjukan lamanya eksekusi sebuah proses. Sebuah sequence diagram, secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan – pesan yang melewati objek – objek ini di dalam use case. Berikut ini adalah sequence diagram dari sistem yang akan dibangun.

a. Sequence Diagram *Login*



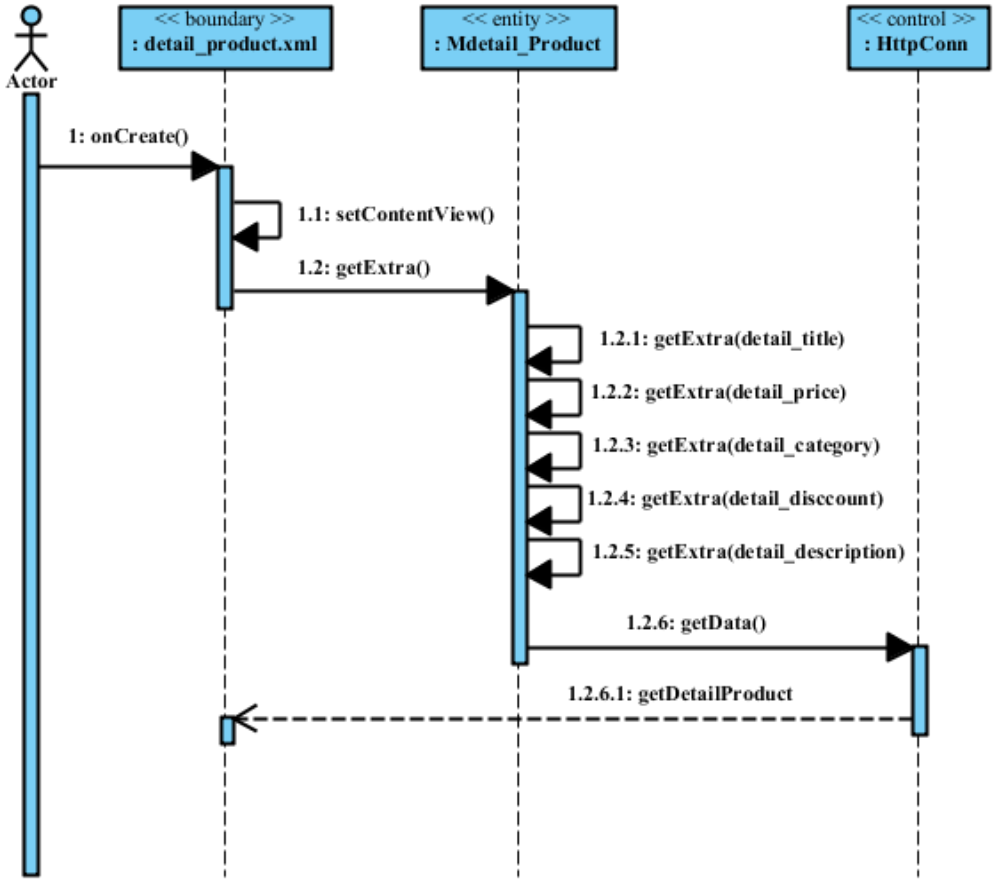
Gambar III.22 Sequence Diagram Login.

b. Sequence Diagram *GetProduct*



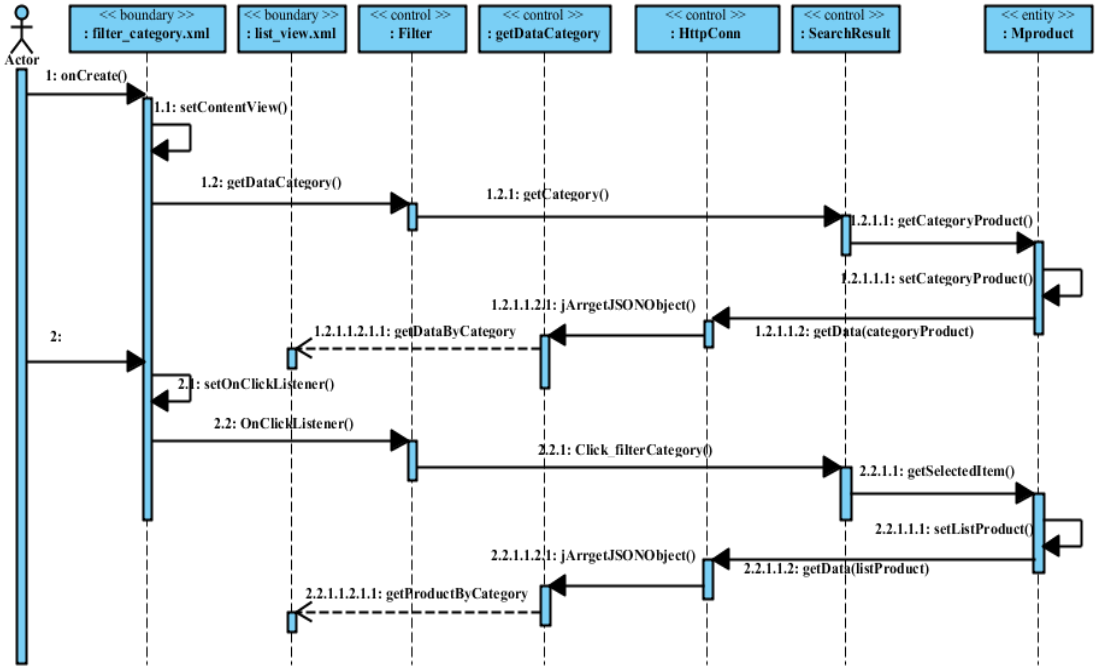
Gambar III.23 Sequence Diagram GetProduct

c. Sequence Diagram *GetDetailProduct*



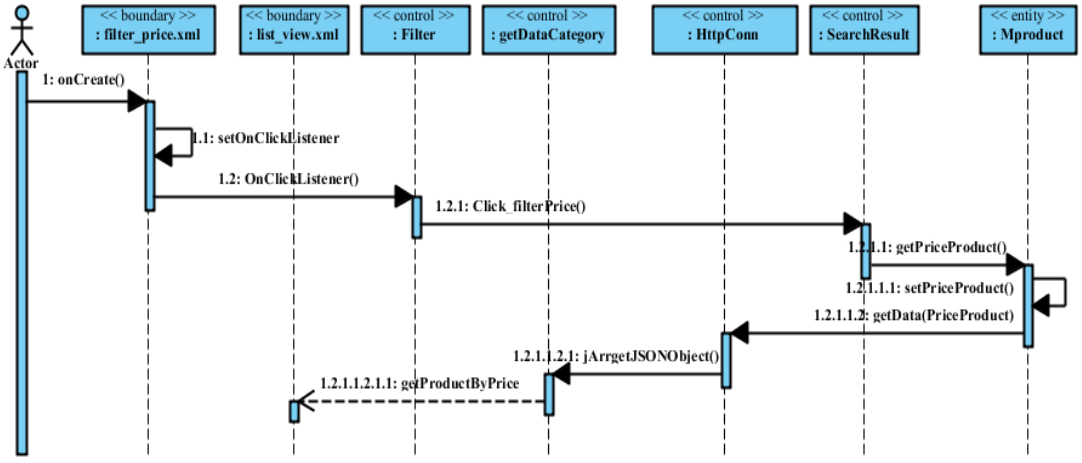
Gambar III.24 Sequence Diagram GetDetailProduct

d. Sequence Diagram *GetProductByCategory*



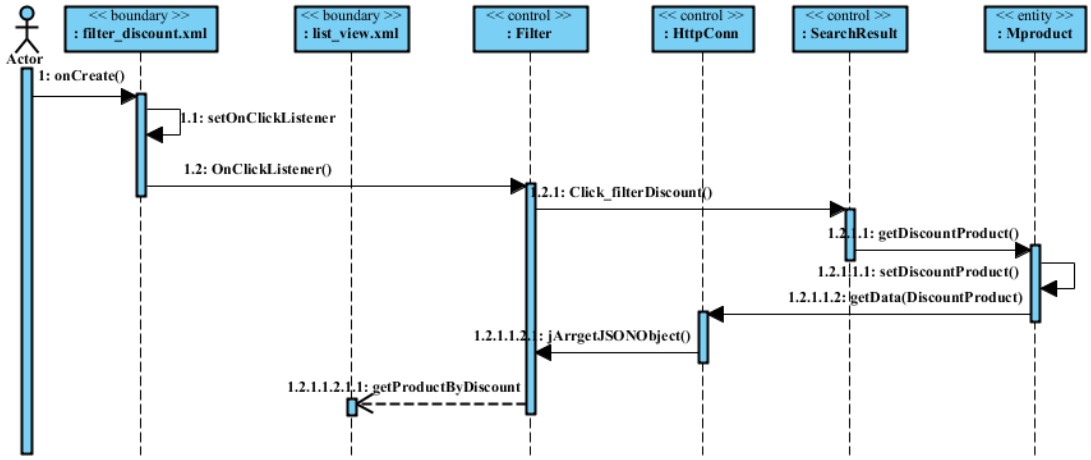
Gambar III.25 Sequence Diagram GetProductByCategory

e. Sequence Diagram *GetProductByRangePrice*



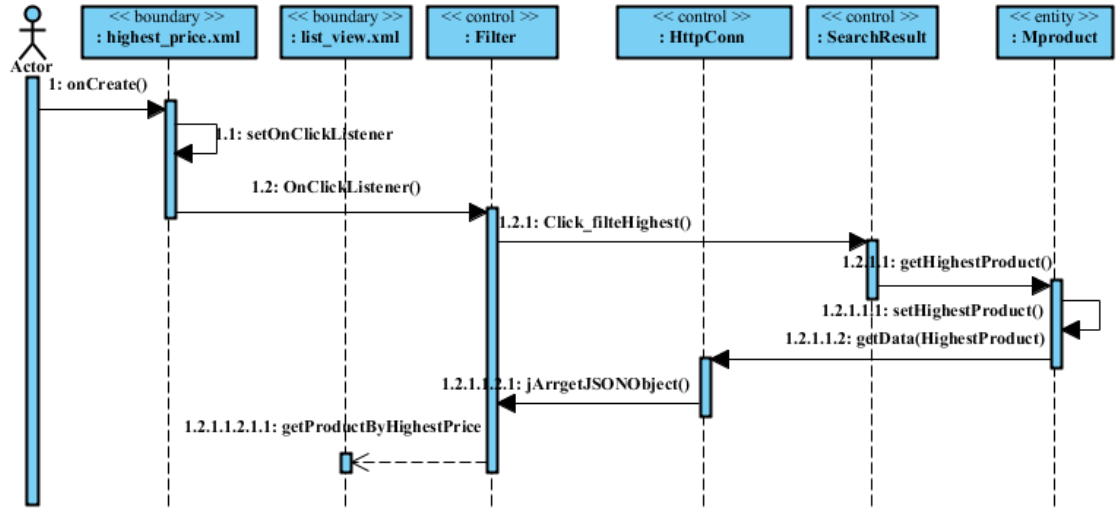
Gambar III.26 Sequence Diagram GetProductByRangePrice

f. Sequence Diagram *GetProductByDiscount*



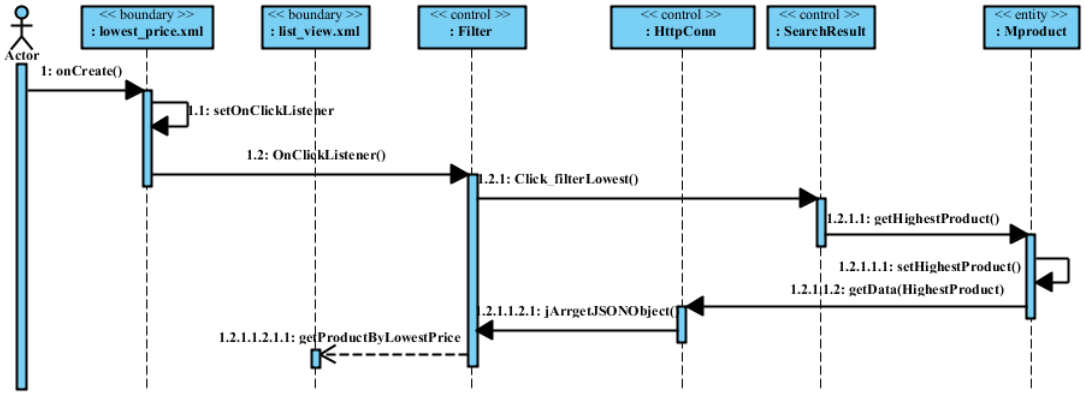
Gambar III.27 Sequence Diagram GetProductByDiscount

g. Sequence Diagram *GetProductByHigestPrice*



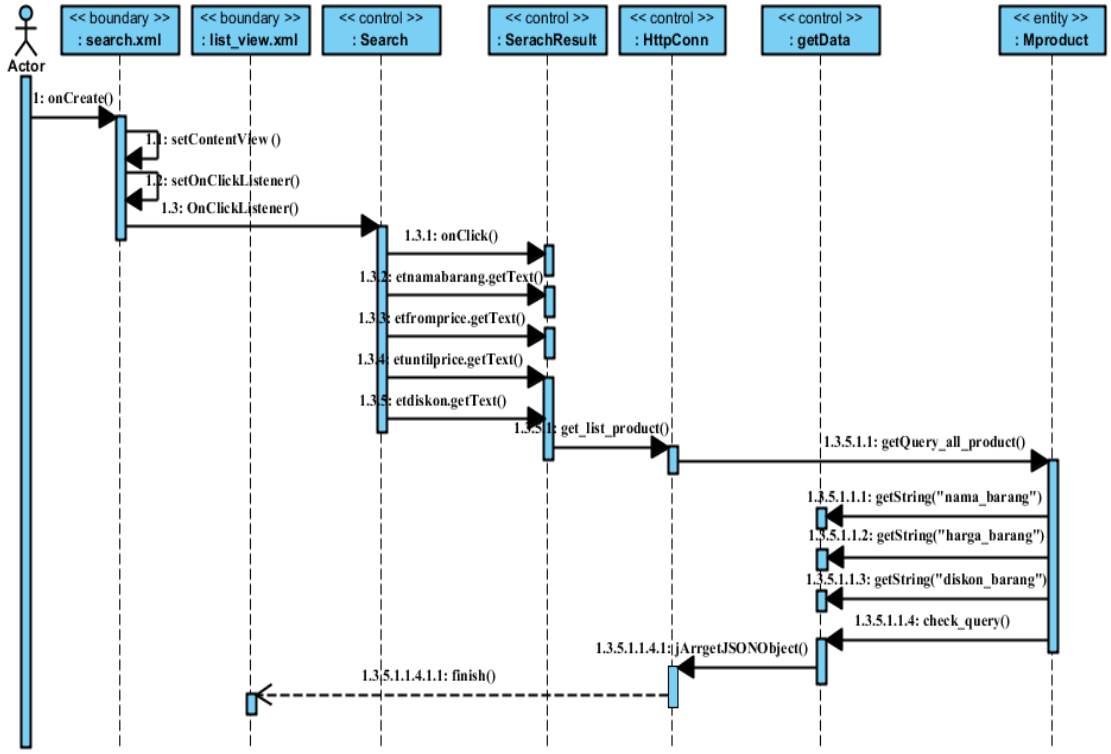
Gambar III.28 Sequence Diagram GetProductByHigestPrice

h. Sequence Diagram *GetProductByLowestPrice*



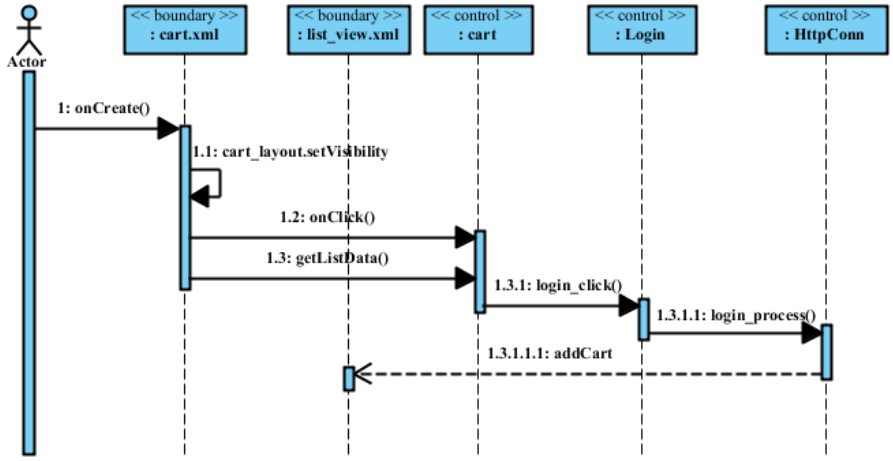
Gambar III.29 Sequence Diagram GetProductByLowestPrice

i. Sequence Diagram *SearchByAdvanced*



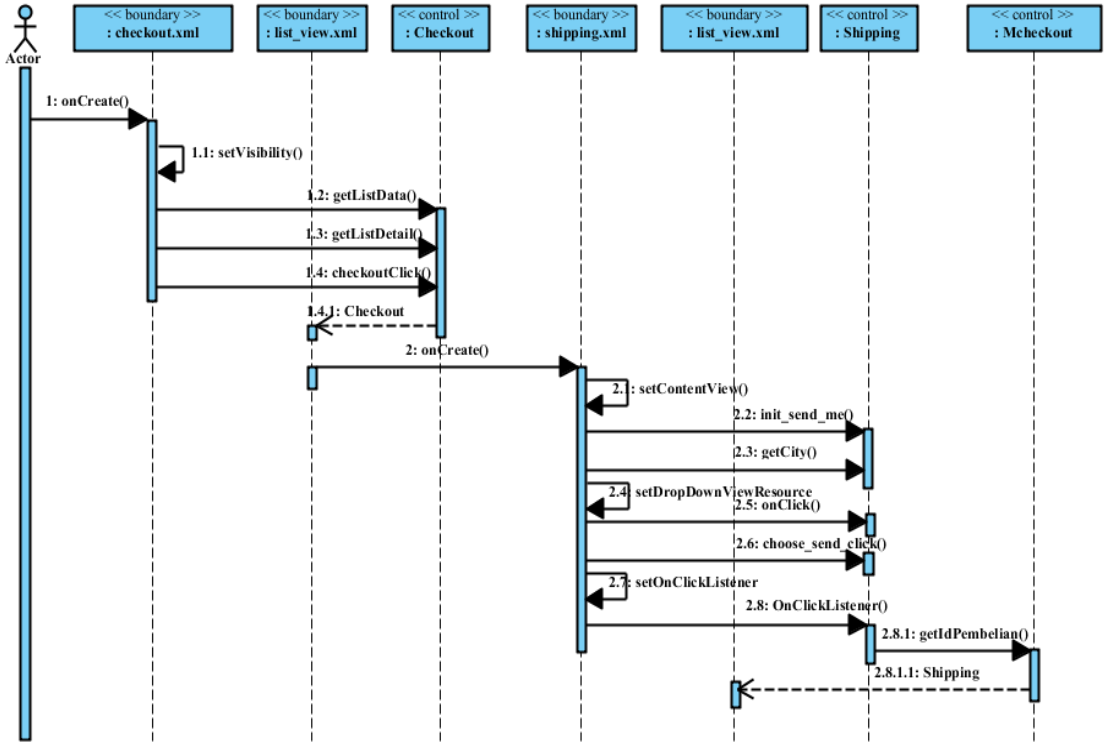
Gambar III.30 Sequence Diagram SearchByAdvanced

j. Sequence Diagram *AddCart*



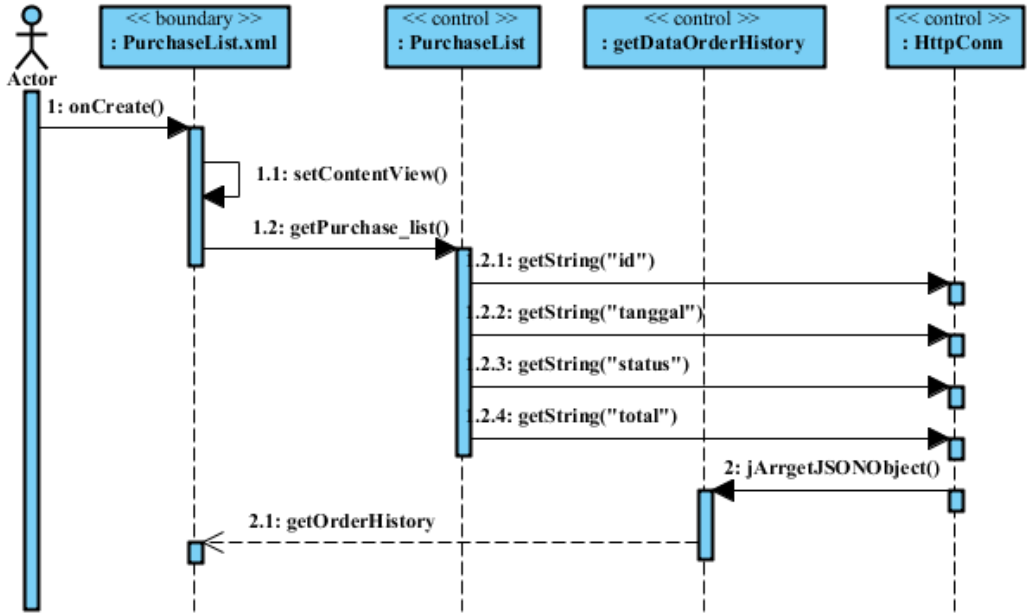
Gambar III.31 Sequence Diagram addCart

k. Sequence Diagram *Checkout*



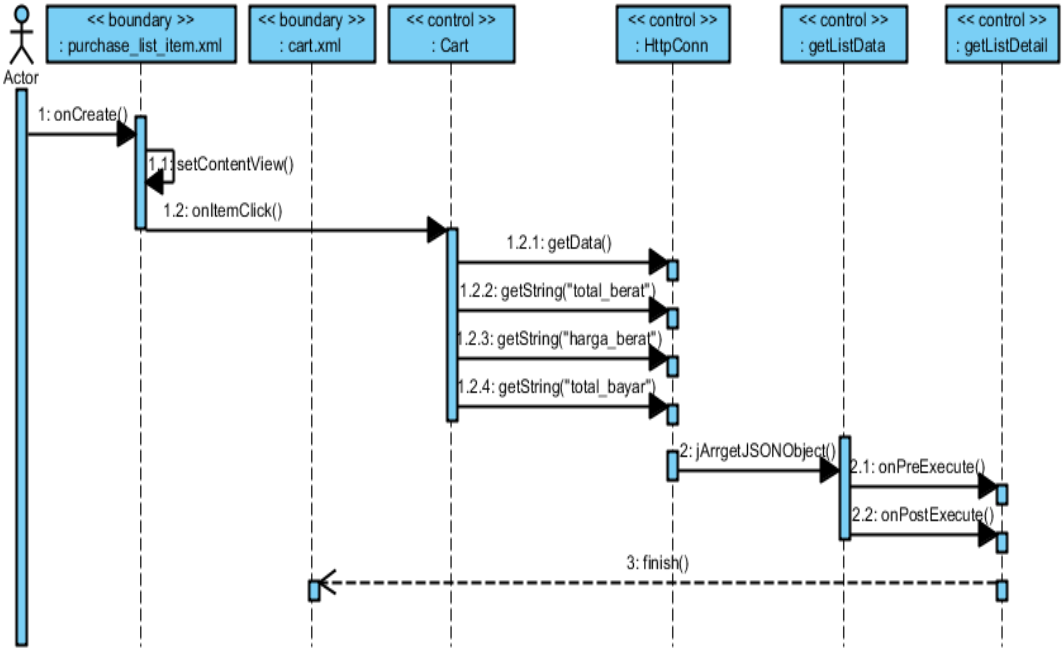
Gambar III.32 Sequence Diagram Checkout

l. Sequence Diagram *OrderHistory*



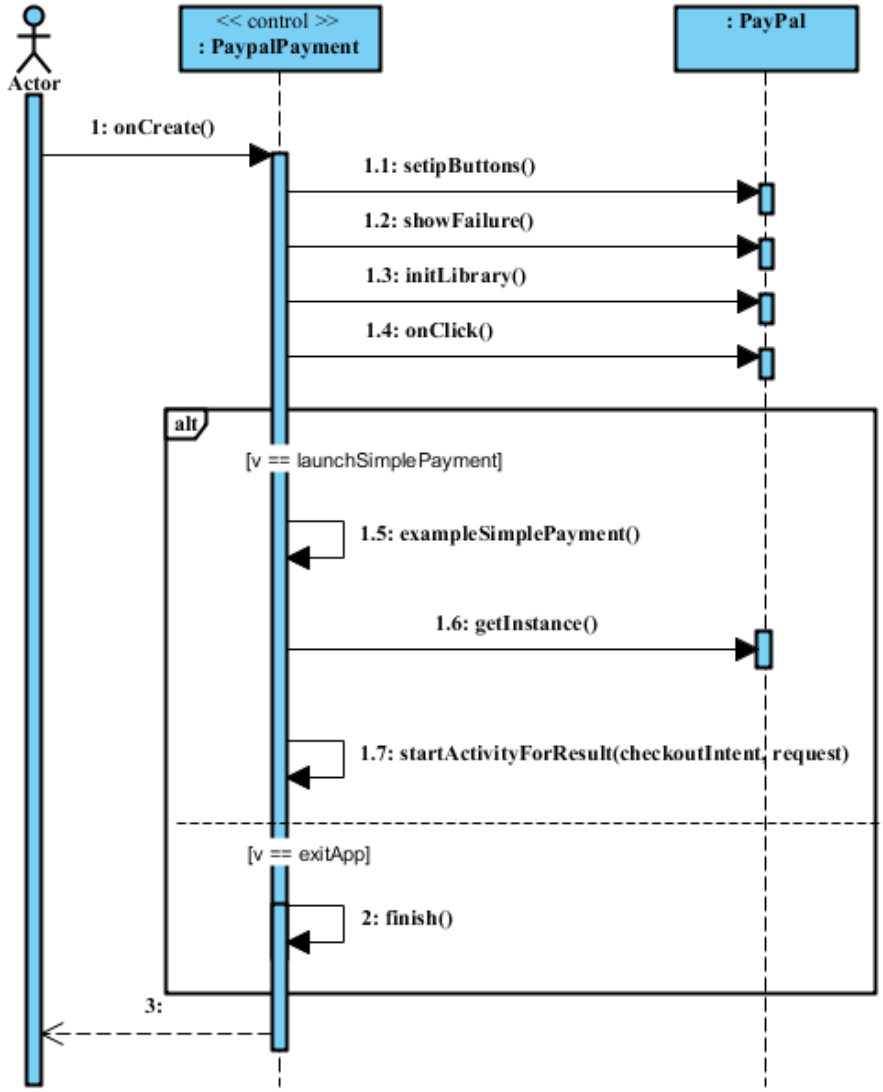
Gambar III.33 Sequence Diagram OrderHistory

m. Sequence Diagram *ConfirmPayment*



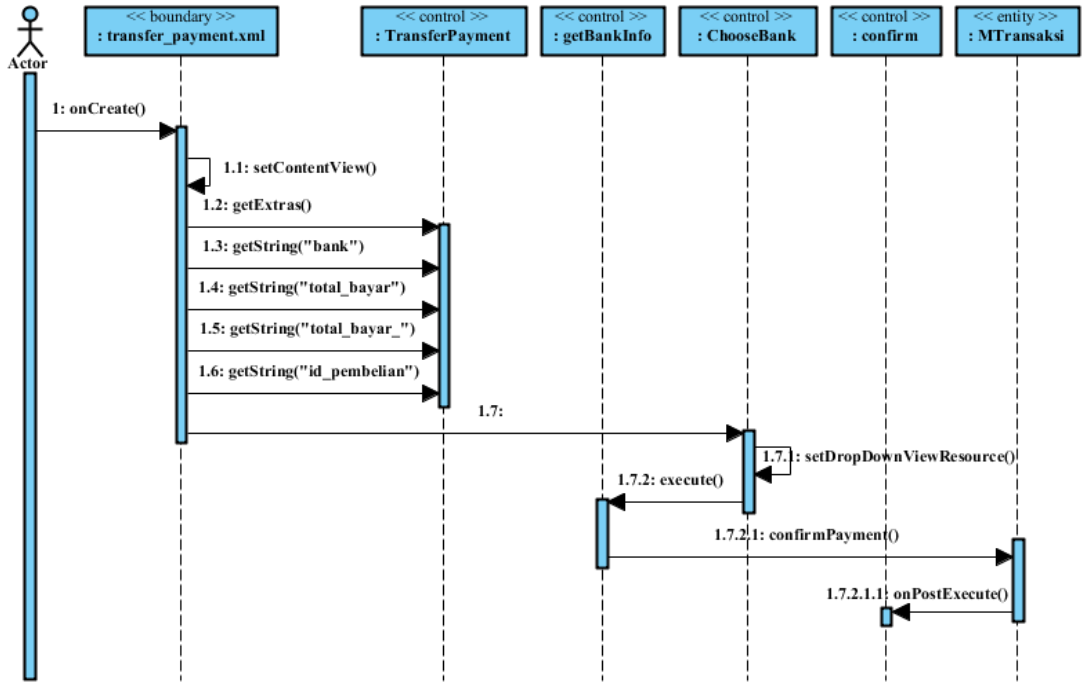
Gambar III.34 Sequence Diagram ConfirmPayment

n. Sequence Diagram *PaypalPayment*



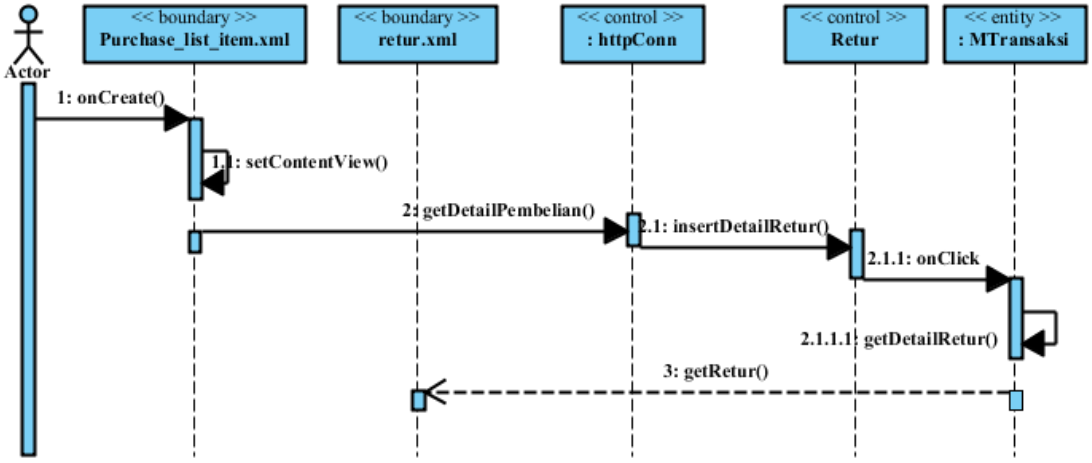
Gambar III.35 Sequence Diagram PaypalPayment

o. Sequence Diagram *TransferPayment*



Gambar III.36 Sequence Diagram TranferPayment

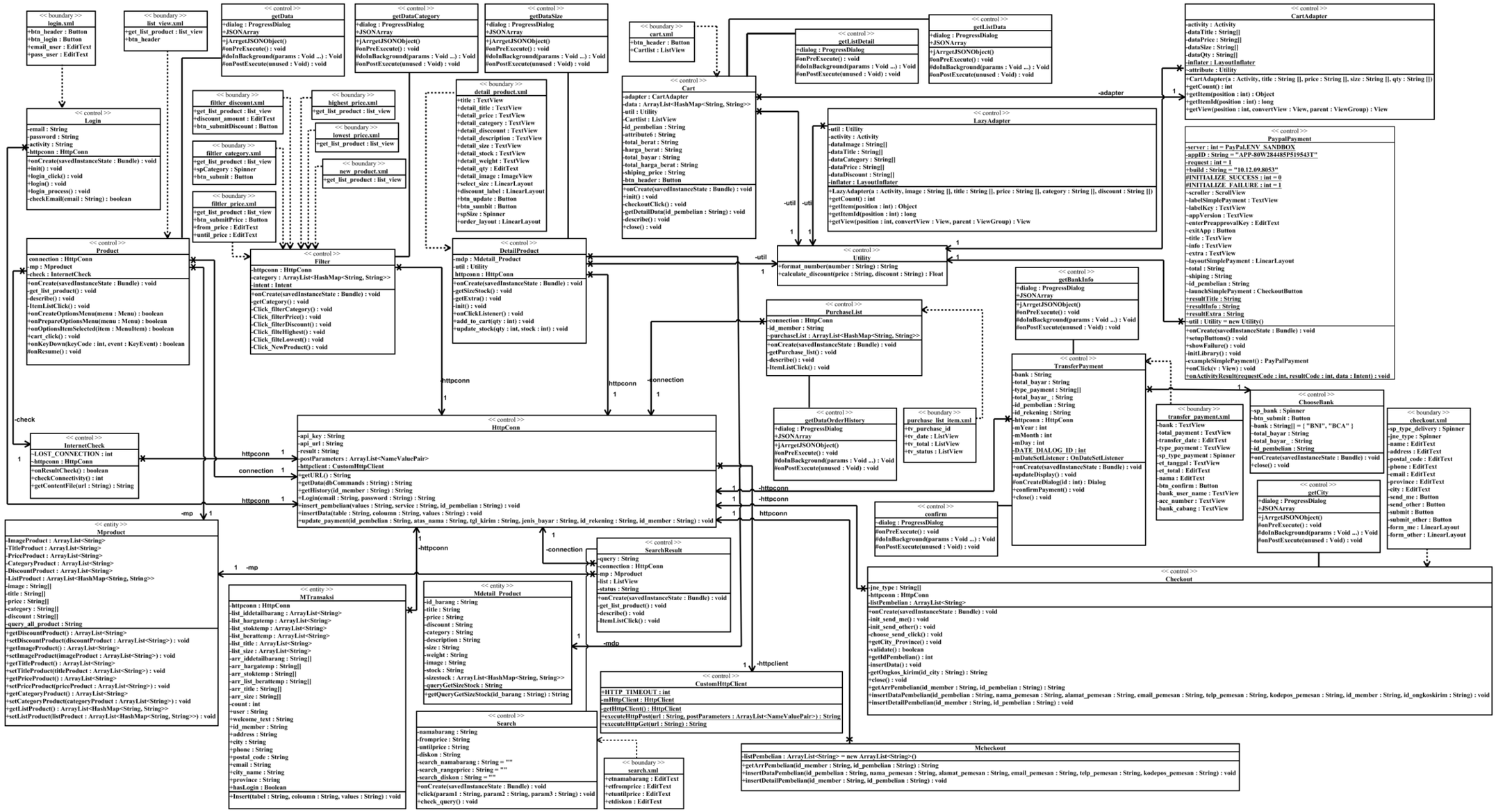
p. Sequence Diagram Retur



Gambar III.37 Sequence Diagram Retur

4. Class Diagram

Class diagram menggambarkan struktur dan deskripsi class beserta hubungan satu sama lain seperti Asosiasi, Generalisasi dan lain sebagainya.



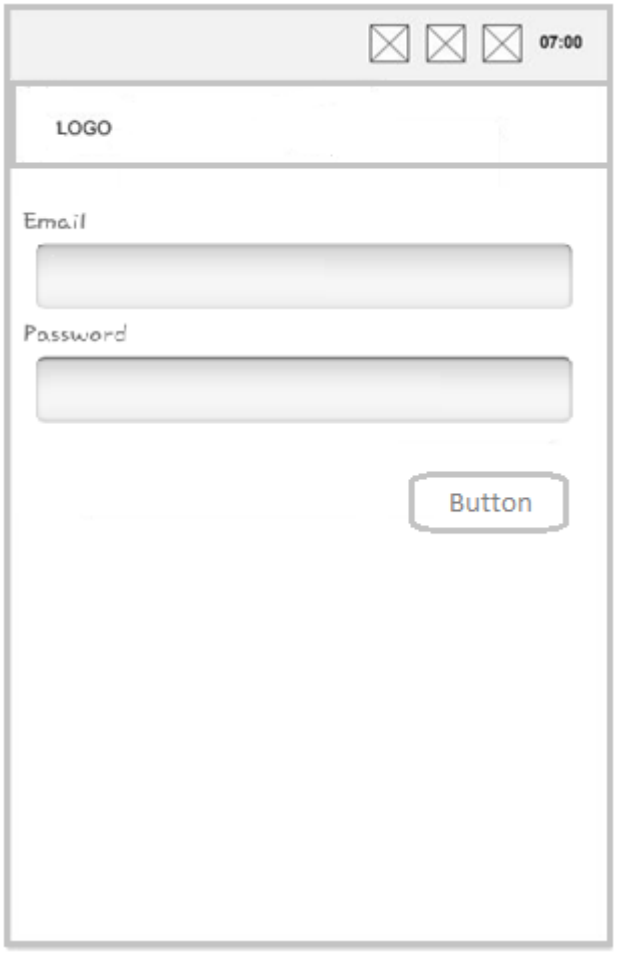
Gambar III.38 Class Diagram Pada Aplikasi M-Commerce

3.2 Perancangan

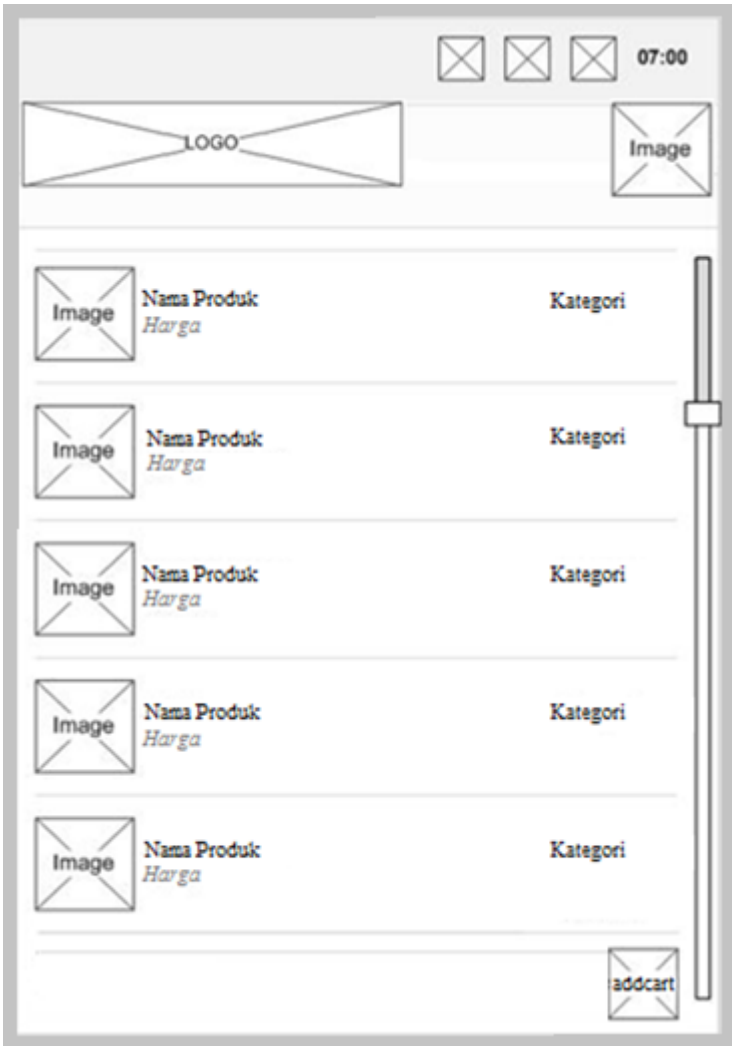
3.2.1 Perancangan Antarmuka

Interface atau antar muka merupakan tampilan dari suau program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan *user*. Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *user*. Berikut ini adalah perancangan antarmuka pada sistem yang akan dibuat.

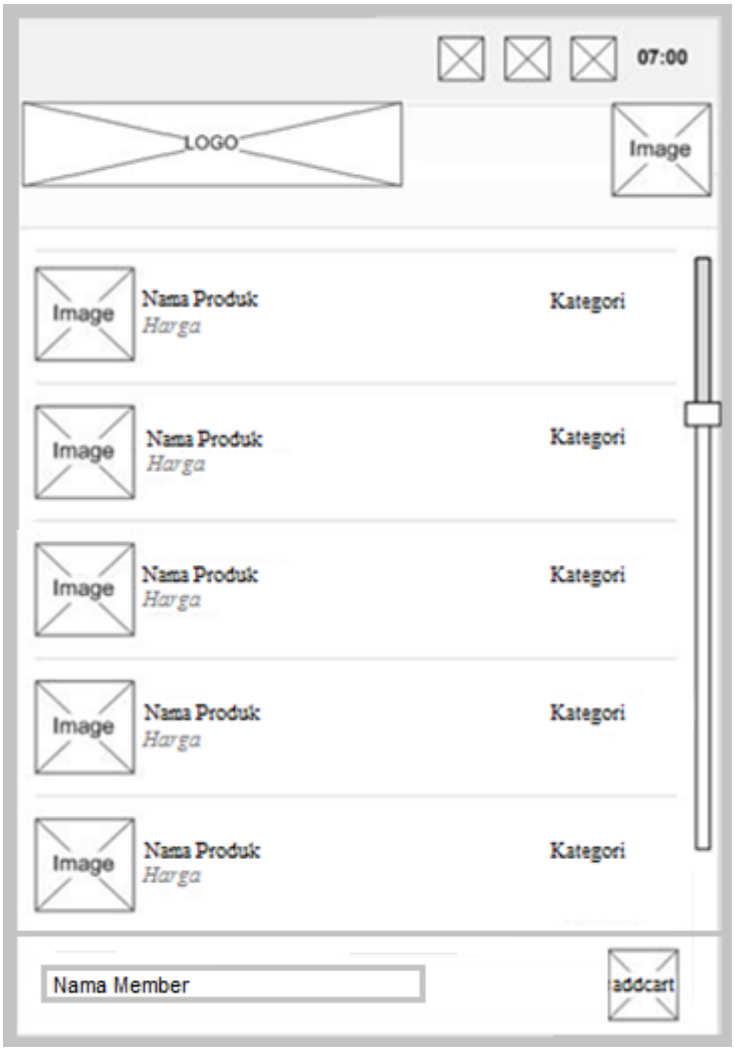
1. Perancangan Antarmuka Android



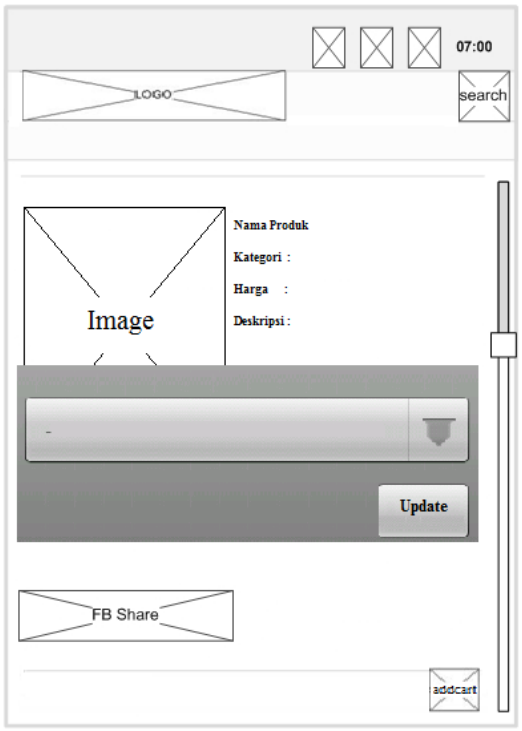
Gambar III.39 Perancangan Antarmuka menu login aplikasi android



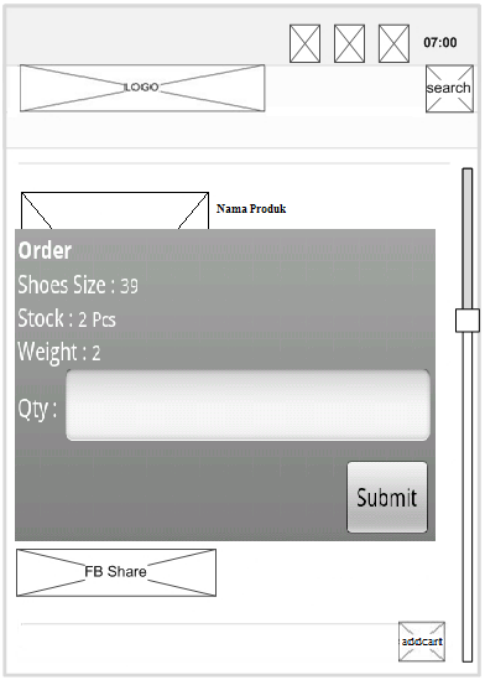
Gambar III.40 Perancangan antarmuka menu utama aplikasi android



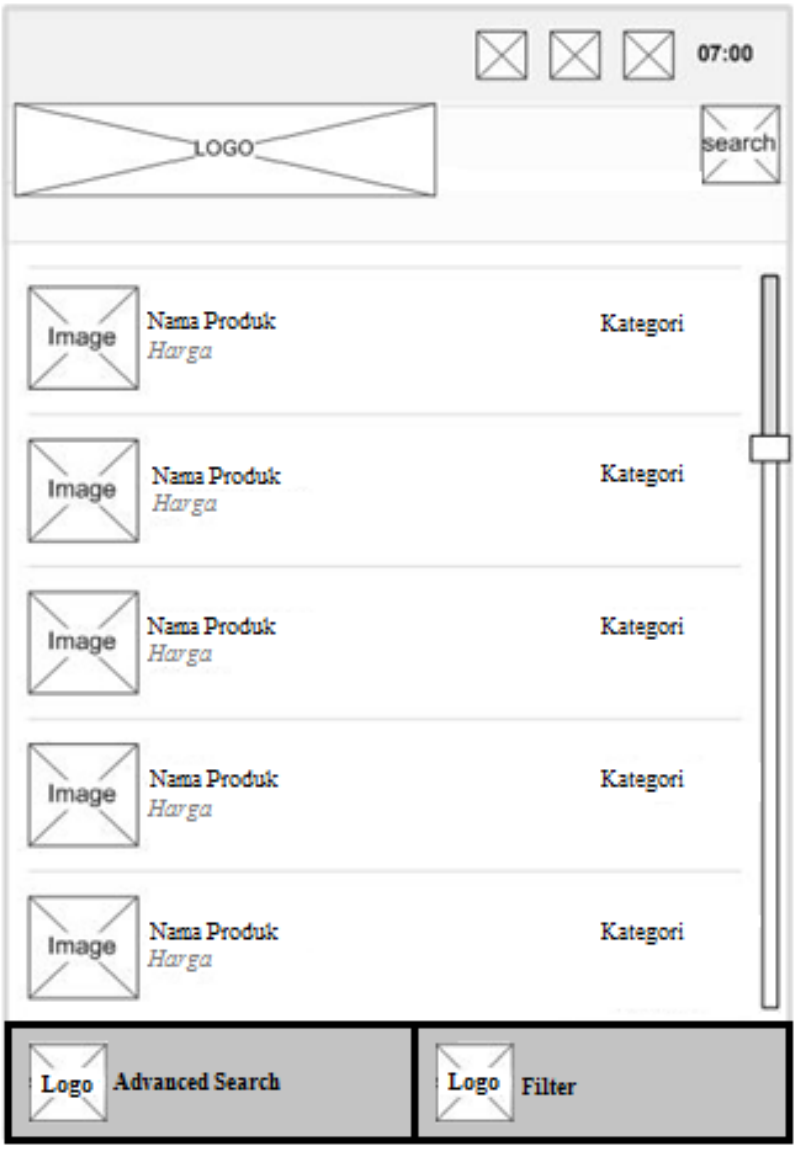
Gambar III.41 Perancangan antarmuka menu utama member



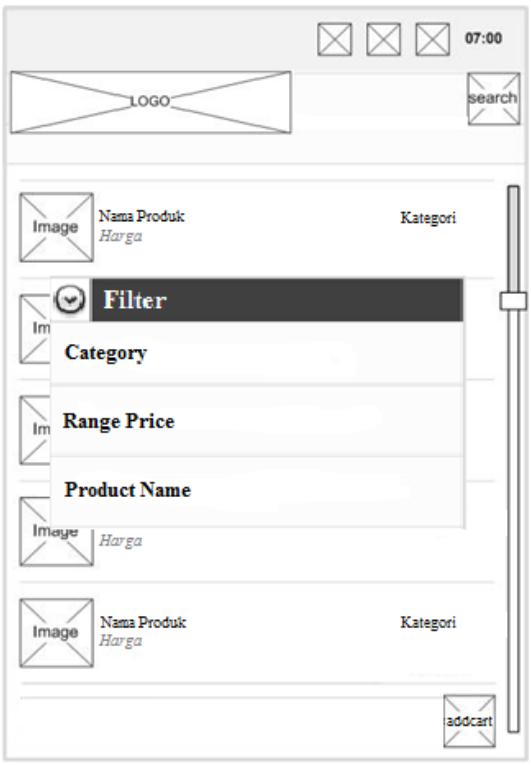
Gambar III.42 Perancangan antarmuka detail produk android



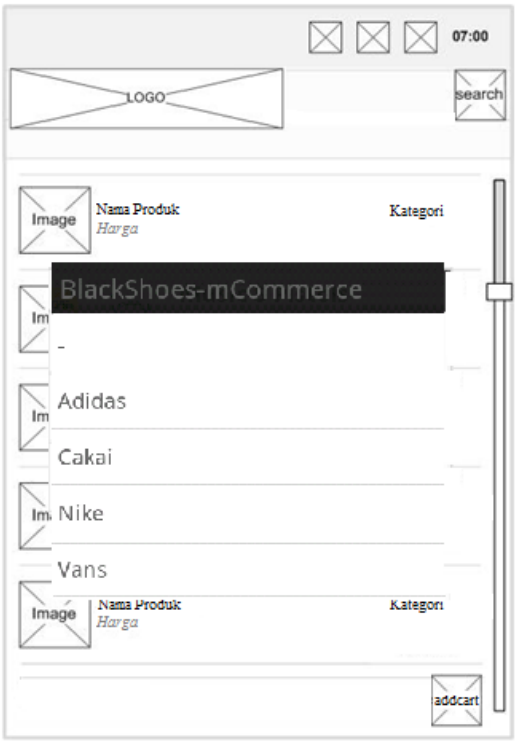
Gambar III.43 Perancangan antarmuka update detail produk android



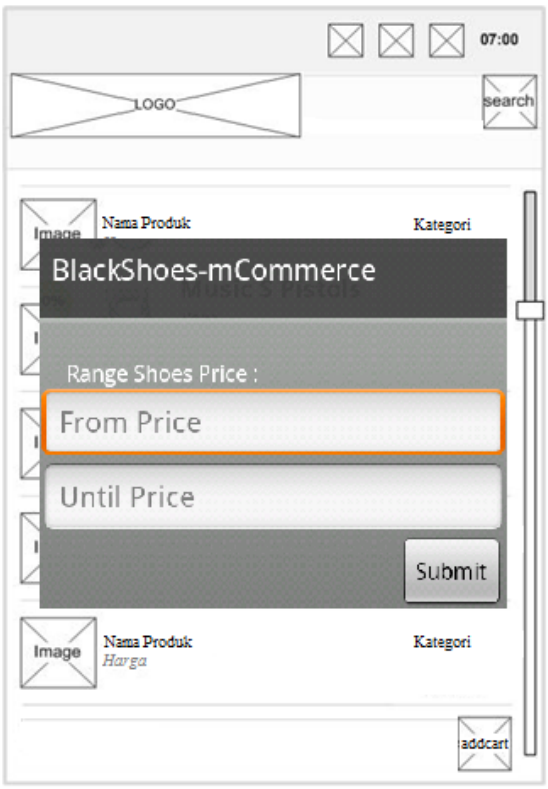
Gambar III.44 Perancangan antarmuka menu android



Gambar III.45 *Perancangan antarmuka filter Produk*



Gambar III.46 Perancangan antarmuka filter berdasarkan category



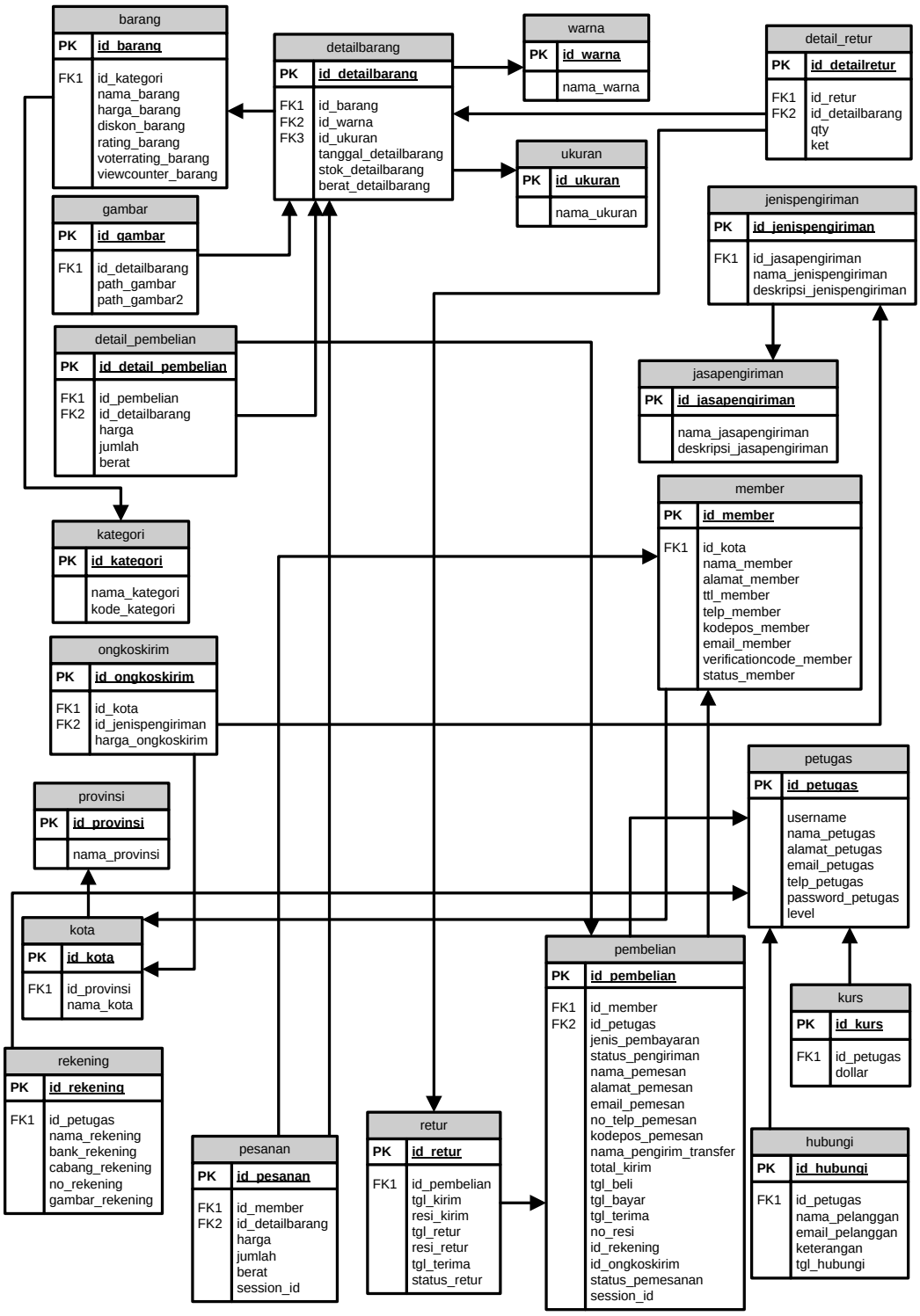
Gambar III.47 Perancangan antarmuka filter berdasarkan range price

3.2.2 Perancangan Database

Perancangan Database pada sistem ini dimaksudkan untuk mempermudah hubungan-hubungan antara tabel satu dengan tabel lainnya.

1. Skema Relasi

Skema relasi menggambarkan hubungan antar data, arti data yang sudah ada dalam keadaan normal dan batasannya dijelaskan dengan baris dan kolom. Adapun keterkaitan tabel relasi yang digunakan dalam sistem *M-Commerce* ini dapat digambarkan sebagai berikut:



Gambar III.48 Skema Relasi

2. Struktur Table

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam pembangunan aplikasi m-commerce di toko Blackshoes adalah sebagai berikut **:**

a. Tabel Petugas

Tabel III.24 Tabel Petugas

Nama Tabel **:** petugas				
Primary Key **:** id_petugas				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_petugas	Integer	5	PK	Primary Key
Username	Varchar	20		username
*Password*_petugas	Varchar	32		*password* petugas
Nama_petugas	Varchar	50		Nama Lengkap petugas
Email_petugas	Varchar	50		Alamat email petugas
Alamat	Varchar	50		Alamat petugas
Telp_petugas	Varchar	20		Telepon petugas
level	Enum(‘admin’, ’gudang’,’kasir’)	-		Level petugas

b. Tabel Member

Tabel III.25 Tabel Member

Nama Tabel **:** Member				
Primary Key **:** Id_Member				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_Member	Integer	11	PK	Primary Key
Id_kota	Integer	5	FK	reference table kota (id_Kota)
*Password*_member	Varchar	32		*Password* untuk login
Nama_member	Varchar	50		Nama_member
Alamat_member	Varchar	80		Alamat member
Ttl_member	Date	-		Tanggal lahir member
Tlp_member	Varchar	20		nomer telepon member
Email_member	Varchar	50		Alamat email

				member
Kodepos_member	Varchar	6		Kode pos member
verificationcode_member	Varchar	32		Verifikasi member
Status_member	enum('0', '1')	-		

c. Tabel Barang

Tabel III.26 Tabel Barang

abel : Barang				
Key : Id_Barang				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_Barang	Integer	11	PK	Primary Key
id_kategori	Integer	11	FK	Foreign Key, reference table kategori (id_ kategori)
Nama_barang	Varchar	50		Nama Barang
Harga_barang	Integer	11		Harga barang
Diskon_barang	Int	11		Diskon barang
Rating_barang	Float			Rating barang
Voterrating_barang	Integer	11		Vote rating barang
*View*counter_barang	Integer	11		*View* counter barang

d. Tabel Gambar

Tabel III.27 Tabel Gambar

abel : Gambar				
Key : id_gambar				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
id_Gambar	Integer	11	PK	Primary Key
Id_detailbarang	Integer	11	FK	Foreign Key, reference table barang (id_detailbarang)
Path_gambar	Varchar	100		Gambar tampak depan
Path_gambar2	Varchar	100		Gambar tampak samping

e. Tabel Detail Barang

Tabel III.28 Tabel Detail Barang

Nama Tabel : Detailbarang				
Primary Key : id_detailbarang				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
id_detailbarang	Integer	11	PK	Primary Key
Id_barang	Integer	11	FK	Foreign Key, reference table barang (id_ barang)
Id_warna	Integer	5		Foreign Key, reference table barang (id_ warna)
Id_Ukuran	Integer	5		Foreign Key, reference table barang (id_ Ukuran)
Tanggal_detailbarang	Datetime	50		Tanggal pemasukan barang
Stok_detailbarang	Integer	5		Stok barang
Berat_detailbarang	Float	2,2		Berat barang

f. Tabel Detail Pembelian

Tabel III.29 Tabel Detail Pembelian

Nama Tabel : detail_pembelian				
Primary Key : Id_ Pembelian				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_detail_pembelian	Integer	11	PK	Primary Key
Id_pembelian	Integer	11	FK	Foreign Key, reference table Detailpembelian (id_ Detailpembelian)
Id_detailbarang	Integer	11	FK	Foreign Key, reference table detailbarang (id_ detailbarang)
Harga	Int	11		Harga barang
jumlah	Int	5		Stok barang
Berat	Float	2,2		Berat barang

g. Tabel Pembelian

Tabel III.30 Tabel Pembelian

Nama Tabel : Pembelian				
Primary Key : id_pembelian				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
id_pembelian	Integer	11	PK	Primary Key
Id_member	Integer	11	FK	Foreign Key, reference table member (id_ member)
jenis_pembayaran	Enum	‘paypal’, ’atm’, ’rekening’, ’internetbanking’		Jenis pembayaran yang dipilih
status_pengiriman	Enum	‘dipesan’,’dibayar’,’lunas’,’dikirim’, ’diterima’		Status pengiriman
nama_pemesan	varchar	50		Nama pemesan
alamat_pemesan	Varchar	80		Alamat pemesan
email_pemesan	varchar	50		Email pemesan
no_telp_pemesan	varchar	20		No telp pemesan
kodepos_pemesan	varchar	6		Kodepos pemesan
nama_bank_pemesan	varchar	50		Nama bank pemesan
no_rekening_pemesan	varchar	20		No rekening pemesan
Nama_pengirim_transfer	Varchar	50		Nama pengirim transfer
Total_kirim	Float	10,2		
Tgl_beli	date	-		Tanggal pesan
Tgl_bayar	Date	-		Tanggal transfer
Tgl_terima	Date	-		Tanggal terima
no_resi_pemesan	varchar	32		No resi pemesan
Id_rekening	Integer	5	FK	Foreign Key, reference table rekening (id_ rekening)
Id_ongkoskirim	Integer	5	FK	Foreign Key, reference table ongkoskirim (id_ ongkoskirim)
status_pembelian	enum	‘ok’,’cancel’		Status pembelian
Session_id	Varchar	32		Session id

h. Tabel Ukuran

Tabel III.31 Tabel Ukuran

Nama Tabel : Ukuran				
Primary Key : id_ukuran				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
id_ukuran	Integer	11	PK	Primary Key
Nama_Ukuran	Varchar	10		Nama ukuran
Deskripsi_ukuran	Text	-		Keterangan ukuran

i. Tabel Kategori

Tabel III.32 Tabel Kategori

Nama Tabel : kategori				
Primary Key : id_kategori				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
id_kategori	Integer	11	PK	Primary Key
Parent_kategori	Integer	11		
Nama_kategori	Varchar	50		Nama kategori
kode_kategori	Varchar	3		Kode kategori

j. Tabel Ongkos Kirim

Tabel III.33 Tabel Ongkos Kirim

Nama Tabel : OngkosKirim				
Primary Key : id_ongkoskirim				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
id_Ongkoskirim	Integer	11	PK	Primary Key
Id_kota	Integer	11	FK	Foreign Key, reference table kota (id_ kota)
Id_Jenispengiriman	Integer	11	FK	Foreign Key, reference table Jenispengiriman (id_ Jenispengiriman)
Harga_ongkoskirim	Integer	11		Harga Ongkos kirim

k. Tabel Kota

Tabel III.34 Tabel Kota

Nama Tabel : Kota				
Primary Key : id_Kota				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
id_Kota	Integer	10	PK	Primary Key
Id_propinsi	Integer	10	FK	Foreign Key, reference table propinsi (id_ propinsi)
Nama_kota	Varchar	50		Nama kota

l. Tabel Provinsi

Tabel III.35 Tabel Provinsi

Nama Tabel : Provinsi				
Primary Key : id_Provinsi				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
id_Provinsi	Integer	10	PK	Primary Key
Nama_provinsi	Varchar	30		Nama provinsi

m. Tabel Jasa Pengiriman

Tabel III.36 Tabel Jasa Pengiriman

Nama Tabel : Jasapengirirman				
Primary Key : id_jasapengirirman				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_jasapengirirman	Integer	11	PK	Primary Key
Nama_jasapengiriman	Varchar	50		Nama jasa pengiriman
Deskripsi_jasapengiriman	Text			Keterangan pengiriman

n. Tabel Jenis Pengiriman

Tabel III.37 Tabel Jenis Pengiriman

Nama Tabel : Jenispengirirman				
Primary Key : id_Jenispengirirman				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_Jenispengirirman	Integer	11	PK	Primary Key
Id_jasapengiriman	Integer	11	FK	Foreign Key, reference table jasapengiriman (id_ jasapengiriman)
Nama_jenis	Varchar	50		Nama jenis pengiriman
Deskripsi	Text			Keterangan pengiriman

o. Tabel Retur

Tabel III.38 Tabel Retur

Nama Tabel : Retur				
Primary Key : id_Retur				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_Retur	Integer	11	PK	Primary Key
Id_pembelian	Integer	11	FK	Foreign Key, reference table invoice (id_ jasapengiriman)
Id_member	Integer	11	FK	Foreign Key, reference table member (id_member)
Tgl_kirim	Date			Tanggal kirim
Resi_kirim	Varchar	20		Resi kirim
Tgl_retur	Date			Tanggal retur
Resi_retur	Varchar	20		Resi retur
Status_retur	Varchar	20		Status retur

p. Tabel Rekening

Tabel III.39 Tabel Rekening

Nama Tabel : Rekening				
Primary Key : id_Rekening				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_Rekening	Integer	11	PK	Primary Key
Nama_rekening	Varchar	50		Nama rekening
Bank_rekening	Integer	11		Bank rekening
Cabang_rekening	Varchar	100		Cabang rekening
No_rekening	Varchar	32		No rekening
Gambar_rekening	Varchar	200		Gambar rekening

q. Tabel Warna

Tabel III.40 Tabel Warna

Nama Tabel : warna				
Primary Key : id_warna				
Nama field	**Tipe data**	**Ukuran**	**Kunci**	**Keterangan**
Id_warna	Integer	11	PK	Primary Key
Nama_warna	Varchar	25		warna barang