BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis Sistem adalah menjelaskan keadaan sistem ke dalam bagian-bagiannya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang ada.

3.1.1 Analisis Masalah

Langkah pertama yang harus dilakukan dalam tahap analisis sistem adalah mengidentisifikasi masalah yang terjadi. Masalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai.

Berdasarkan *observasi* dan *interview* dengan pihak Mitra Teknik, didapatkan kesimpulan permasalahan yaitu media pemasaran masih terbatas dikarenakan dilakukan secara manual dengan membagikan kartu nama kepada para calon pembeli, menghadiri acara-acara khusus menembak yang sering diselenggarakan di kota-kota diluar Kota Bandung sehingga kesulitan untuk mencakup wilayah tersebut karena membutuhkan waktu dan biaya tambahan, wilayah penjualan saat ini masih terbatas membuat pemilik ingin mengembangkan wilayah cakupan usaha yang lebih luas serta banyaknya arsip maupun laporan yang berdifat manual sehingga sering terjadinya kehilangan data.

Berdasarkan permasalahan yang telah diuraikan maka perlu dibangunnya sistem yang terkomputerisasi untuk memaksimalkan dalam proses penjualannya.

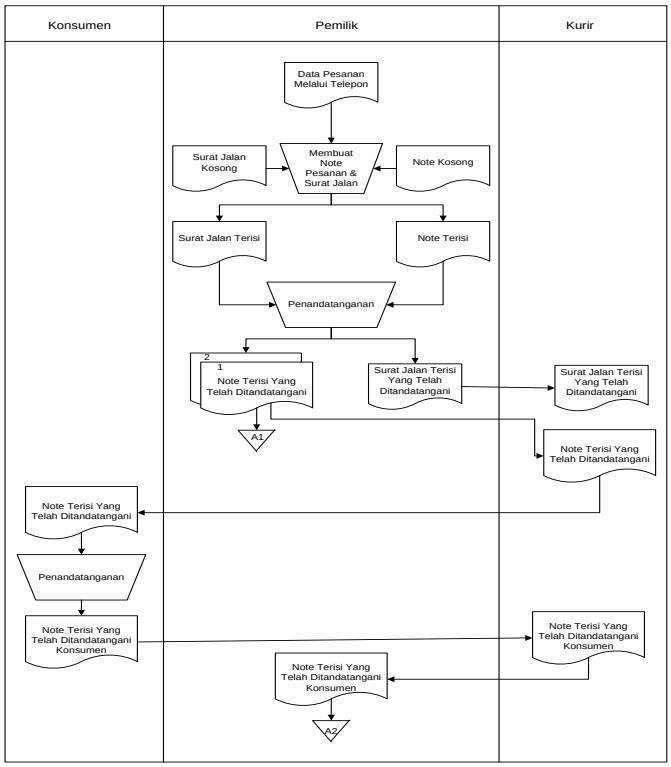
3.1.2 Analisis Prosedur Sistem yang Sedang Berjalan

Prosedur adalah urutan kegiatan yang sedang berjalan sesuai dengan proses dan tahapan yang ada yang menjelaskan mengenai proses apa yang dikerjakan, siapa yang mengerjakan proses tersebut, bagaimana proses tersebut dikerjakan dan dokumen apa saja yang terlibat dalam proses tersebut.

Berdasarkan hasil wawancara dengan pemilik Mitra Teknik prosedur penjualan barang yang sedang berjalan di Mitra Teknik adalah sebagai berikut :

1. Konsumen menelepon pemilik untuk memesan barang.
2. Pemilik membuat nota pesanan yang dibuat rangkap dua dan menbuat surat jalan.
3. Nota pesanan dan surat jalan ditandatangani oleh pemilik selanjutnya diberikan kepada kurir.
4. Kurir mengantarkan barang dan nota pesanan kepada konsumen.
5. Konsumen mendatangani nota pesanan dan diberikan kembali kapada kurir.
6. Kurir memberikan nota pesanan yang telah ditandatangani konsumen kepada pemilik.

Untuk lebih jelasnya dapat dilihat pada gambar diagram alir data prosedur penjualan barang pada gambar 3.1 berikut.



Gambar 3.1 *Flowmap* Penjualan Barang

Keterangan :

A1 : Arsip Nota Pesanan

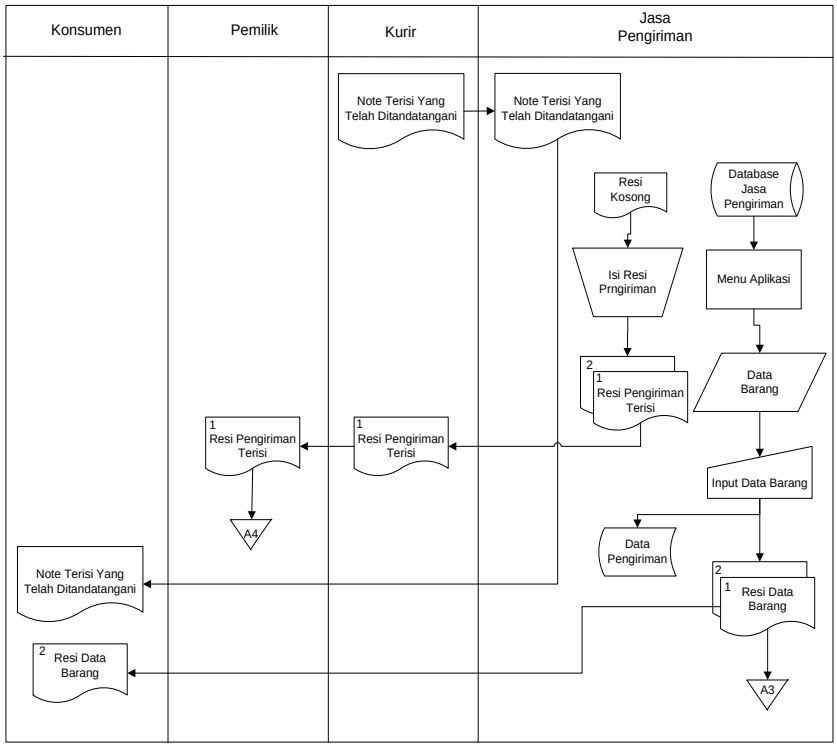
A2 : Arsip Nota Pesanan Yang Telah ditandatangani Konsumen

3.1.1.2 Prosedur Pengiriman Barang melalui Jasa Pengiriman

Berdasarkan hasil wawancara dengan pemilik Mitra Teknik prosedur pengiriman melalui jasa pengiriman barang adalah sebagai berikut :

1. Kurir memberikan nota pesanan dan barang kepada jasa pengiriman.
2. Jasa Pengiriman membuat resi pengiriman data barang kemudian diberikan kepada kurir.
3. Jasa pengiriman membuat resi barang untuk dilampirkan bersama barang dan nota pesanan.
4. Kurir memberikan resi pengiriman kepada pemilik Mitra Teknik.

Untuk lebih jelasnya dapat dilihat pada gambar diagram alir data prosedur pengiriman barang melalui jasa pengiriman berikut.



Gambar 3.2 Flowmap Pengiriman Barang

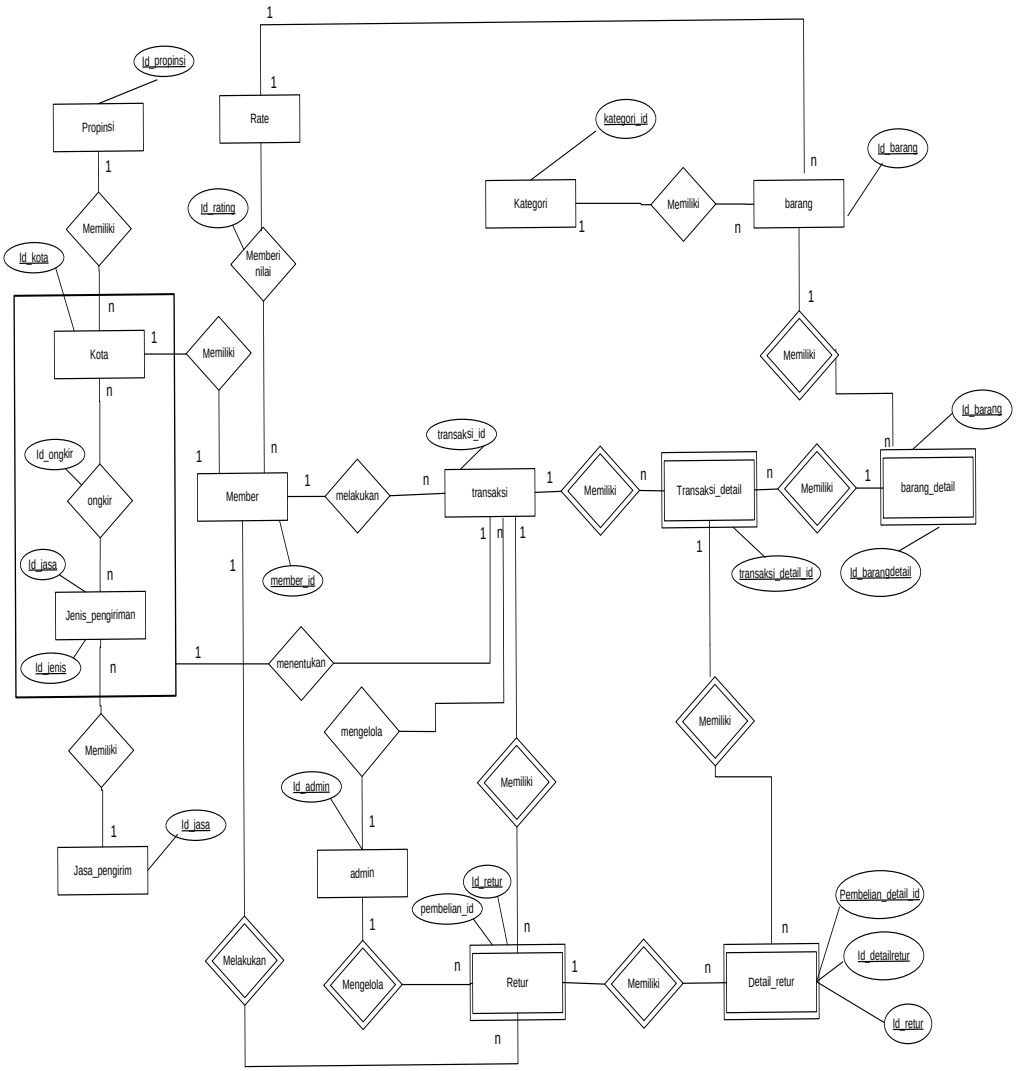
Keterangan :

A3: Arsip Resi Barang Milik Jasa Pengiriman

A4 : Arsip Resi Pengiriman Barang

3.1.3 ERD (Entity Relationship Diagram)

*Entity Relationship Diagram*atau ERD digunakan sebagai alat bantu dalam memodelkan data dan menghubungkan relasi antar data pada sistem *e-commerce* penjualan produk. Anaslisis basis data pembuatan *e-commerce* dituangkan dalam gambar E-R pada gambar 3.3 berikut ini.



Gambar 3.3 ERD (Entity Relationship Diagram)

3.1.4 Kamus Data

No	Nama Entitas	Nama Atribut
1	Member	customer_id, nama_lengkap, tempat_lahir, tanggal_lahir, jenis_kelamin, alamat, provinsi, kabupaten_kota, kodepos, telepon, handphone, paypall, nama_bank, nama_rekening, no_rekening, user_name, password, foto, email, status_confirm
2	Transaksi	transaksi_id, tanggal_transaksi, customer, nama_customer, nama_bank, no_rekening, atas_nama, ongkos_kirim, diskon, total_transaksi, total_transaksi_dollar, jenis_pengiriman, jenis_paket, kota_tujuan, alamat_tujuan, jenis_pembayaran, tanggal_pembayaran, status, status_baca, resi_jne
3	transaksi_detail	transaksi_detail_id, transaksi, barang, nama_barang, harga, jumlah, subtotal
4	Barang	barang_id, kategori, sub_kategori, kode_barang, nama_barang, berat, harga, stok, deskripsi_barang, rekomendasi, status_aktif, status_barang, gambar_1, gambar_2, gambar_3, gambar_4,
5	Trate	customer_id, barang_id, rate
6	Kota	id, propinsi_id, city_name
7	Propinsi	id_propinsi, nama_propinsi
8	Paket	paket_id, nama_paket, deskripsi_paket
19	paket_detail	paket_detail_id, paket, nama_paket_detail, deskripsi_paket_detail
10	Retur	id_retur, customer_id, transaksi_id, nama_barang, keterangan_retur, tanggal_retur, tanggal_kirim_balik, status
11	ongkos_kirim	ongkir_id, paket, paket_detail, kota_asal, kota_tujuan, tarif
12	sub_kategori	sub_kategori_id, kategori, nama_sub_kategori, deskripsi_sub_kategori
13	Kategori	kategori_id, nama_kategori, deskripsi_kategori
14	Diskon	diskon_id, judul, tanggal_awal, tanggal_akhir, minimal_beli, besar_diskon, status_diskon, deskripsi

3.1.3 Aturan Bisnis Sistem Yang Akan Dibangun

1. Untuk melakukan pemesanan produk pembeli harus terlebih dahulu mendaftar menjadi member di Mitra Teknik. Pada saat pendaftaran harus memilih salah satu status keanggotaan yaitu member dan reseller.
2. Untuk member dengan status reseller memiliki harga yang berbeda dibandingan dengan member biasa.
3. Pengiriman barang dilakukan setelah member malakukan konfirmasi pembayaran.
4. Batas waktu konfirmasi pembayaran adalah 1 X 24 jam.
5. Retur dilakukan jika barang rusak atau tidak berjalan sebagaimana fungsinya bukan karena rusak dalam proses pengiriman dan batas waktu proses retur barang ditentukan oleh pemilik toko.
6. Untuk pengelolaan stok yaitu stok akan berkurang ketika member melakukan pemesanan barang. Stok akan dikembalikan ke stok semula jika member membatalkan barang, pemesanan dibatalkan oleh admin dan ketika pemesanan dibatalkan oleh sistem karena melebihi batas waktu pembayaran.
7. Penetapan diskon dilakukan oleh pemilik toko.

3.2 Analisis Metode Item Based Collaborative Filtering

Pemanfaatan sistem rekomendasi cerdas menggunakan metode *Item based collaborative filtering* untuk membantu pelanggan untuk membeli produk yang belum dibeli sedangkan produk tersebut banyak dibeli oleh pelanggan yang lainnya dan memiliki rating tinggi yang sekiranya akan membuat pelanggan yang belum membeli produk tersebut tertarik untuk membelinya. *Item-based collaborative filtering* merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian rating terhadap suatu produk dengan produk yang dibeli,pembangkitan rekomendasi akan dihitung dengan formula :

$$R = \left(\frac{\sum B * rating * 100}{Max * MaxB}\right)/C$$

Keterangan :

B : Nilai pembelian produk (1 jika membeli & 0 jika tidak membeli).

Rating : Jumlah rating yang diberikan oleh pelanggan ke produk yang disukai.

100 : Nilai maksimum 100, nilai rekomendasi tert inggi adalah 100.

Max : Nilai maksimum rating bernilai 5.

MaxB : Nilai maksimum pembelian yang dilakukan oleh pelanggan, jika pelanggan membeli lebih dari satu produk, dalam formula ini tetap akan dihitung membeli satu produk.

C : Jumlah pengguna yang mempunyai pola pembelian atau peratingan yang sama dengan pelanggan yang login pada sistem.

Contoh :

Tabel 3.1 Pembangkit Rekomendasi

Kode Produk	M1	M2	M3	M4	M5
ST01	1^4	0	1^3	0	1^4
ST02	0	1^4	0	1^5	1^5
ST03	1^5	0	1^5	0	1^3
ST04	0	1^2	1^5	1^4	1^2
ST05	0	1^3	1^5	0	1^2

ST01-ST05= Kode produk pada Mitra Teknik, M1-M5 = member yang telah terdaftar. Angka 1 dan 0 = 1 jika pelanggan tersebut membeli produk MITRA TEKNIK, dan 0 jika tidak melakukan pembelian produk MITRA TEKNIK.

Angka Pangkat (1-5) = rating produk yang sudah diberikan oleh pelanggan.

Skala Rating = 1 (sangat tidak menarik), 2 (tidak menarik), 3 (cukup menarik), 4 (menarik), dan 5 (sangat menarik).

Jika pelanggan dengan kode M2 login dan melakukan pembelian atau peratingan pada sistem maka perhitungan pembangkit rekomendasi adalah sebagai berikut :

Rating produk ST01 = (((1*4*100)/5*1)+0+((1*3*100)/5*1)+0+

((1*4*100)/5*1))/4= 55

Rating produk ST02 = (0+((1*4*100)/5*1)+0+((1*5*100)/5*1)+

((1*5*100)/5*1)+)/4 = 70

Rating produk ST03 = (((1*5*100)/5*1)+0+((1*5*100)/5*1)+0+

((1*3*100)/5*1))/4 = 60

Rating produk ST04 = (0+((1*2*100)/5*1)+

((1*5*100)/5*1)+ +((1*4*100)/5*1)+((1*2*100)/5*1))/4 = 65

Rating Produk ST05 = 0+((1*3*100)/5*1)+(((1*5*100)/5*1)+0+

((1*2*100)/5*1))/4 = 50

Jadi rekomendasi produk untuk pelanggan P1akan ditunjukkan pada Tabel 3.2 berikut :

Tabel 3.2 Hasil Rekomendasi

NO	Kode Produk	Nilai Rekomenda
1	ST02	70
2	ST04	65
3	ST03	60

4	ST01	55
5	ST05	50

Namun dilihat dari data pembelian produk pelanggan M2 sudah membeli produk dengan kode ST02 dan ST05 sehingga sistem akan merekomendasikan produk :

Tabel 3.3 Rekomendasi Produk

NO	Kode Produk	Nilai Rating
1	ST04	65
2	ST03	60
3	ST01	55

3.3 Analisis SEO (Search Engine Optimization)

SEO adalah singkatan dari *search engine optimization* yang secara arti kata adalah sebuah proses yang dilakukan oleh pemilik website agar websitenya bisa optimal di search engine, sehingga website tersebut bisa masuk di urutan atas di hasil pencarian search engine dengan kata kunci tertentu.

Berikut contoh penerapan meta tag di halaman website (index.html) :

```
<html>

<head>

    <title>Mitra Teknik</title>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <meta name="description" content=" Mitra Teknik merupakan usaha yang bergerak di bidang penjualan alat-alat perlengkapan menembak ">
```

<meta name="author" content="Mitra Teknik">

<meta name="keywords" content=" aksesoris menembak, sasaran tembak, silhouette ">

</head>

</html>

3.4 Analisis SSL

3.4.1 Cara *Install SSL* di *cPanel*

Berikut adalah langkah-langkah menginstall SSL di cpanel :

Langkah 1: Siapkan Email *Approver*

Sebelum meregistrasi SSL Siapkan *email* untuk *approval SSL* yang biasanya di kirim dari penerbit SSL, tanpa email ini kita tidak akan mendapatkanSSL resmi. Contoh email yang dibutuhkan : mitrateknikbandung@gmail.com

Langkah 2: Buat*Key* 2048bit

Pertana perlu membuat *keypad hosting*. Minimun nilai adalah 2048. Kemudian *Login* kecpanel pilih SSL/TLS *Manager* klik *Generate, view, upload, or delete your private keys.* Pada kolom *Host* masukannama domain yang ingin di instal SSL misalnya: tomassutrisno.com kemudian klik *Generate*.

Langkah 3: Buat*CSR (Certificate Signing Request*)

Setelah membuat *Key* saatnya mengenerate CSR, CSR ini digunakan untuk membeli SSL keperusahaan penerbit SSL. pada SSL/TLS *Manager* di cpanel klik *Generate, view, or delete SSL certificate signing requests*. Apabila *Langkah* ini sudah dilakukan, cpanel akan memberikan sebuah kode CSR.

Langkah 4: *Install* SSL

Setelah mendapatkan SSL resmi dari penerbit, kini saatnya untuk menginstall SSL pada *hosting, .upload* CRT pada kolom isian *Certificate* pada cPanel. Klik *Generate, view, upload, or delete SSL certificates*

Langkah 5: Aktifasi SSL ke HTTPS

Semua sudah selesai.kini saatnya mengaktifkan fitur HTTPS pada website. Pada SSL/TLS *manager* klik *Setup a SSL certificate to work with your site.* Pilih nama domain yang ingin di setup SSLnya. misal: tomassutrisno.com, kemudian klik *Install Certificate* yang terdapat di bagian bawah kolom. Apabila *install* berhasil silahkan untuk mencoba akses https://tomassutrisno.com pada web browser.

3.5 Analisis Kebutuhan Non Fungsional

Analisa kebutuhan non fungsional menggambarkan keadaan sistem yang ada di Mitra Teknik diantaranya perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.5.1 Analisis Perangkat Keras (Hardware)

Analisis perangkat keras (hardware) merupakan proses analisis yang lebih menekankan kepada aspek pemanfaatan perangkat keras yang selama ini telah dimiliki Mitra teknik.

Adapun spesifikasi perangkat keras komputer yang ada di Mitra Teknik adalah sebagai berikut :

1. Processor AMD Phenom II 3,4 GHz
2. Kapasitas RAM 2 Gb
3. Kapasitas Hardisk 500 Gb
4. Monitor ukuran 14” dengan resolusi 1024 x 768
5. *Keyboard* dan *mouse.*
6. Printer.

3.5.2 Analisis Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan di Mitra Teknik adalah sebagai berikut :

1. Sistem Operasi Windows 7 Ultimate 32-bit

2. Microsoft Office 2007
3. Web browser Mozilla firefox
4. Code Editor Macromedia Dreamweaver 8
5. Mysql versi 5.0
6. Web server XAMPP 2.5.8

3.5.3 Analisis Pengguna Saat Ini

Berdasarkan hasil penelitian, dapat dipaparkan pengguna yang terlibat dalam sistem yang sedang berjalan yaitu admin,members dan pengunjung.

1. Admin

Admin berfungsi untuk menjaga kebenaran data yang terkandung didalam *database*, menentukan pengguna berserta hak aksesnya, atau bahkan dalam kasus tertentu bisa melakukan manipulasi terhadap data yang ada. Karakteristik yang harus dimiliki adalah mengerti dan menguasai komputer dengan baik, mengetahui bagaimana melakukan pengolahan dan perawatan terhadap *database* yang ada pada *website*, dan memiliki pengetahuan tentang internet.

2. Member

Members berguna untuk melakukan pembelian barang secara *online* di Mitra Teknik setelah melakukan pendaftaran. Karakteristik yang dimiliki adalah memiliki kemampuan di bidang komputer dan dapat melakukan *browsing* diinternet.

3. Pengunjung

Pengunjung adalah pengguna biasa yang hanya dapat melihat konten yang ada di *website* tanpa dapat melakukan pemesanan. Karakteristik yang dimiliki adalah memiliki kemampuan di bidang komputer dan dapat melakukan *browsing* di internet. Dapat diambil kesimpulan bahwa pengguna yang ada pada Mitra Teknik sudah terbiasa menggunakan komputer, dan cukup memenuhi syarat sebagai pengguna sistem yang akan dikembangkan, sehingga tidak diperlukan pelatihan khusus mengenai penggunaan komputer, cukup berupa dokumen atau buku panduan untuk membantu menjalankan aplikasi

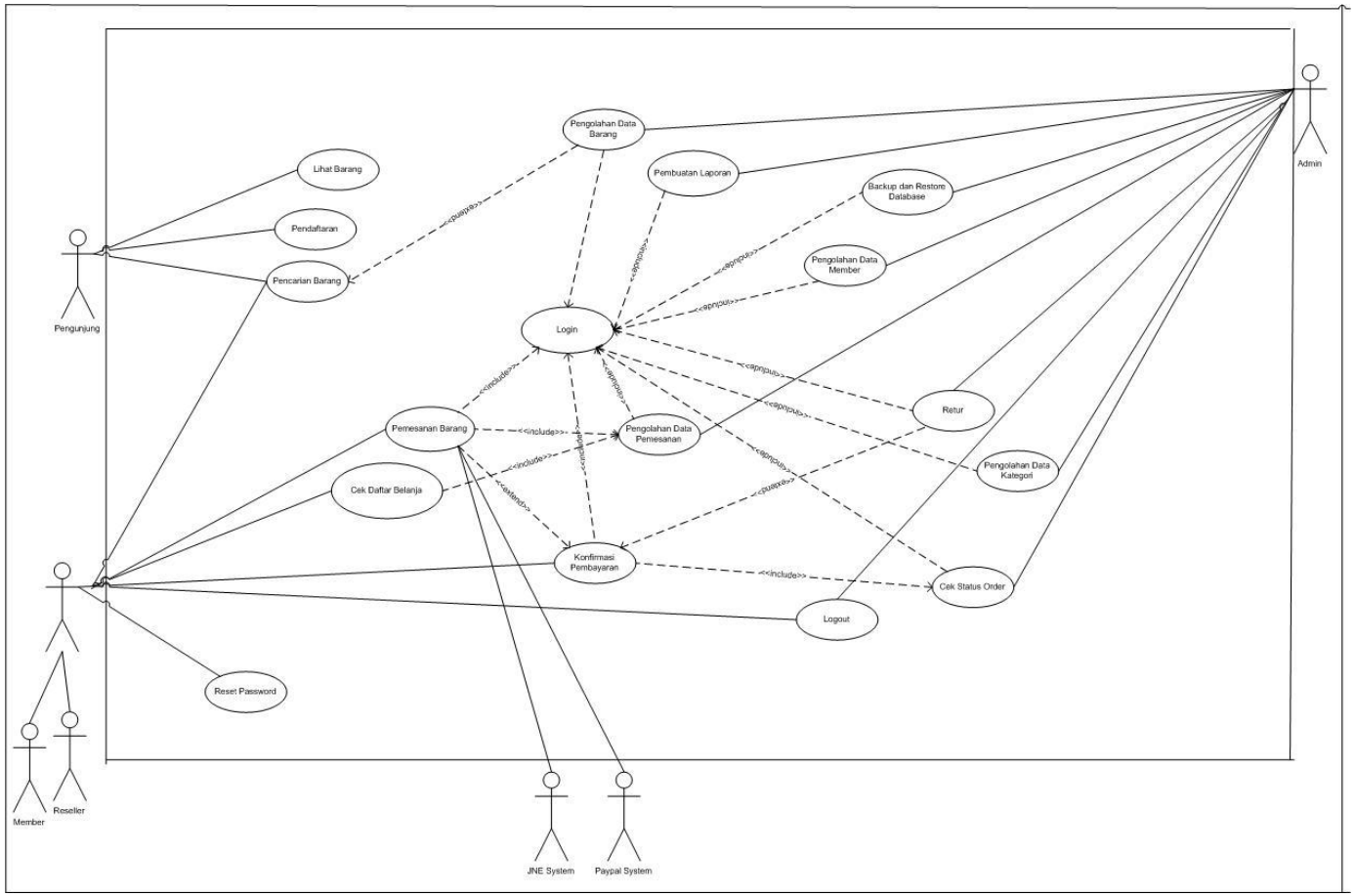
3.6 Identifikasi Aktor

Aplikasi e-commerce akan digunakan oleh empat aktor yaitu, admin, pengunjung, dan member (member dan reseller). Deskripsi masing-masing aktor adalah sebagai berikut :

1. Aktor pertama adalah admin adalah sebagai pengelola yang memiliki semua hak akses yang ada pada sistem.
2. Aktor kedua adalah pengunjung yang hanya bisa melihat dan melakukan pencarian barang.
3. Aktor ketiga adalah member yang memiliki akses untuk melakukan transaksi pembelian. Member mempunyai dua status yaitu, member dan reseller.
4. Aktor keempat adalah jasa pengiriman yang terleibat secara tidak langsung ke dalam sistem.

3.7 Use Case Diagram

Use case merupakan interaksi antara aktor dan sistem dan manggamberikan segala aktivitas yang dilakukan oleh sistem dari sudut pandang terluar.



Gambar 3.4 *Use Case Diagram E-Commerce* Untuk Mitra Teknik

3.7.1 Use Case Scenario

Use Case Scenario menjelaskan interaksi antara aktor dan *use case* dalam melakukan sebuah proses tertentu. Berikut adalah *use case scenario* E-Commerce Mitra teknik.

1. Skenario Use Case Login

Interaksi antara aktor admin dan *membe*r dengan *use case login* dapat dilihat pada tabel 3.4

Tabel 3.4 Skenario *Use Case Login*

Identifikasi			
Nama	*Login*		
Tujuan	Masuk ke dalam sistem		
Deskripsi	Proses Login merupakan proses autentifikasi untuk menggunakan hak akses sebagai administrator atau member menggunakan sistem		
Aktor	*Admin, Member*		
***Use Case* yang Berkaitan**	*Pemesanan Barang, Pengolahan Data Barang, Konfirmasi Pembayaran, Pengolahan Data Pemesanan, Pengolahan Data Barang, Back Up Dan Restore Database, Pembuatan Laporan, Pengolahan Data Katagori, Pengolahan Data Konfirmasi, Retur, Pengaturan Akun.*		
Skenario Utama			
Kondisi Awal	Sistem menampilkan form *login*		
Aksi Aktor		**Reaksi Sistem**	
1	Mengisi form *login*	2	Mengecek data pada tabel di *database*
		3	Bila data ada dan cocok maka akan masuk ke dalam sistem dan menampilkan menu utama sesuai hak akses

Skenario Alternatif (Autentifikasi Gagal)			
Aksi Aktor		**Reaksi Sistem**	
		4	Menampilkan pesan yang menjelaskan bahwa data *login* salah
5	Mengisi kembali form *login*	6	Mengecek data akun di *database*
		7	Bila akun ada dan cocok maka akan masuk ke dalam sistem dan menampilkan menu utama sesuai hak akses
Kondisi Akhir		Berhasil masuk ke dalam sistem dan menampilkan menu utama	

2. Skenario *Use Case* Reset Password

Interaksi antara aktor admin, dan *member* dengan *use case* reset password dapat dilihat pada tabel 3.5.

Tabel 3.5 Skenario *Use Case* Reset Password

Identifikasi	
Nama	Permintaan Reset Password
Tujuan	Mereset *password* aktor
Deskripsi	Proses untuk mereset *password,* memberitahu admin untuk memberikan *password* baru dan mengirimnya ke email yang diberikan pada saat pendaftaran
Aktor	*Member*
***Use Case* yang Berkaitan**	-
Skenario Utama	

Kondisi Awal	Memilih menu reset *password*		
Aksi Aktor		**Reaksi Sistem**	
1	Mengisi form email atau *username* yang sebelumnya didaftarkan	2	Mencari data berdasarkan email atau *username* yang diisikan, kemudian mereset *password* dan mengirimnya ke email yang ada pada *database*
Kondisi Akhir	*Password* telah direset dan dikirim ke email		

3. Skenario Use Case Logout

Interaksi antara aktor admin, dan member dengan *use case logout* dapat dilihat pada tabel 3.6.

Tabel 3.6 Skenario *Use Case Logout*

Identifikasi			
Nama	*Logout*		
Tujuan	Keluar dari sistem		
Deskripsi	Merupakan proses berhenti menggunakan aplikasi dan keluar dari sesi *login*		
Aktor	*Admin, Member*		
***Use Case* yang Berkaitan**	-		
Skenario Utama			
Kondisi Awal	Sistem menampilkan menu utama sesuai hak akses		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu keluar	2	Menghentikan akses dan kembali ke menu sebelum *login*
Kondisi Akhir	Berhasil *logout* dari sistem		

4. Skenario *Use Case* Pendaftaran Member

Interaksi antara aktor pengunjung dengan *use case* pendaftaran member dapat dilihat pada tabel 3.7.

Tabel 3.7 Skenario *Use Case* Pendaftaran Member

Identifikasi			
Nama	*Pendaftaran Member*		
Tujuan	Melakukan pendaftaran *member* dan *reseller*		
Deskripsi	Proses untuk registrasi menjadi *member* dan *reseller*		
Aktor	Pengunjung		
***Use Case* yang Berkaitan**	-		
Skenario Utama			
Kndisi Awal	Menampilkan halaman beranda (*home*)		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu daftar	2	Menampilkan form pendaftaran member
3	Mengisi form pendaftaran member	4	Memvalidasi data masukan, jika data tidak valid maka tampil pesan kesalahan dan menampilkan kembali form pendaftaran, jika data valid maka data akan disimpan dan mengirimkan email konfirmasi registrasi berhasil.
Kondisi Akhir	Pengunjung berhasil mendaftar sebagai *member* dan *reseller*		

5. Skenario *Use Case* Lihat Barang

Interaksi antara aktor pengunjung dan *member* dengan *use case* lihat barang dapat dilihat pada tabel 3.8.

Tabel 3.8 Skenario *Use Case* Lihat Barang

Identifikasi			
Nama	Lihat Barang		
Tujuan	Melihat info barang		
Deskripsi	Merupakan proses untuk melihat info barang yang tersedia		
Aktor	Pengunjung, *Member*		
***Use Case* yang Berkaitan**	-		
Skenario Utama			
Kondisi Awal	Menampilkan halaman beranda (*home*)		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih barang yang akan ditampilkan	2	Mengambil data barang dari *database* kemudian menampilkannya
Kondisi Akhir	Halaman barang berhasil ditampilkan		

6. Skenario *Use Case* Pencarian Barang

Interaksi antara aktor pengunjung dan member dengan *use case* pencarian barang dapat dilihat pada tabel 3.9.

Tabel 3.9 Skenario *Use Case* Pencarian Barang

Identifikasi	
Nama	Pencarian Barang
Tujuan	Mencari dan Melihat Info Barang
Deskripsi	Proses untuk mencari data barang kemudian menampilkannya
Aktor	Pengunjung, *Member*

***Use Case* yang Berkaitan**	-		
Skenario Utama			
Kondisi Awal	Menampilkan halaman beranda (*home*)		
Aksi Aktor		**Reaksi Sistem**	
1	Mengetikkan data barang yang akan dicari	2	Mencari data barang di *database*, jika data ditemukan maka akan menampilkan list data barang, jika data tidak ditemukan maka akan menampilkan pesan data yang dicari tidak ditemukan
Kondisi Akhir	Menampilkan hasil pencarian barang		

7. Skenario *Use Case* Pemesanan Barang

Interaksi antara aktor member dengan *use case* Pemesanan barang dapat dilihat pada tabel 3.10.

Tabel 3.10 Skenario *Use Case* Pemesanan Barang

Identifikasi	
Nama	Pemesanan Barang
Tujuan	Melakukan pemesanan barang
Deskripsi	Proses untuk memilih barang yang akan dipesan
Aktor	*Member*
***Use Case* yang Berkaitan**	*Jasa Pengiriman, Paypall, Pengolahan Data Pemesanan,Login*
Skenario Utama	
Kondisi Awal	Menampilkan halaman beranda (*home*)

Aksi Aktor		Reaksi Sistem	
1	Memilih kategori dan barang yang akan dipesan	2	Mengecek apakah member sudah login, jika sudah login maka akan menampilkan form pemesanan barang, jika belum login maka akan menampilkan pesan harus login
3	Menekan tombol tambahkan		
4	Memasukkan jumlah barang yang akan dipesan	5	Menyimpan data pemesanan barang pada keranjang
7	Menekan tombol keranjang belanja	8	Menampilkan form daftar pemesanan
9	Mengisi alamat pengiriman dan memilih jenis paket	10	Menampilkan jenis paket pengiriman
11	Menekan simpan pemesanan	12	Menyimpan data barang yang dipesan dan data pengiriman ke *database*
Kondisi Akhir	Member berhasil melakukan pemesanan barang		

8. Skenario *Use Case* Konfirmasi pembayaran

Interaksi antara aktor member dengan *use case* konfirmasi pembayaran dapat dilihat pada tabel 3.11.

Tabel 3.11 Skenario *Use Case* Konfirmasi pembayaran

Identifikasi	
Nama	Konfirmasi pembayaran
Tujuan	Melakukan konfirmasi pembayaran untuk sebuah order
Deskripsi	Proses konfirmasi pembayaran order untuk pembayaran yang dilakukan melalui transfer bank dan paypall

Aktor	*Member*		
***Use Case* yang Berkaitan**	*Login, Pemesanan Barang*		
Skenario Utama			
Kondisi Awal	Menampilkan menu aberanda (home)		
Aksi		**Reaksi Sistem**	
1	Memilih menu transfer bank	2	Menampilkan form pembayaran via bank
3	Mengisi form konfirmasi pembayaran meliputi no. rekening, nama bank, atas nama, tanggal dan waktu pembayaran	4	Menyimpan data masukan ke dalam database.
		5	Menyimpan data konfirmasi pembayaran ke dalam database
6	Menampilkan status tunggu konfirmasi.		
Skenario Alternatif (Pembayaran Melalui Paypall)			
7	Memilih Menu Paypall	8	Menampilkan form pembayaran dari system paypall
9	Memilih menu bayar sekarang	10	Menempilkan pesan telah melakukan pembayaran
Kondisi Akhir	Member berhasil melakukan konfirmasi pembayaran		

9. Skenario *Use Case* Cek Status *Order*

Interaksi antara aktor member dengan *use case* Cek Status *Order* dapat dilihat pada tabel 3.12.

Tabel 3.12 Skenario *Use Case* Cek Status *Order*

Identifikasi			
Nama	Cek Status Order		
Tujuan	Melihat status order		
Deskripsi	Proses mengecek data order pada database dan menampilkannya kepada member		
Aktor	*Member*		
***Use Case* yang Berkaitan**	*Konfirmasi pembayaran, pemesanan barang*		
Skenario Utama			
Kondisi Awal	Menampilkan menu beranda (home)		
Aksi		**Reaksi Sistem**	
1	Memilih menu daftar belanja	2	Mencari data order pada database Jika data ditemukan maka akan ditampilkan data pemesanan
Skenario Alternatif (Data order tidak ditemukan)			
		3	Menampilkan pesan data order tidak ditemukan
4	Mengisi kembali form cek status order	5	Mencari data order pada database berdasarkan order ID, Jika data ditemukan maka akan ditampilkan. berhasil
Kondisi Akhir	Menampilkan status data order		

10. Skenario *Use Case* Pengolahan Data *Member*

Interaksi antara aktor admin dengan *use case* pengolahan data *member* dapat dilihat pada tabel 3.13 .

Tabel 3.13 Skenario *Use Case* Pengolahan Data *Member*

Identifikasi			
Nama	Pengolahan Data Akun Member		
Tujuan	Mengelola Data Member pada sistem		
Deskripsi	Proses pengolahan data member merupakan proses untuk mengatur dan mengolah data member pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data dan cari data member. menampilkannya kepada member		
Aktor	*Admin*		
***Use Case* yang Berkaitan**	*Login*		
Skenario Utama			
Kondisi Awal	Menampilkan menu awal (beranda)		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu Member	2	Menampilkan Halaman member
		3	Menampilkan form tambah member
Skenario Alternatif (Hapus Data member)			
5	Memilih data user yang akan dihapus dan menekan tombol Hapus	6	Menampilkan pesan konfirmasi apakah data akan dihapus
7	Menekan tombol ya pada pesan konfirmasi	8	Menghapus data user pada database

9	Menekan tombol tidak pada pesan konfirmasi	10	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman member
Kondisi Akhir	Data member berhasil diolah		

11. Skenario *Use Case* Pengolahan Kategori

Interaksi antara aktor admin dan operator dengan *use case* pengolahan data kategori dapat dilihat pada tabel 3.14

Tabel 3.14 Skenario *Use Case* Pengolahan Data Kategori

Identifikasi			
Nama	Pengolahan Data kategori		
Tujuan	Mengelola data kategori pada sistem		
Deskripsi	Proses pengolahan data kategori merupakan proses untuk mengatur dan mengolah data kategori pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data		
Aktor	*Admin*		
***Use Case* yang Berkaitan**	*Login*		
Skenario Utama			
Kondisi Awal	Menampilkan data katagori dan menu pengolahan kategori		
Aksi		**Reaksi Sistem**	
1	Memilih menu kategori	2	Menampilkan Halaman kategori.
3	Mengisi form tambah data kategori dan menekan tombol simpan	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke halaman kategori, jika tidak valid maka akan

			menampilkan pesan kesalahan dan menampilkan kembali form tambah data kategori
Skenario Alternatif (Ubah Data Kategori)			
5	Memilih menu ubah data pada data	6	Menampilkan form ubah data yang berisi data dari database yang akan diubah.
7	Mengubah data yang ada pada form ubah data dan menekan tombol simpan	8	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke halaman kategori, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data kategori
Skenario Alternatif (Hapus Data Kategori)			
9	Memilih data kategori yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data kategori pada database
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman kategori
Kondisi Akhir	Data kategori telah berhasil diolah		

12. *Use case* Pembuatan Laporan

Interaksi antara aktor admin dan operator dengan *use case* pembuatan laporan dapat dilihat pada tabel 3.15

Tabel 3.15 Skenario *Use Case* Pembuatan Laporan

Identifikasi			
Nama	Pembuatan laporan		
Tujuan	Membuat laporan dari data yang telah diolah		
Deskripsi	Merupakan proses pembuatan laporan dari data yang telah diolah sebelumnya mencakup barang, order, dan pembayaran.		
Aktor	*Admin*		
***Use Case* yang Berkaitan**	*Login*		
Skenario Utama			
Kondisi Awal	Menampilkan menu pilihan laporan		
Aksi Aktor		**Reaksi**	
1	Memilih data yang akan dijadikan Sistem laporan	2	Mengecek data pada database, jika data ditemukan maka akan menampilkan data, jika tidak ditemukan maka akan menampilkan pesan data tidak ditemukan
3	Memilih jangka waktu yang akan dibuat laporan dan menekan tombol cetak laporan	4	Membuat laporan berdasarkan jangka waktu yang dipilih
Kondisi Akhir	laporan berhasil dibuat		

13. Skenario Use Case Backup dan Restore

Interaksi antara aktor admin dan operator dengan *use case Backup* dan *Restore Database* dapat dilihat pada tabel 3.16 .

Tabel 3.16 Skenario Use Case Backup dan Restore Database

Identifikasi			
Nama	Backup dan Restore Database		
Tujuan	Melakukan proses backup dan restore pada database		
Deskripsi	Proses mengambil semua data pada database (backup) dan mengembalikan data pada database (restore)		
Aktor	*Admin*		
***Use Case* yang Berkaitan**	*Login*		
Skenario Utama			
Kondisi Awal	Menampilkan menu backup dan restore database		
Aksi		**Reaksi Sistem**	
1	Menekan tombol Backup dan Restore database	2	Menampilkan Halaman backup dan restore database
3	Memilih data tabel yang akan di backup	4	Menampilkan data data yang telah pilih untuk di backup
5	Menekan tombol backup untuk mulai membackup	6	Menyimpan file SQL ke tempat penyimpanan file atau hard drive
Skenario Alternatif (Restore Database)			
7	Memilih tombol restore database	8	Menampilkan form restore database
9	Memilih table yang akan di backup		
10	Menekan tombol restore	11	Membaca data dari file SQL
Kondisi Akhir	Data telah di backup atau di restore		

14. Skenario *Use Case* Pengolahan Data Konfirmasi

Interaksi antara aktor member dengan *use case* Konfirmasi pembayaran dapat dilihat pada tabel 3.17.

Tabel 3.17 Skenario *Use Case* Pengolahan Data Konfirmasi

Identifikasi			
Nama	Retur		
Tujuan	Melakukan retur barang		
Deskripsi	Proses meretur barang yang dilakukan oleh member		
Aktor	*Member*		
***Use Case* yang Berkaitan**	*Login*		
Skenario Utama			
Kondisi Awal	Menampilkan Halaman retur barang		
Aksi		**Reaksi Sistem**	
1	Membuka menu detail pemesanan	2	Mengecek pada database, jik ada maka data detail pemesanan akan ditampilkan
3	Memilih menu retur barang	3	Memvalidasi data masukkan jika data valid maka proses retur akan berhasil dilakukan
Kondisi Akhir	Admin berhasil melakukan pengolahan data kretur barang		

15. Skenario *Use Case* Pengolahan Data Barang

Interaksi antara aktor admin dan operator dengan *use case* pengolahan data barang dapat dilihat pada tabel 3.18.

Tabel 3.18 Skenario *Use Case* Pengolahan Data Barang

Identifikasi			
Nama	Pengolahan Data Barang		
Tujuan	Mengelola data barang pada sistem		
Deskripsi	Proses pengolahan data barang merupakan proses untuk mengatur dan mengolah data barang pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data dan cari data.		
Aktor	*Admin*		
***Use Case* yang Berkaitan**	*Login*		
Skenario Utama			
Kondisi Awal	Menampilkan data barang dan menu pengolahan barang		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu tambah data barang	2	Menampilkan form tambah data barang
3	Mengisi form tambah data barang dan menekan tombol simpan	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke halaman barang, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data barang
Skenario Alternatif (Ubah Data Barang)			
5	milih menu ubah data pada data barang yang diinginkan	6	Menampilkan form ubah data yang berisi data dari database yang akan diubah.

7	Mengubah data yang ada pada form ubah data dan menekan tombol simpan	8	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke halaman barang, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data barang
Skenario Alternatif (Hapus Data Barang)			
9	Memilih data barang yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data barang pada database
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman barang
15	Memasukan data cari dan memilih tombol cari	16	Mencari data pada database, jika data ditemukan maka data akan ditampilkan, jika tidak ditemukan maka akan menampilkan pesan data tidak ditemukan.
Kondisi Akhir	Data barang telah berhasil diolah		

16. Skenario *Use Case* Pengolahan Data Pemesanan

Interaksi antara aktor admin dengan *use case* pengolahan data *Pemesanan* dapat dilihat pada tabel 3.19..

Tabel 3.19 *Use Case* Skenario Pengolahan Data Pemesanan

Identifikasi	
Nama	Pengolahan Data Pemesanan
Tujuan	Mengelola data Pemesanan pada sistem

Deskripsi	Proses pengolahan data order merupakan proses untuk mengatur dan mengolah data order pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data dan cari data order.		
Aktor	*Admin*		
***Use Case* yang Berkaitan**	*Pemesanan barang, login*		
Skenario Utama			
Kondisi Awal	Menampilkan data pemesanan dan menu pengolahan data pemesanan		
Aksi		**Reaksi Sistem**	
1	Memilih menu Pemesanan	2	Mencari data pada database, jika data ditemukan maka data akan ditampilkan, jika tidak ditemukan maka akan menampilkan pesan data tidak ditemukan.
Skenario Alternatif (Ubah Data Order)			
3	Memilih menu konfirmasi pembayaran	4	Menampilkan form ubah data yang berisi data pemesanan dari database yang akan diubah.
5	Mengubah data yang ada pada form ubah data dan menekan tombol ceklist	6	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke halaman order, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data order
Skenario Alternatif (Hapus Data Order)			
7	Memilih data order yang akan dihapus	8	Menampilkan pesan konfirmasi apakah data akan dihapus

9	Menekan tombol ya pada pesan konfirmasi	10	Menghapus data order pada database
11	Menekan tombol tidak pada pesan konfirmasi	12	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman order
Kondisi Akhir	Data Pemesanan telah berhasil diolah		

17. Skenario *Use Case* Retur Barang

Interaksi antara aktor member dengan *use case* pengembalian barang dapat dilihat pada tabel 3.20

Tabel 3.20 Skenario *Use* Case Retur Barang

Identifikasi		
Nama	Pengembalian Barang	
Tujuan	Melakukan proses pengembalian barang	
Deskripsi	Merupakan proses pengembalian barang yang tidak sesuai dengan proses pembelian	
Aktor	*Member*	
***Use Case* yang Berkaitan**	*Pemesanan barang, pengolahan data pemesanan*	
Skenario Utama		
Kondisi Awal	Menampilkan menu Pengisian data order barang	
Aksi		**Reaksi Sistem**

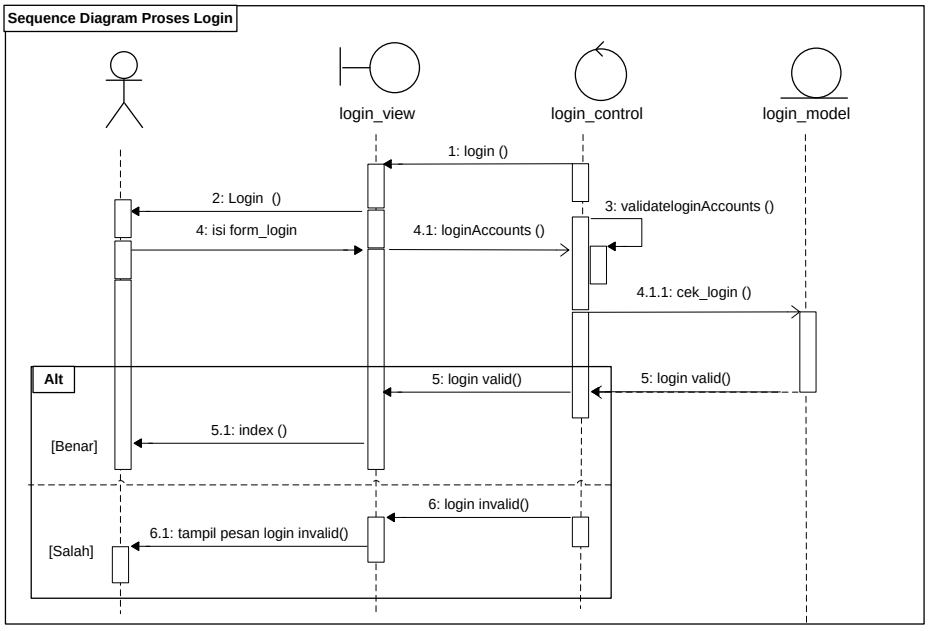
1	Memasukan nomor resi pengiriman	2	Mencari data pada database, jika data ditemukan maka data akan ditampilkan, jika tidak ditemukan maka akan menampilkan pesan data tidak ditemukan.
Kondisi Akhir	Data pengembalian barang telah berhasil dibuat		

3.7.2 *Sequence* Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa pesan (*message*) yang digambarkan terhadap waktu.

1. Sequence Diagram Login Admin

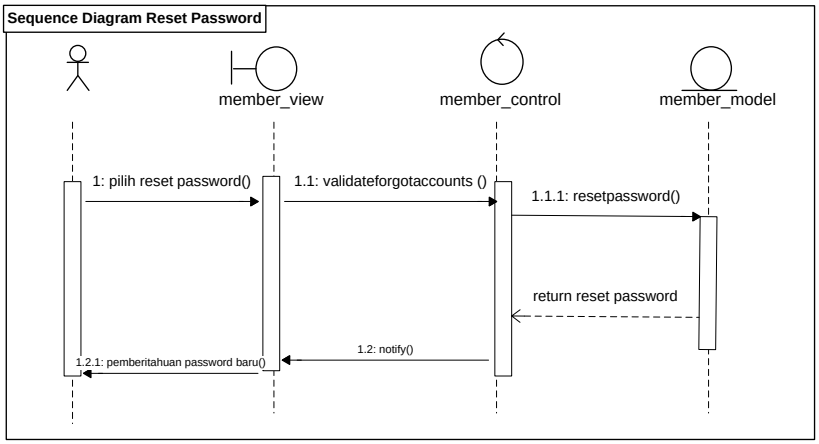
Sequence Diagram Login Admin dapat dilihat pada gambar 3.5



Gambar 3.5 *Sequence Diagram* Login Admin

2. *Sequence Diagram* Reset Password

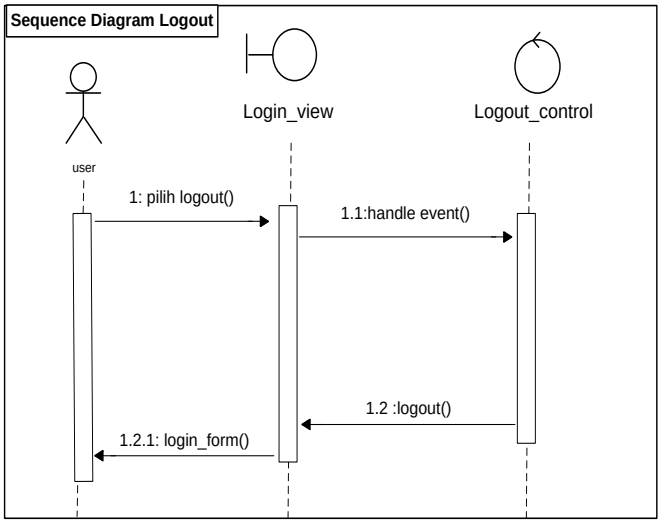
Sequence diagram Permintaan Reset Password dapat dilihat pada gambar 3.6



Gambar 3.6 *Sequence Diagram* Permintaan Reset Password

3. Sequence Diagram Logout

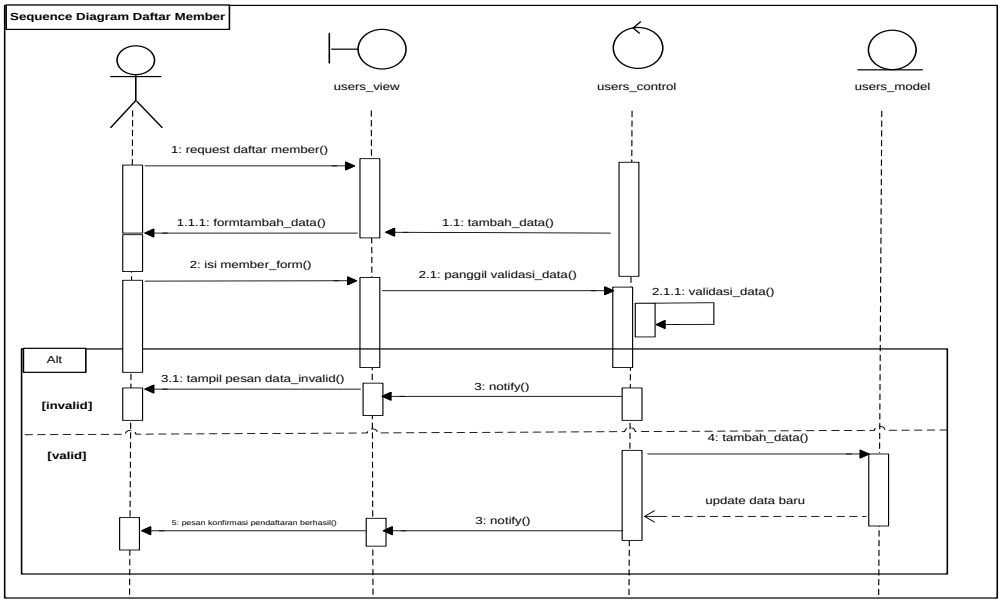
Sequence diagram Logout dapat dilihat pada gambar 3.7



Gambar 3.7 Sequence Diagram Logout

4. Sequence Diagram Daftar Member

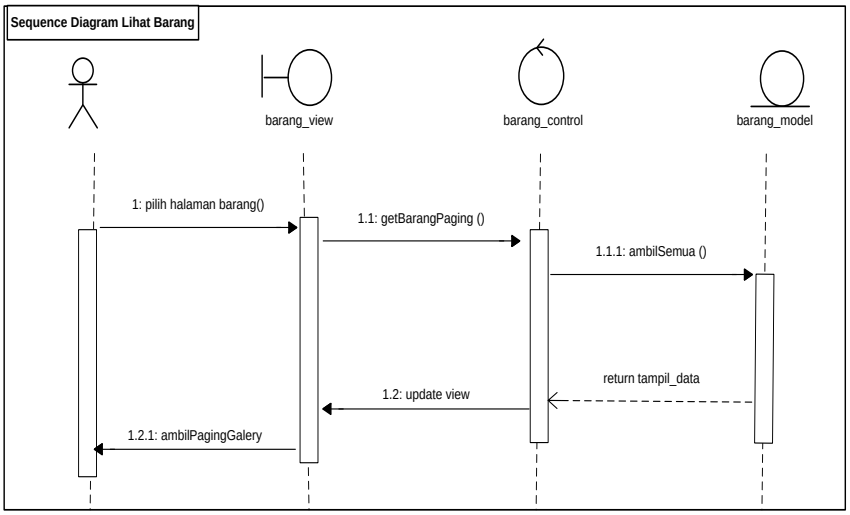
Sequence diagram daftar member dapat dilihat pada gambar 3.8



Gambar 3.8 *Sequence Diagram* Daftar Member

5. Sequence Diagram Lihat Barang

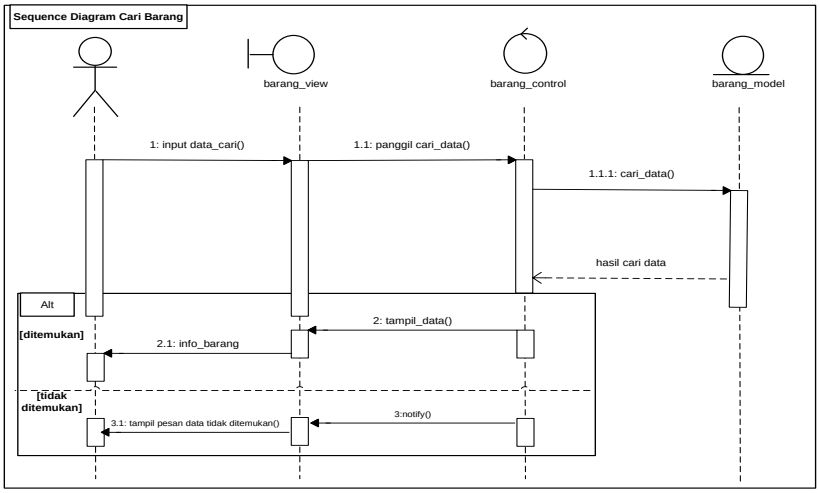
Sequence diagram lihat barang dapat dilihat pada gambar 3.9



Gambar 3.9 *Sequence Diagram* Lihat Barang

6. *Sequence Diagram* Pencarian Barang

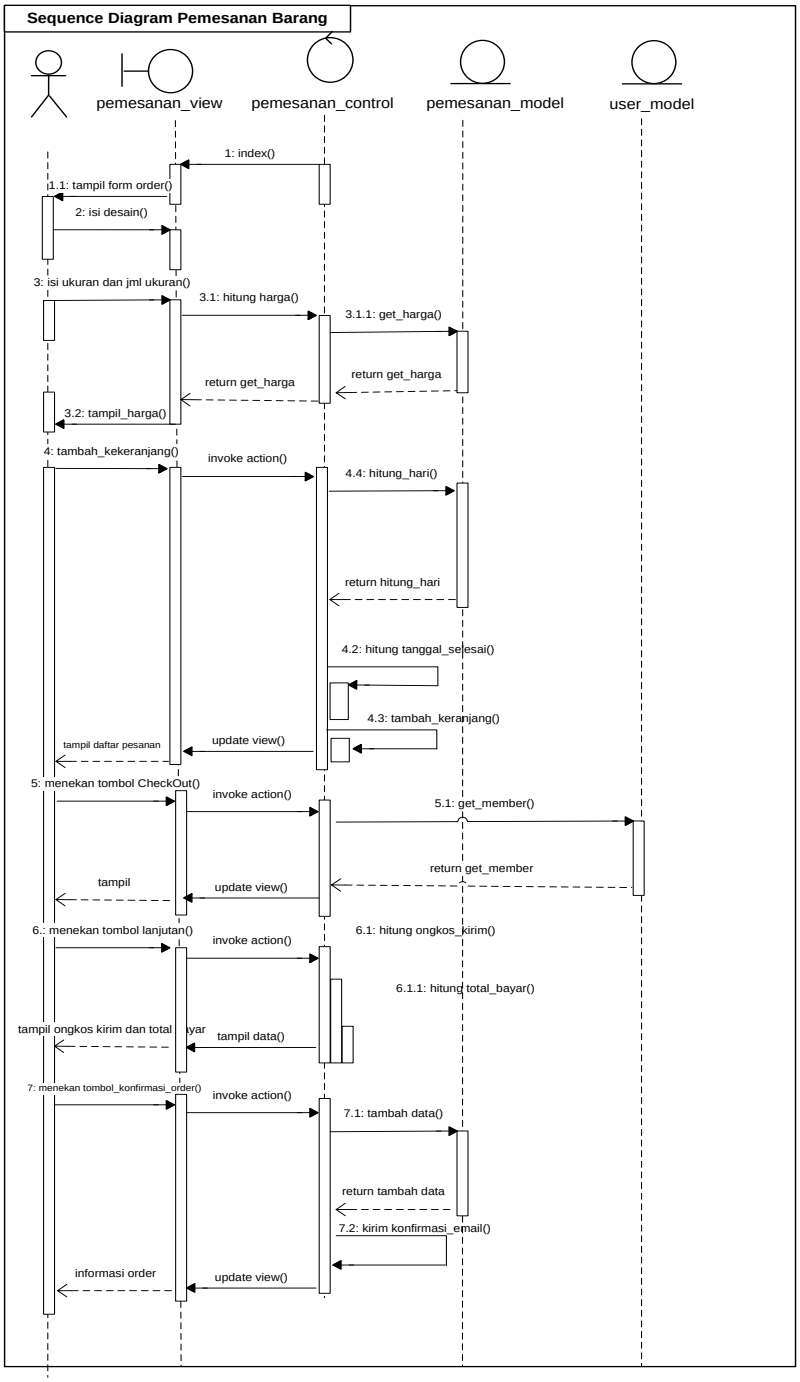
Sequence diagram pencarian barang dapat dilihat pada gambar 3.10



Gambar 3.10 *Sequence Diagram* Pencarian Barang

7. *Sequence Diagram* Pemesanan Barang

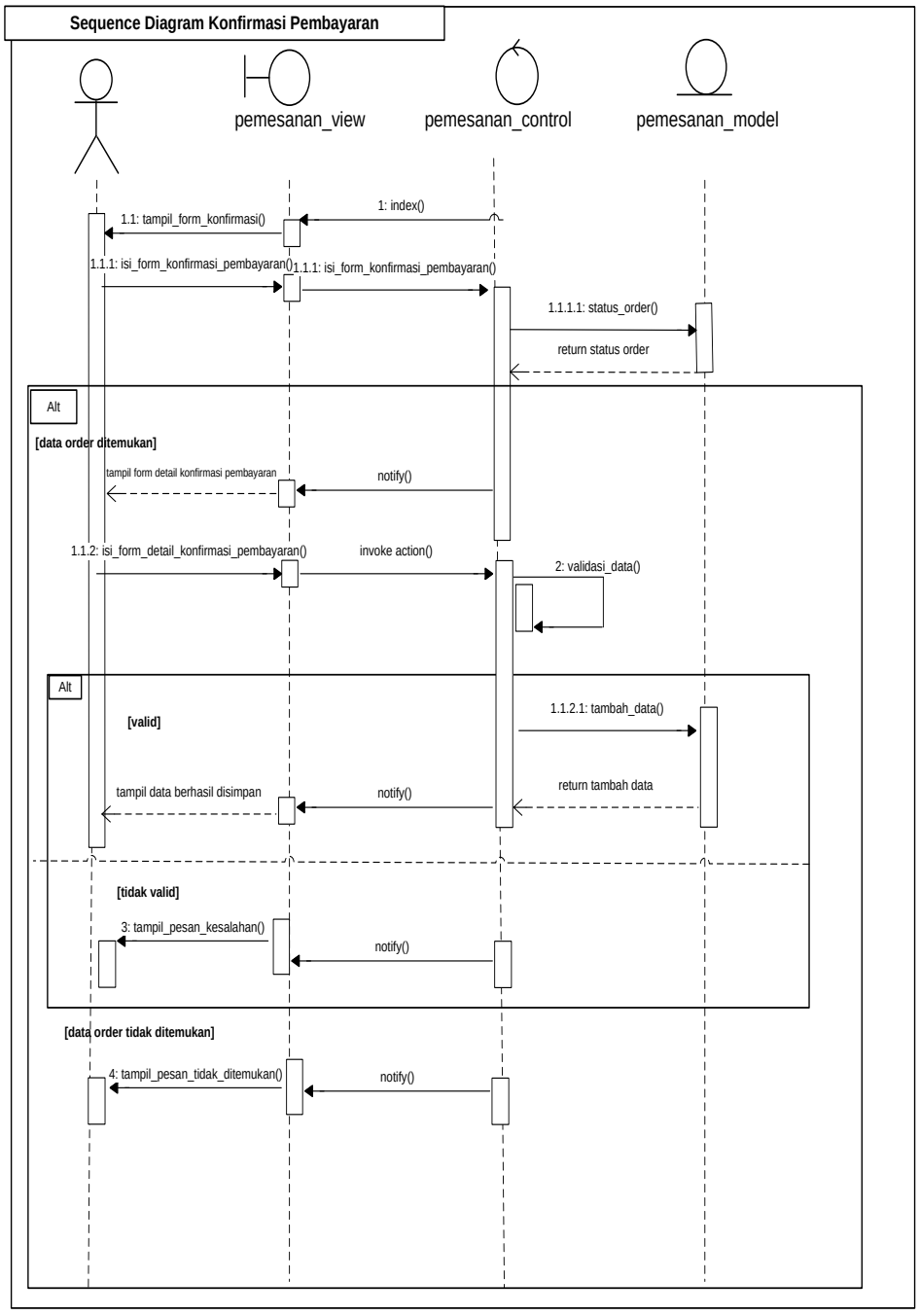
Sequence diagram Pemesanan barang dapat dilihat pada gambar 3.11



Gambar 3.11 *Sequence Diagram* Pemesanan Barang

8. Sequence Diagram Konfirmasi Pembayaran

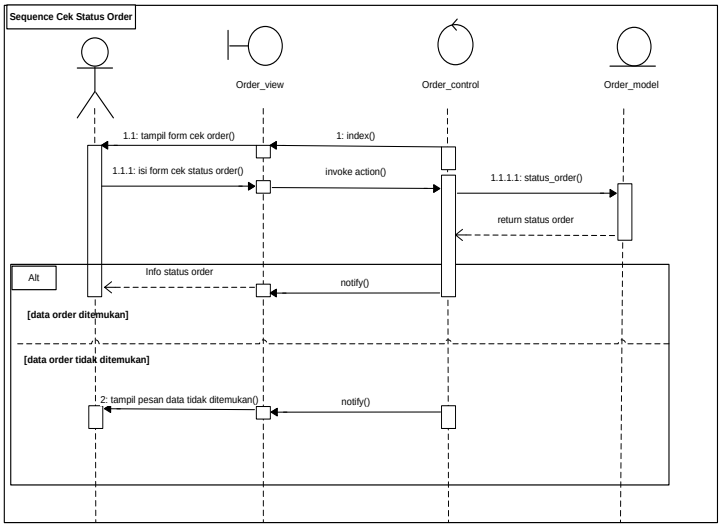
Sequence diagram konfirmasi pembayaran dapat dilihat pada gambar 3.12



Gambar 3.12 *Sequence Diagram* Konfirmasi Pembayaran

9. *Sequence Diagram* Cek Status Order

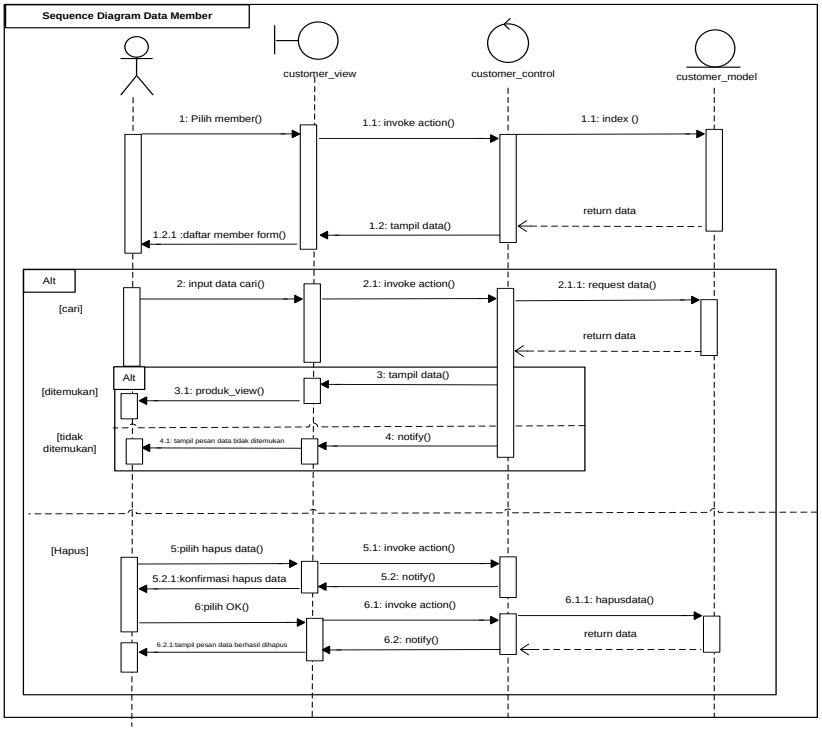
Sequence diagram cek status order dapat dilihat pada gambar 3.13



Gambar 3.13 *Sequence Diagram* Cek Status Order

10. *Sequence Diagram* Pengolahan Data Member

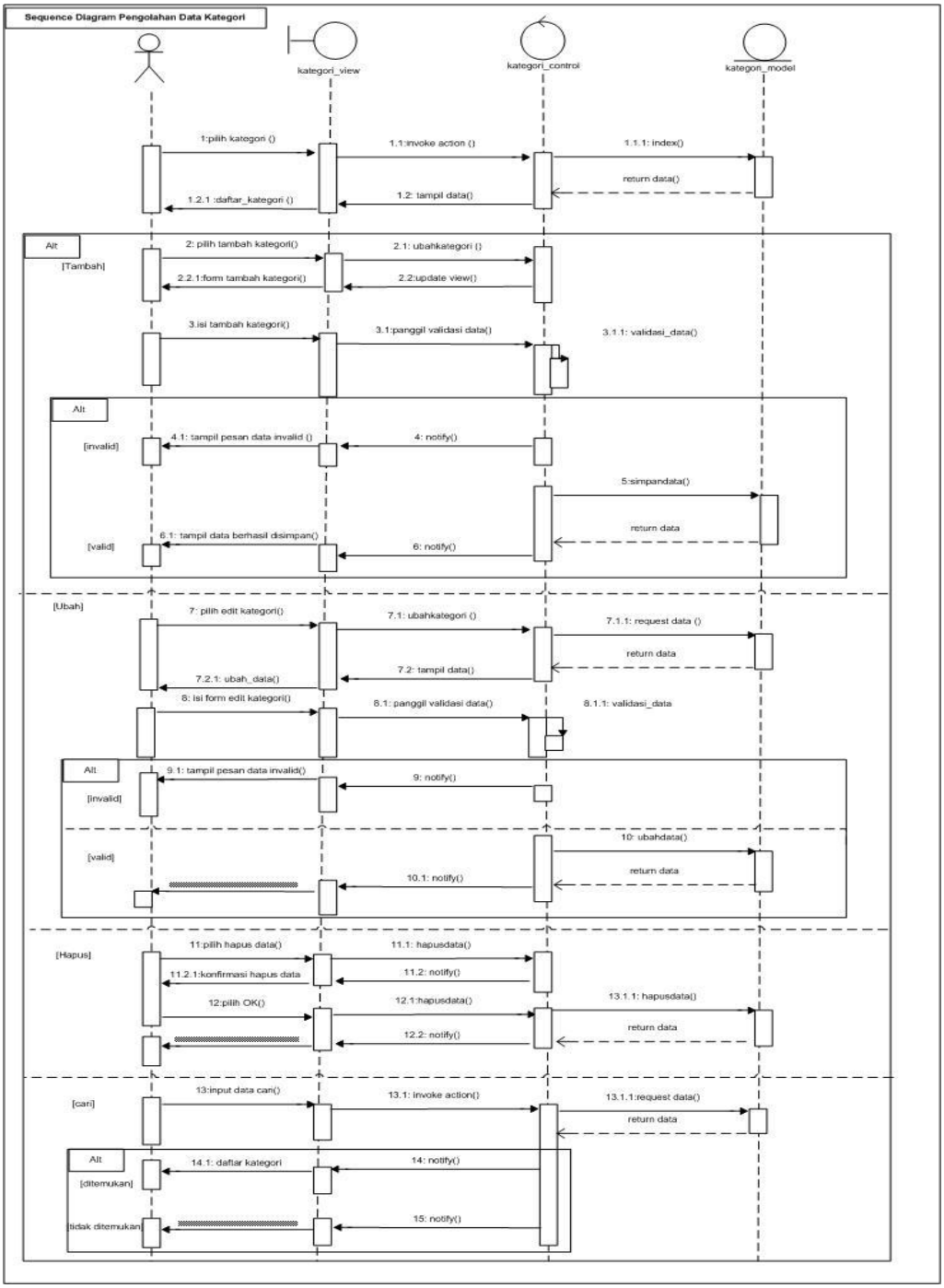
Sequence diagram pengolahan data member dapat dilihat pada gambar 3.14



Gambar 3.14 *Sequence Diagram* Pengolahan Data member

11. *Sequence Diagram* Pengolahan Data Kategori

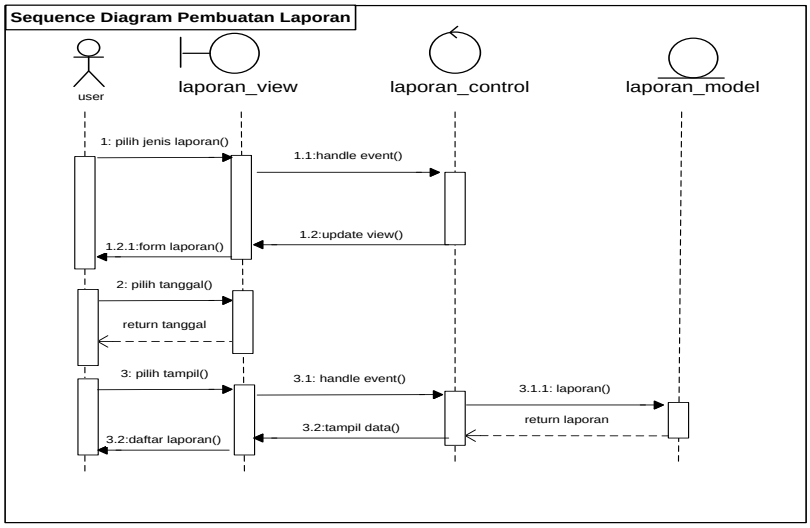
Sequence diagram pengolahan data kategori dapat dilihat pada gambar 3.15



Gambar 3.15 *Sequence Diagram* Pengolahan Data Kategori

12. *Sequence Diagram* Pembuatan laporan

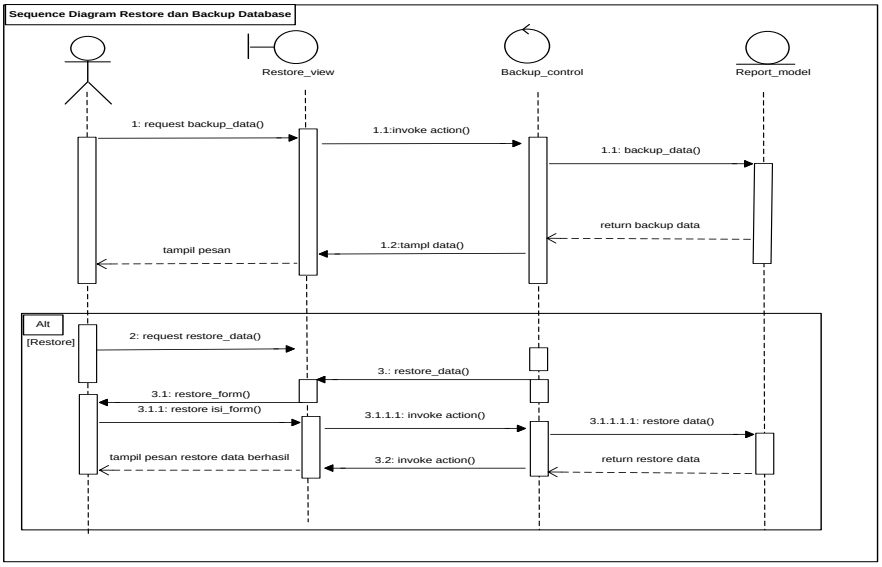
Sequence diagram pembuatan laporan dapat dilihat pada gambar 3.16



Gambar 3.16 *Sequence Diagram* Pembuatan Laporan

13. Sequence Diagram Backup dan Restore Database

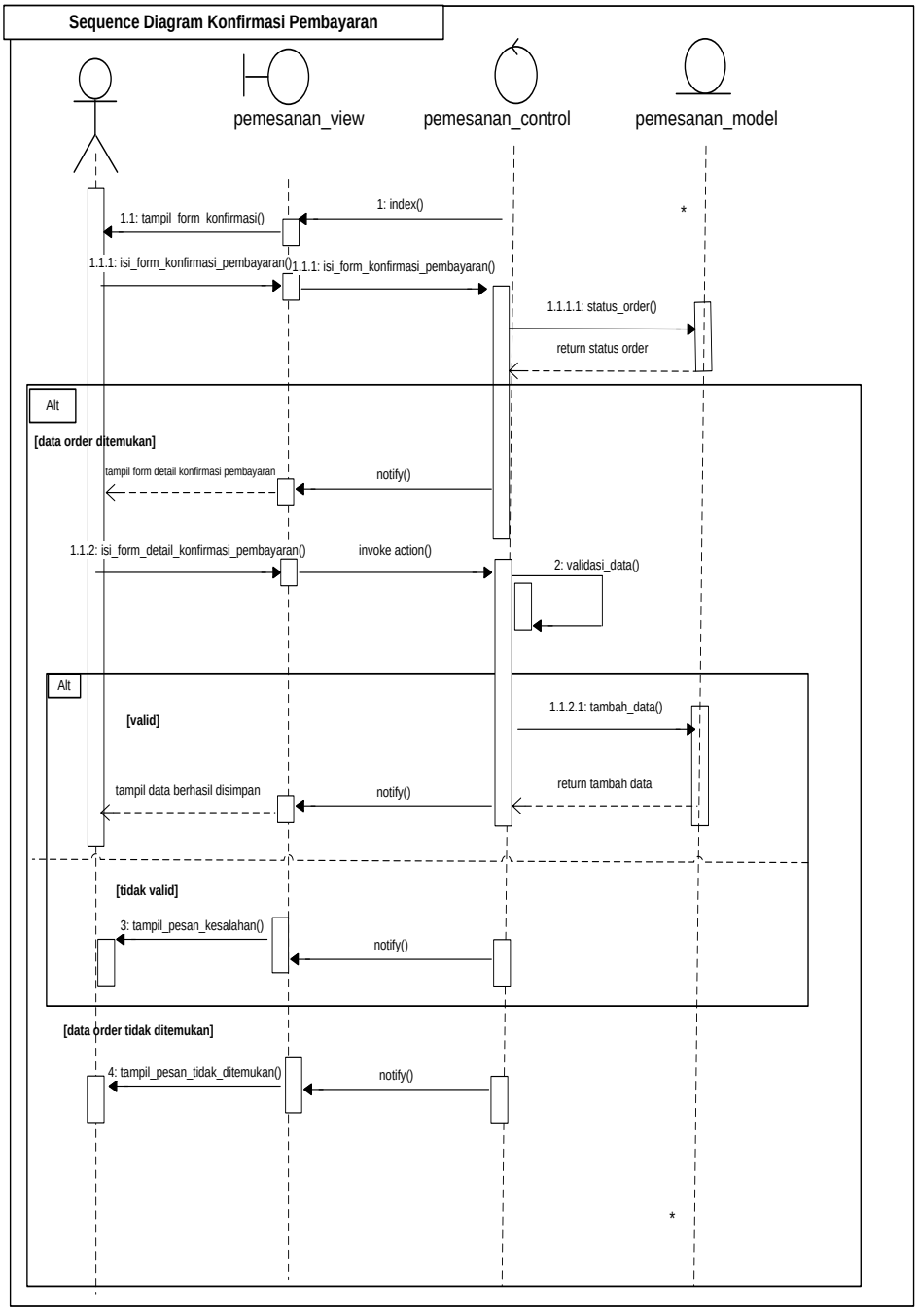
Sequence diagram backup dan restore database dapat dilihat pada gambar 3.17



Gambar 3.17 *Sequence Diagram* Backup dan Restore *Database*

14. *Sequence Diagram* Konfirmasi Pembayaran

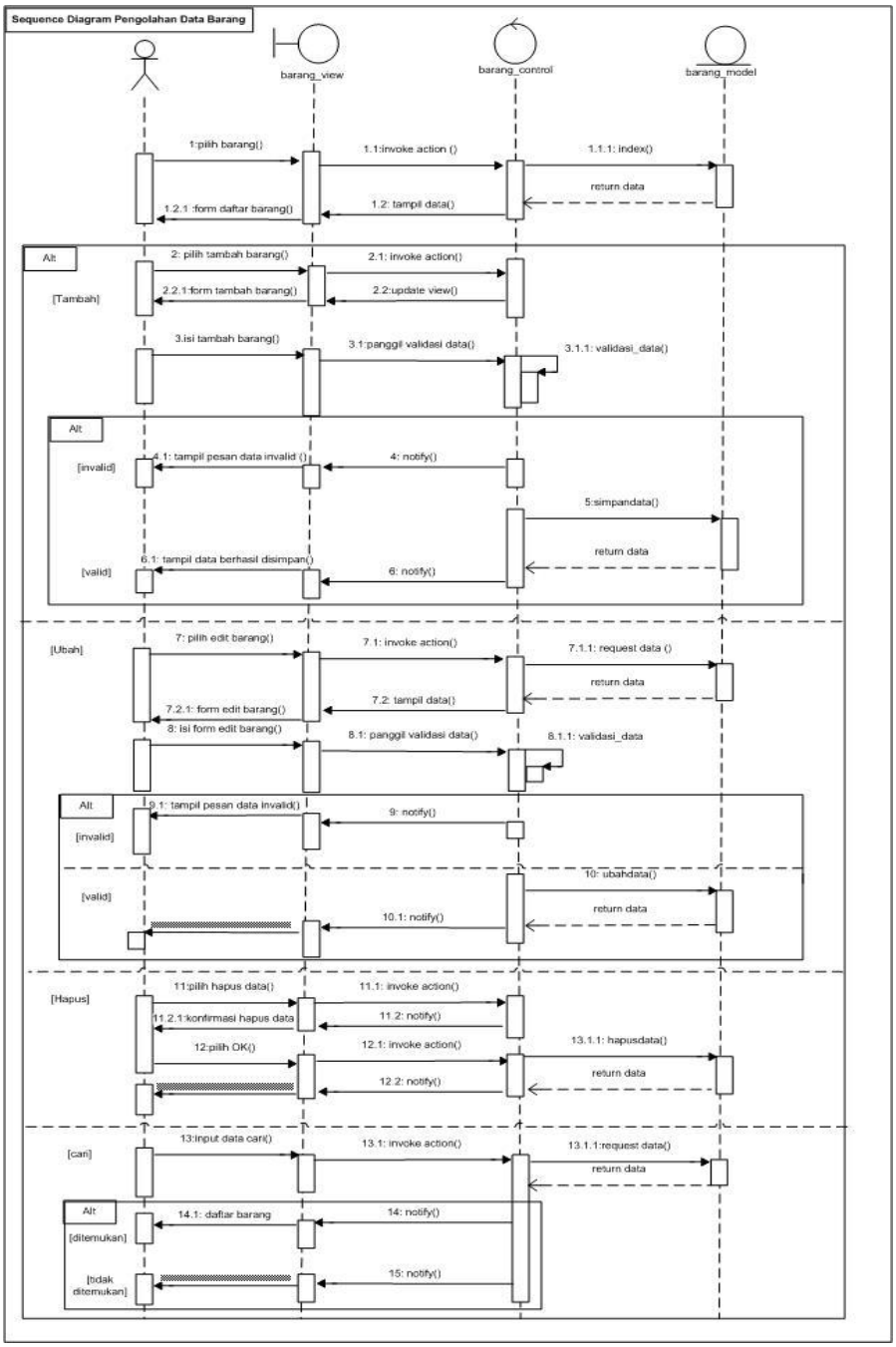
Sequence diagram konfirmasi pembayaran dapat dilihat pada gambar 3.18



Gambar 3.18 *Sequence Diagram* Konfirmasi Pembayaran

15. *Sequence Diagram* Pengolahan Data Barang

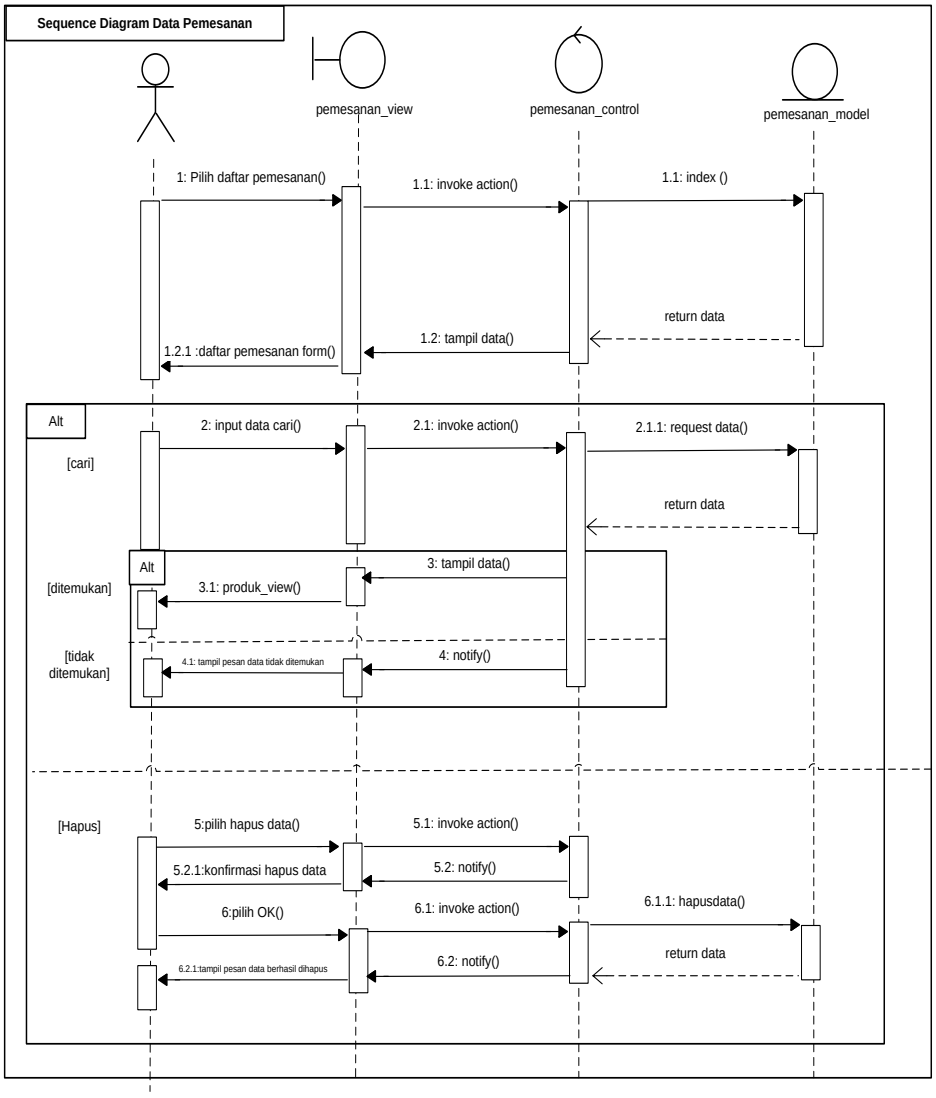
Sequence diagram pengolahan data barang dapat dilihat pada gambar 3.19



Gambar 3.19 *Sequence diagram* pengolahan data barang

16. *Sequence Diagram* Pengolahan Data Pemesanan

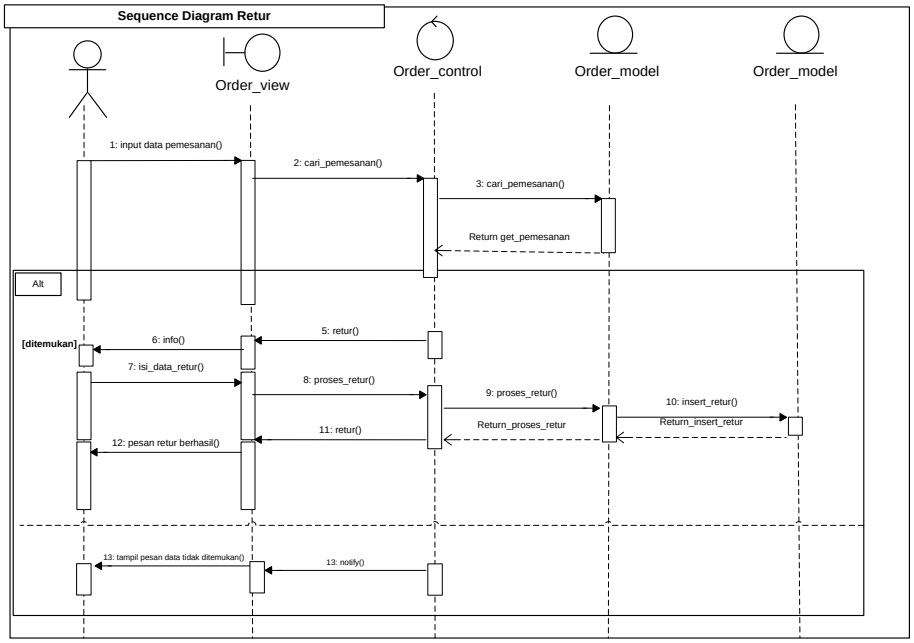
Sequence diagram pengolahan data pemesanan dapat dilihat pada gambar 3.20



Gambar 3.20 *Sequence Diagram* Pengolahan Data Pemesanan

17. Sequence Diagram Retur Barang

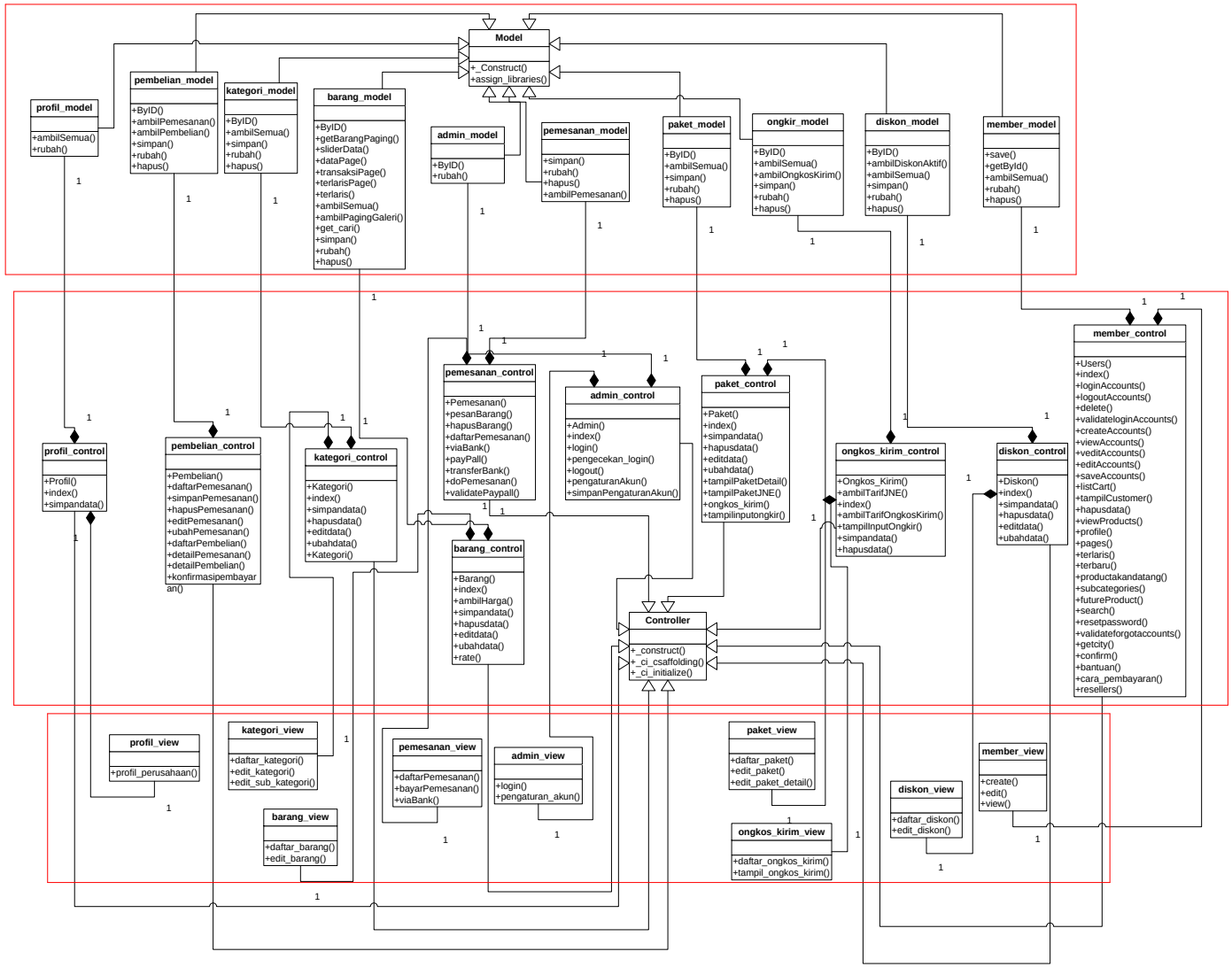
Sequence diagram Pengembalian Barang dapat dilihat pada gambar 3.21



Gambar 3.21 *Sequence Diagram* Retur

3.7.3 *Class* Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut class diagram E-commerce Mitra Teknik pada gambar 3.22.



Gambar 3.22 Class diagram E-commerce Mitra Teknik

Setiap kelas pada gambar 3.22 memiliki jenis kelasnya masing-masing seperti yang terlihat pada tabel 3.22.

Tabel 3.21 Deskrpsi *Class*

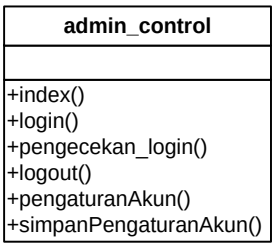
No	Nama	Jenis
1	admin_control	Control
2	barang_control	Control
3	diskon_control	Control
4	kategori_control	Control
5	pemesanan_control	Control
6	profil_control	Control
7	member_control	Control
8	Laporan_control	Control
9	Ongkos_kirim_control	Control
10	pembelian_control	Control
11	barang_model	Model
12	diskon_model	Model
13	kategori_model	Model
14	pemesanan_model	Model
15	profil_model	Model
16	admin_model	Model
17	member_model	Model
18	Laporan_model	Model
19	Ongkos_kirim_model	Model
20	pembelian_model	Model
21	admin_view	View
22	barang_view	View
23	customer_view	View
24	diskon_view	View
25	kategori_view	View

26	profil_view	View
27	users_view	View
28	pemesanan_view	View
29	Ongkos_kirim_view	View
30	pembelian_view	View

Deskripsi dari setiap *class* yang ada dijelaskan sebagai berikut :

1. *Class* admin_control

Class admin_control merupakan *class* yang digunakan untuk mengatur proses pengolahan pada data admin dan bertugas untuk memanggil *class-class* yang lainnya. Atribut dan operasi yang dimiliki *class* admin_control dapat dilihat pada gambar 3.23



Gambar 3.23 Deskripsi *Class* admin_control

a. Deskripsi Atribut

Class category_control tidak memiliki atribut.

b. Deskripsi Operasi

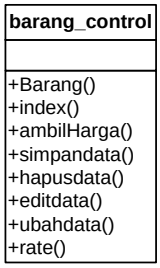
Deskripsi operasi *class* category_control dapat dilihat pada tabel 3.22

Tabel 3.22 Deskripsi Operasi *Class* category_control

Nama	Keterangan
index() :	Merupakan Prosedur yang dijalankan ketika class pertama kali dipanggil, prosedur ini akan memanggil semua *class* model yang berhubungan dengan admin dan menampilkannya di halaman utama admin
login () : void	Prosedur yang dipanggil ketika user melakukan proses login, memvalidasi dan memanggil class Login_Model untuk menampilkan menu login admin
pengecekan_login () : void	Prosedur untuk memvalidasi data login yang dimasukan admin apabila benar maka sistem akan menampilan menu utama admin, apabila salah akan kembai ke menu login
logout () : void	Prosedur ketika memanggil logout_control, digunakan untuk keluar dari sistem dan kembali menampilkan form login
pengaturanAkun () : void	Prosedur yang dipanggil ketika proses edit data admin
simpanPengaturanAkun () : void	Prosedur yang dipanggil ketika proses simpan data admin yang telah diubah pada database.

2. *Class* barang_control

Class barang_control merupakan *class* yang digunakan untuk mengatur proses pengolahan pada data barang dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* barang_control dapat dilihat pada gambar 3.24



Gambar 3.24 Deskripsi *Class* barang_control

a. Deskripsi Atribut

Class barang_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* barang_control dapat dilihat pada tabel 3.23

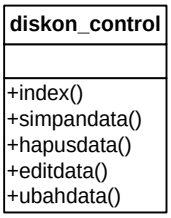
Tabel 3.23 Deskripsi Operasi barang_control

Nama	Keterangan
index() :	Merupakan Prosedur yang dijalankan ketika class pertama kali dipanggil, prosedur ini akan menampilkan menu data barang
ambilHarga () : void	Prosedur yang dipanggil untuk mengambil data dari *class* barang_model dan menampilkan data barang dari *database*
simpandata () : void	Prosedur yang dipanggil untuk menyimpan data barang yang dimasukan pada form tambah barang.
hapusdata () : void	Prosedur untuk menghapus data barang, akan menampilkan pesan konfirmasi apakah data barang akan dihapus, jika ya maka akan memanggil barang_model untuk melakukan penghapusan data pada database.
editdata () : void	Prosedur yang digunakan untuk memanggil class barang_view untuk menampilkan form edit barang.
ubahdata () : void	Prosedur untuk melakukan penyimpanan perubahan data barang yang telah di edit ke database

Rate () : void	Prosedur untuk melakukan penyimpanan rating barang yang telah dipilih member

3. *Class* diskon_control

Class diskon_control merupakan *class* yang digunakan untuk mengatur proses pengolahan diskon dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* diskon_control dapat dilihat pada gambar 3.25



Gambar 3.25 Deskripsi *Class* diskon_control

a. Deskripsi Atribut

Class diskon_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* diskon_control dapat dilihat pada tabel 3.24

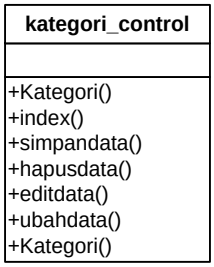
Tabel 3.24 Deskripsi Operasi diskon_control

Nama	Keterangan
index() :	Merupakan Prosedur yang dijalankan ketika class pertama kali dipanggil, prosedur ini akan menampilkan menu daftar diskon
simpandata () : void	Prosedur yang dipanggil untuk menyimpan data diskon yang dimasukan pada form tambah diskon

hapusdata () : void	Prosedur untuk menghapus data diskon, akan menampilkan pesan konfirmasi apakah data diskon akan dihapus, jika ya maka akan memanggil diskon_model untuk melakukan penghapusan data pada database.
editdata () : void	Prosedur yang digunakan untuk memanggil class diskon_view untuk menampilkan form edit diskon
ubahdata () : void	Prosedur untuk melakukan penyimpanan perubahan data diskon yang telah di edit ke database

4. *Class* kategori_control

Class kategori_control merupakan *class* yang digunakan untuk mengatur proses pengolahan kategori dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* kategori_control dapat dilihat pada gambar 3.26



Gambar 3.26 Deskripsi *Class* kategori_control

a. Deskripsi Atribut

Class kategori_control tidak memiliki atribut.

b. Deskripsi Operasi

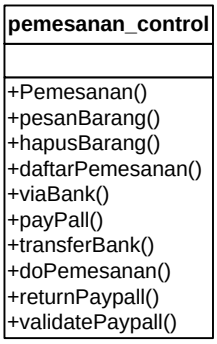
Deskripsi operasi *class* kategori_control dapat dilihat pada tabel 3.25

Tabel 3.25 Deskripsi Operasi kategori_control

Nama	Keterangan
index() :	Merupakan Prosedur yang dijalankan ketika class pertama kali dipanggil, prosedur ini akan menampilkan menu daftar kategori
simpandata () : void	Prosedur yang dipanggil untuk menyimpan data kategori yang dimasukan pada form tambah kategori
hapusdata () : void	Prosedur untuk menghapus data kategori, akan menampilkan pesan konfirmasi apakah data kategori akan dihapus, jika ya maka akan memanggil kategori_model untuk melakukan penghapusan data pada database.
editdata () : void	Prosedur yang digunakan untuk memanggil class kategori_view untuk menampilkan form edit kategori
ubahdata () : void	Prosedur untuk melakukan penyimpanan perubahan data kategori yang telah di edit ke database
Kategori () : void	Prosedur yang digunakan untuk menampilkan kategori ke menu kategori

5. *Class* pemesanan_control

Class pemesanan_control merupakan *class* yang digunakan untuk mengatur proses pengolahan pemesanan atau transaksi dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* pemesanan_control dapat dilihat pada gambar 3.27



Gambar 3.27 Deskripsi *Class* pemesanan_control

a. Deskripsi Atribut

Class pemesanan_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* pemesanan_control dapat dilihat pada tabel 2.28

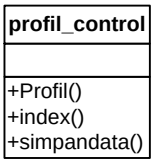
Tabel 3.26 Deskripsi Operasi pemesanan**_control**

Nama	Keterangan
Pemesanan	Prosedur yang berfungsi mengkonversi nilai mata uang dari total pembelian customer ke dollar
pesanBarang () : void	Prosedur yang berfugsi untuk memanggil session form pemesanan barang
hapusBarang () : void	Prosedur yang berfungsi untuk menghapus barang yang telah dipesan di daftar pemesanan
daftarPemesanan () : void	Prosedur yang berfungsi menampilkan daftar pemesanan
Bank () : void	Prosedur yang berfungsi memanggil menu pembayaran via bank
validatePaypall () : void	Prosedur untuk mengirimkan notifikasi pembayaran via paypall yang dilakukan member ke menu admin

paypall () : void	Prosedur untuk link ke situs paypall
transferBank () : void	Prosedur yang berfungsi untuk menyimpan data transfer bank dan menampilkannya di notifikasi admin
doPemesanan () : void	Prosedur untuk menyimpan data pemesanan dan menampilkan data pemesanan di notifikasi admin

6. *Class* profil_control

Class profil_control merupakan *class* yang digunakan untuk mengatur proses pengolahan profil perusahaan dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* profil_control dapat dilihat pada gambar 3.28.



Gambar 3.28 Deskripsi *Class* profil_control

a. Deskripsi Atribut

Class profil_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* profil_control dapat dilihat pada tabel 3.29

Tabel 3.27 Deskripsi Operasi profil_control

Nama	Keterangan
Index () :	Merupakan prosedur yang dijalankan di menu profil
simpandata () : void	Prosedur yang dipanggil untuk menyimpan data profil perusahaan ke database

Profil () :	Merupakan prosedur yang dijalankan ketika class pertama kali dipanggil, prosedur ini akan menampilkan menu profil

7. member_control

Class users_control merupakan *class* yang digunakan untuk mengatur proses pengolahan data member dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* users_control dapat dilihat pada gambar 3.29



Gambar 3.29 Deskripsi *Class* member_control

a. Deskripsi Atribut

Class users_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* member_control dapat dilihat pada tabel 3.30

Tabel 3.28 Deskripsi Operasi member_control

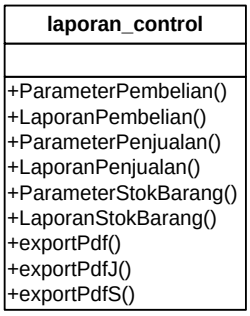
Nama	Keterangan
index() :	Merupakan Prosedur yang dijalankan ketika class pertama kali dipanggil, prosedur ini akan menampilkan menu utama member
loginAccounts () : void	Prosedur yang dipanggil untuk masuk ke menu login
logoutAccounts () : void	Prosedur yang digunakan untuk keluar dari sistem
delete () : void	Prosedur untuk menghapus data daftar belanja member
validateloginAccounts () : void	Prosedur yang dipanggil ketika user melakukan proses login, memvalidasi dan memanggil class Login_Model untuk mencari data user yang ada pada database.
createAccounts () : void	Prosedur yang digunakan untuk membuat data account
viewAccounts () : void	Prosedur yang dipanggil untuk menampilkan data account member
veditAccounts () : void	Prosedur yang digunakan untuk memanggil class account_view untuk menampilkan form edit account
editAccounts () : void	Prosedur untuk melakukan penyimpanan perubahan dataaccount yang telah di edit ke database
saveAccounts () : void	Prosedur yang digunakan untuk menyimpan data pendaftaran account ke database
hapusdata () : void	Prosedur untuk menghapus data member, akan menampilkan pesan konfirmasi apakah data member akan dihapus, jika ya maka akan memanggil customer_model untuk melakukan

	penghapusan data pada database.
viewProducts () : void	Prosedur yang digunakan untuk menampilkan daftar barang di menu member
profile () : void	Prosedur yang digunakan untuk menampilkan data perusahaan di menu member
terlaris () : void	Prosedur yang akan memanggil session data terlaris
terbaru : () void	Prosedur yang akan memanggil session data terbaru
productakandatang () : void	Prosedur yang akan memanggil session product yang akan datang
subcategories () : void	Prosedur untuk menampilkan kategori dan subkategori barang di menu member
search () : void	Prosedur untuk melakukan pencarian data barang ke database kemudian menampilkannya di menu member
Resetpassword () : void	Prosedur untuk memanggil resetpassword
Validateforgotaccounts () : void	Prosedur untuk melakukan resetpassword accounts
Getcity () : void	Prosedur untuk menampilkan data kota
Confirm () : void	Prosedur untuk melakukan konfirmasi
Bantuan () : void	Prosedur untuk menampilkan bantuan
Cara_pembayaran () :	Prosedur untuk menamfilkan form cara pembayaran
Resellers () :	Prosedur untuk memenggil status resellers

8. *Class* laporan_control

Class laporan_control merupakan *class* yang digunakan untuk mengatur proses pengolahan data laporan dan bertugas untuk memanggil *class-class* yang

berkaitan. Atribut dan operasi yang dimiliki *class* laporan_control dapat dilihat pada gambar 3.30



Gambar 3.30 Deskripsi *Class* laporan_control

a. Deskripsi Atribut

Class laporan_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* laporan_control dapat dilihat pada tabel 3.31

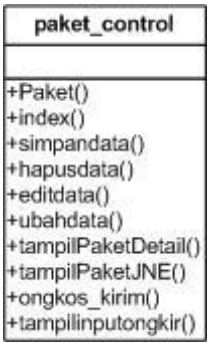
Tabel 3.29 Deskripsi Operasi laporan_control

Nama	Keterangan
ParameterPembelian () : void	Prosedur yang digunakan untuk memanggil parameter pembelian
LaporanPembelian () : void	Prosedur yang digunakan untuk mengambil data pembelian
ParameterPenjualan () : void	Prosedur yang digunakan untuk memanggil parameter pembelian
LaporanPenjualan () : void	Prosedur yang digunakan untuk mengambil data penjualan
ParameterStokBarang () : void	Prosedur yang digunakan untuk memanggil parameter stok barang
LaporanStokBarang () : void	Prosedur yang digunakan untuk mengambil data stok barang
exportPdf () : void	Prosedur yang digunakan untuk mengexport data pembelian dalam format

	pdf
exportPdfJ () : void	Prosedur yang digunakan untuk mengexport data pembelian dalam format pdf
exportPdfS () : void	Prosedur yang digunakan untuk mengexport data pembelian dalam format pdf

9. *Class* ongkos_kirim_control

Class ongkos_kirim_control merupakan *class* yang digunakan untuk mengatur proses pengolahan data ongkos kirim dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* ongkos_kirim_control dapat dilihat pada gambar 3.31



Gambar 3.31 Deskripsi *Class* ongkos_kirim_control

a. Deskripsi Atribut

Class ongkos_kirim_control tidak memiliki atribut.

b. Deskripsi Operasi

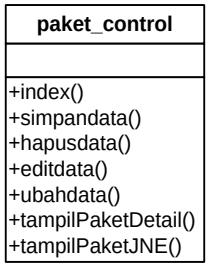
Deskripsi operasi *class* ongkos_kirim_control dapat dilihat pada tabel 3.31

Tabel 3.30 Deskripsi Operasi ongkos_kirim_control

Nama	Keterangan
ambilTarifJNE () : void	Prosedur yang digunakan untuk mengambil data ongkos kirim dari situs jasa pengiriman JNE
index () : void	Merupakan Prosedur yang dijalankan ketika class pertama kali dipanggil, prosedur ini akan menampilkan menu daftar ongkos kirim
ambilTarifOngosKirim () : void	Prosedur yang digunakan untuk mengambil data ongkos kirim dari database
tampilInputOngkir () : void	Prosedur yang digunakan untuk menampilkan menu input ongkos kirim
simpandata () : void	Prosedur yang dipanggil untuk menyimpan data ongkos yang dimasukan pada form input ongkos kirim
hapusdata () : void	Prosedur untuk menghapus data ongkos, akan menampilkan pesan konfirmasi apakah data akan dihapus, jika ya maka akan memanggil ongkir_model untuk melakukan penghapusan data pada database.
Ongkos_kirim () : void	Prosedur untuk menampilkan form ongkos kirim

10. *Class* paket_control

Class paket_control merupakan *class* yang digunakan untuk mengatur proses pengolahan data paket dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* paket_control dapat dilihat pada gambar 3.32



Gambar 3.32 Deskripsi *Class* paket_control

a. Deskripsi Atribut

Class paket_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* paket_control dapat dilihat pada tabel 3.32

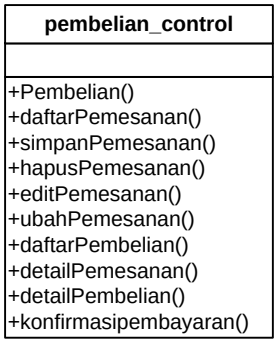
Tabel 3.31 Deskripsi Operasi paket_control

Nama	Keterangan
index() :	Merupakan Prosedur yang dijalankan ketika class pertama kali dipanggil, prosedur ini akan menampilkan menu daftar ongkos kirim
simpandata () : void	Prosedur yang dipanggil untuk menyimpan data supplier yang dimasukan pada form tambah ongkos kirim
hapusdata () : void	Prosedur untuk menghapus data ongkos kirim, akan menampilkan pesan konfirmasi apakah data ongkos kirim akan dihapus, jika ya maka akan memanggil ongkir_model untuk melakukan penghapusan data pada database.
editdata () : void	Prosedur yang digunakan untuk memanggil class ongkos_kirim_view untuk menampilkan form edit data ongkos kirim
ubahdata () : void	Prosedur untuk melakukan penyimpanan perubahan data ongkos kirim yang telah di edit ke database
tampilPaketDetail () : void	Prosedur yang digunakan untuk menampilkan jenis paket pengiriman

tampilPaketJNE () : void	Prosedur yang digunakan untuk mengambil data dari situs jasa pengiriman JNE

11. *Class* pembelian_control

Class pembelian merupakan *class* yang digunakan untuk mengatur proses pengolahan data pembelian dan bertugas untuk memanggil *class-class* yang berkaitan. Atribut dan operasi yang dimiliki *class* pembelian_control dapat dilihat pada gambar 3.33



Gambar 3.33 Deskripsi *Class* pembelian_control

a. Deskripsi Atribut

Class pembelian_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* pembelian_control dapat dilihat pada tabel 3.33

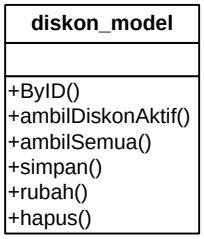
Tabel 3.32 Deskripsi Operasi pembelian_control

Nama	Keterangan
daftarPemesanan () : void	Prosedur yang dipanggil untuk menampilkan daftar barang yang akan dipesan ke supplier
simpan () : void	Prosedur yang dipanggil untuk menyimpan data pembelian

hapus () : void	Prosedur untuk menghapus data pembelian, akan menampilkan pesan konfirmasi apakah data pembelian akan dihapus, jika ya maka akan memanggil pembelian_model untuk melakukan penghapusan data pada database.
edit () : void	Prosedur untuk melakukan penyimpanan perubahan data account yang telah di edit ke database
ubah () : void	Prosedur untuk melakukan penyimpanan perubahan data pembelian yang telah di edit ke database
daftarPembelian () : void	Prosedur untuk menampilkan daftar pembelian
detailPemesanan () : void	Prosedur untuk menampilkan detail pemesanan
detailPembelian () : void Konfirmasipemesanan () : void	Prosedur untuk menampilkan detail pembelian Prosedur untuk melakukan konfirmasi pemesanan

12. *Class* diskon_model

Class diskon_model merupakan *class* yang berisi prosedur-prosedur untuk mengelola diskon. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* diskon_model dapat dilihat pada gambar 3.34



Gambar 3.34 Deskripsi *Class* diskon_model

a. Deskripsi Atribut

Class customer _model tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* customer _model dapat dilihat pada tabel 3.34

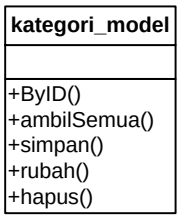
Tabel 3.33 Deskripsi Operasi *Class* diskon_model

Nama	Keterangan
ById () : void	Prosedur untuk mengambil data dari database berdasarkan id diskon.
ambilDiskonAktif () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan diskon yang aktif.
ambilSemua () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data diskon.
simpan () : void	Prosedur yang dipanggil ketika proses menyimpan data diskon pada database.
rubah () : void	Prosedur yang dipanggil ketika proses ubah data diskon, melakukan perintah update data pada database.
hapus () : void	Prosedur yang dipanggil ketika proses menghapus data diskon data pada database.

13. *Class* kategori_model

Class kategori_model merupakan *class* yang berisi prosedur-prosedur untuk mengelola data kategori. *Class* ini berhubungan langsung dengan *database* untuk

proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* kategori_model dapat dilihat pada gambar 3.35



Gambar 3.35 Deskripsi *Class* kategori_model

a. Deskripsi Atribut

Class kategori _model tidak memiliki atribut.

b. Deskripsi Operasi

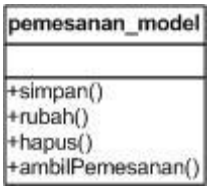
Deskripsi operasi class kategori_model dapat dilihat pada tabel 3.35

Tabel 3.34 Deskripsi Operasi *Class* kategori_model

Nama	Keterangan
ById () : void	Prosedur untuk mengambil data dari database berdasarkan id kategori.
ambilSemua () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data kategori.
simpan () : void	Prosedur yang dipanggil ketika proses menyimpan datakategori pada database.
rubah () : void	Prosedur yang dipanggil ketika proses ubah data kategori, melakukan perintah update data pada database.
hapus () : void	Prosedur yang dipanggil ketika proses menghapus data kategori data pada database.

14. *Class* pemesanan_model

Class pemesanan_model merupakan *class* yang berisi prosedur-prosedur untuk mengelola data transaksi. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* pemesanan_model dapat dilihat pada gambar 3.36



Gambar 3.36 Deskripsi *Class* pemesanan_model

a. Deskripsi Atribut

Class pemesanan_model tidak memiliki atribut.

b. Deskripsi Operasi

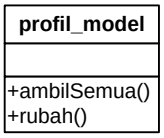
Deskripsi operasi *class* pemesanan_model dapat dilihat pada tabel 3.36

Tabel 3.35 Deskripsi *Class* pemesanan_model

Nama	Keterangan
simpan () : void	Prosedur yang dipanggil ketika proses menyimpan data transaksi pada database.
rubah () : void	Prosedur yang dipanggil ketika proses ubah data transaksi, melakukan perintah update data pada database.
hapus () : void	Prosedur yang dipanggil ketika proses menghapus datatransaksi data pada database.
ambilPemesanan () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data transaksi.

15. *Class* profil_model

Class profil_model merupakan *class* yang berisi prosedur-prosedur untuk mengelola data profil perusahaan. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* profil_model dapat dilihat pada gambar 3.37



Gambar 3.37 Deskripsi *Class* profil_model

a. Deskripsi Atribut

Class profil_model tidak memiliki atribut.

b. Deskripsi Operasi

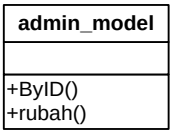
Deskripsi operasi *class* profil_model dapat dilihat pada tabel 3.37

Tabel 3.36 Deskripsi *Class* profil_model

Nama	Keterangan
ambilSemua () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data profil.
rubah () : void	Prosedur yang dipanggil ketika proses ubah data profil, melakukan perintah update data pada database.

16. *Class* admin_model

Class user_model merupakan *class* yang berisi prosedur-prosedur untuk mengelola data admin. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* user_model dapat dilihat pada gambar 3.38



Gambar 3.38 Deskripsi *Class* admin_model

a. Deskripsi Atribut

Class user_model tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* user_model dapat dilihat pada tabel 3.38

Tabel 3.37 *class* user_model

Nama	Keterangan
ById () : void	Prosedur untuk mengambil data dari database berdasarkan id admin.
rubah () : void	Prosedur yang dipanggil ketika proses ubah data admin, melakukan perintah update data pada database.

17. *Class* member_model

Class users_model merupakan *class* yang berisi prosedur-prosedur untuk mengelola data member. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* member_model dapat dilihat pada gambar 3.39



Gambar 3.39 Deskripsi *Class* member_model

a. Deskripsi Atribut

Class users_model tidak memiliki atribut.

b. Deskripsi Operasi

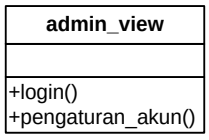
Deskripsi operasi *class* users_model dapat dilihat pada tabel 3.39

Tabel 3.38 *class* member_model

Nama	Keterangan
save () : void	Prosedur yang dipanggil ketika proses menyimpan data customer pada database setelah pembuatan account baru.
getById () : void	Prosedur untuk mengambil data dari database berdasarkan id customer.
ambilSemua () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data member.
rubah () : void	Prosedur yang dipanggil ketika proses ubah data customer, melakukan perintah update data pada database.
hapus () : void	Prosedur yang dipanggil ketika proses menghapus data customer data pada database.

18. *Class* admin_view

Class admin_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan admin. Atribut dan operasi yang dimiliki *class* admin_view dapat dilihat pada gambar 3.40



Gambar 3.40 Deskripsi *Class* admin_view

a. Deskripsi Atribut

Class admin_view tidak memiliki atribut.

b. Deskripsi Operasi

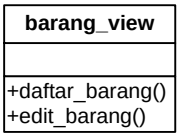
Deskripsi operasi *class* admin_view dapat dilihat pada tabel 3.40

Tabel 3.39 Deskripsi Operasi *Class* admin_view

Nama	Keterangan
login () : void	Prosedur untuk menampilkan form login
pengaturan_akun () : void	Prosedur untuk menampilkan form ubah data admin

19. *Class* barang_view

Class barang_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengunjung dan customer. Atribut dan operasi yang dimiliki *class* barang_view dapat dilihat pada gambar 3.41



Gambar 3.41 Deskripsi *Class* barang_view

a. Deskripsi Atribut

Class barang_view tidak memiliki atribut.

b. Deskripsi Operasi

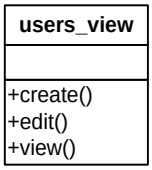
Deskripsi operasi *class* barang_view dapat dilihat pada tabel 3.41

Tabel 3.40 Deskripsi Operasi *Class* barang_view

Nama	Keterangan
daftar_barang () : void	Prosedur untuk menampilkan form data barang
edit_barang () : void	Prosedur untuk menampilkan form ubah data barang

20. *Class* member_view

Class users_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan member. Atribut dan operasi yang dimiliki *class* users_view dapat dilihat pada gambar 3.42



Gambar 3.42 Deskripsi *Class* member_view

a. Deskripsi Atribut

Class users_view tidak memiliki atribut.

b. Deskripsi Operasi

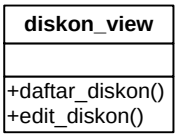
Deskripsi operasi *class* users_view dapat dilihat pada tabel 3.54

Tabel 3.41 Deskripsi Operasi *Class* member_view

Nama	Keterangan
create () : void	Prosedur untuk menampilkan form pendaftaran member
edit () : void	Prosedur untuk menampilkan form ubah data member
view () : void	Prosedur untuk menampilkan form data member

21. *Class* diskon_view

Class diskon_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan admin. Atribut dan operasi yang dimiliki *class* diskon_view dapat dilihat pada gambar 3.43



Gambar 3.43 Deskripsi *Class* diskon_view

a. Deskripsi Atribut

Class diskon_view tidak memiliki atribut.

b. Deskripsi Operasi

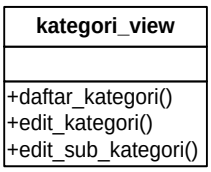
Deskripsi operasi *class* diskon_view dapat dilihat pada tabel 3.43

Tabel 3.42 Deskripsi Operasi *Class* diskon_view

Nama	Keterangan
daftar_diskon () : void	Prosedur untuk menampilkan form daftar diskon
edit_diskon () : void	Prosedur untuk menampilkan form ubah data diskon

22. *Class* kategori_view

Class kategori_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan admin. Atribut dan operasi yang dimiliki *class* kategori_view dapat dilihat pada gambar 3.44



Gambar 3.44 Deskripsi *Class* kategori_view

a. Deskripsi Atribut

Class kategori_view tidak memiliki atribut.

b. Deskripsi Operasi

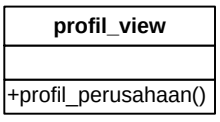
Deskripsi operasi *class* kategori_view dapat dilihat pada tabel 3.34

Tabel 3.43 Deskripsi Operasi *Class* kategori_view

Nama	Keterangan
daftar_kategori () : void	Prosedur untuk menampilkan form daftar kategori
edit_kategori () : void	Prosedur untuk menampilkan form ubah daftar kategori
edit_sub_kategori () : void	Prosedur untuk menampilkan form ubah daftar sub kategori

23. *Class* profil_view

Class profil_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan admin. Atribut dan operasi yang dimiliki *class* profil_view dapat dilihat pada gambar 3.45



Gambar 3.45 Deskripsi *Class* profil_view

a. Deskripsi Atribut

Class profil_view tidak memiliki atribut.

b. Deskripsi Operasi

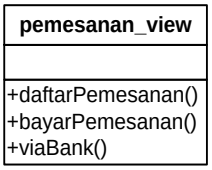
Deskripsi operasi *class* profil_view dapat dilihat pada tabel 3.45

Tabel 3.44 Deskripsi Operasi *Class* kategori_view

Nama	Keterangan
tampil_perusahaan () : void	Prosedur untuk menampilkan form data perusahaan

24. *Class* pemesanan_view

Class pemasanan_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan customer. Atribut dan operasi yang dimiliki *class* pemesanan_view dapat dilihat pada gambar 3.46



Gambar 3.46 Deskripsi *Class* pemesanan_view

a. Deskripsi Atribut

Class pemesanan_view tidak memiliki atribut.

b. Deskripsi Operasi

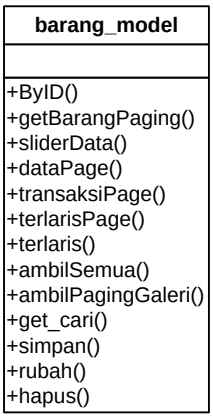
Deskripsi operasi *class* pemesanan_view dapat dilihat pada tabel 3.46

Tabel 3.45 Deskripsi Operasi *Class* pemesanan_view

Nama	Keterangan
daftarPemesanan () : void	Prosedur untuk menampilkan form daftar pemesanan
bayarPemesanan () : void	Prosedur untuk menampilkan form bayar pemesanan
viaBank () : void	Prosedur untuk menampilkan form bayar via bank

25. *Class* *class* barang_model

Class barang_model merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan barang. Atribut dan operasi yang dimiliki *class* barang_model dapat dilihat pada gambar 3.46



Gambar 3.47 Deskripsi *Class* barang_model

a. Deskripsi Atribut

Class barang_model tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* barang_model dapat dilihat pada tabel 3.39

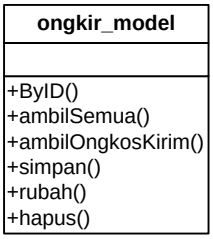
Tabel 3.46 *class* barang_model

Nama	Keterangan
save () : void	Prosedur yang dipanggil ketika proses menyimpan data barang pada database setelah pembuatan account baru.
getById () : void	Prosedur untuk mengambil data dari database berdasarkan barang
ambilSemua () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data barang

rubah () : void	Prosedur yang dipanggil ketika proses ubah data barang, melakukan perintah update data pada database.
hapus () : void	Prosedur yang dipanggil ketika proses menghapus data barang data pada database.
terlaris () : void	Prosedur yang akan memanggil session data barang terlaris

26. *Class* *class* ongkir_model

Class ongkir_model merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan ongkir. Atribut dan operasi yang dimiliki *class* ongkir_model dapat dilihat pada gambar 3.48



Gambar 3.48 Deskripsi *Class* ongkir_model

a. Deskripsi Atribut

Class ongkir_model tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* ongkir_model dapat dilihat pada tabel 3.48

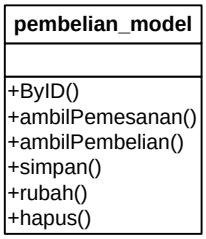
Tabel 3.47 *class* ongkir_model

Nama	Keterangan
simpan () : void	Prosedur yang dipanggil ketika proses menyimpan data ongkir pada database
getById () : void	Prosedur untuk mengambil data dari database ongkir

ambilSemua () : void	Prosedur untuk memanggil data ongkir dari database
ambilOngkosKirim () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data ongkir
rubah () : void	Prosedur yang dipanggil ketika proses ubah data ongkir, melakukan perintah update data pada database.
hapus () : void	Prosedur yang dipanggil ketika proses menghapus data congkir pada database.

27. *Class* *class* pembelian_model

Class pembelian_model merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pembelian. Atribut dan operasi yang dimiliki *class* pembelian_model dapat dilihat pada gambar 3.49



Gambar 3.49 Deskripsi *Class* pembelian_model

a. Deskripsi Atribut

Class pembelian_model tidak memiliki atribut.

b. Deskripsi Operasi

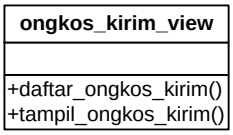
Deskripsi operasi *class* pembelian_model dapat dilihat pada tabel 3.49

Tabel 3.48 *class* pembelian_model

Nama	Keterangan
ById () : void	Prosedur untuk mengambil data dari database berdasarkan id diskon.
ambilPemesanan () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan data pemesanan
ambilSPembelian () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data pembelian
simpan () : void	Prosedur yang dipanggil ketika proses menyimpan data diskon pada database.
rubah () : void	Prosedur yang dipanggil ketika proses ubah data diskon, melakukan perintah update data pada database.
hapus () : void	Prosedur yang dipanggil ketika proses menghapus data diskon data pada database.

28. *Class* *class* ongkos_kirim_view

Class ongkos_kirim_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan ongkos kirim Atribut dan operasi yang dimiliki *class* ongkos_kirim_view dapat dilihat pada gambar 3..50



Gambar 3.50 Deskripsi *Class* ongkos_kirim_view

a. Deskripsi Atribut

Class ongkos_kirim_view tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* ongkos_kirim_view dapat dilihat pada tabel 3.50

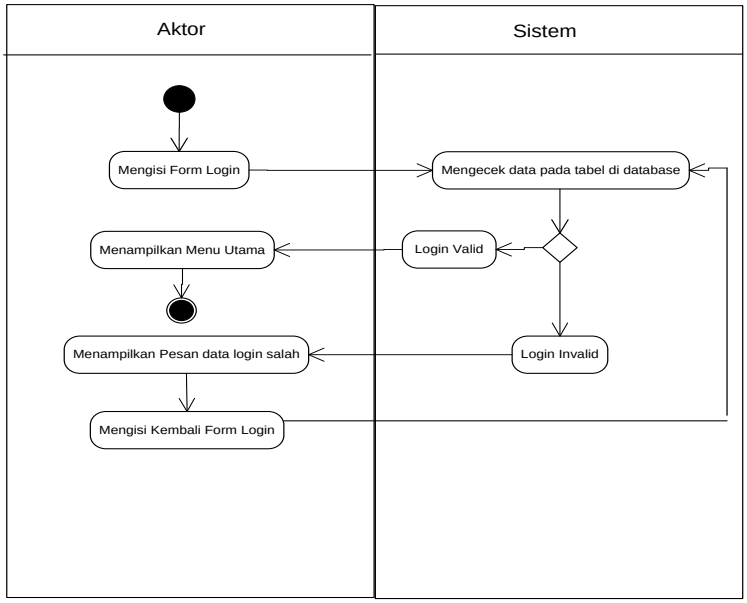
Tabel 3.49 *class* ongkos_kirim_view

Nama	Keterangan
Daftar_ongkos_kirim() : void	Prosedur yang dipanggil untuk menampilkan daftar ongkos kirim.
tampil_ongkos_kirim () : void	Prosedur mengambil data dari database, digunakan untuk menampilkan data ongkos kirim

3.7.4 *Activty Diagram*

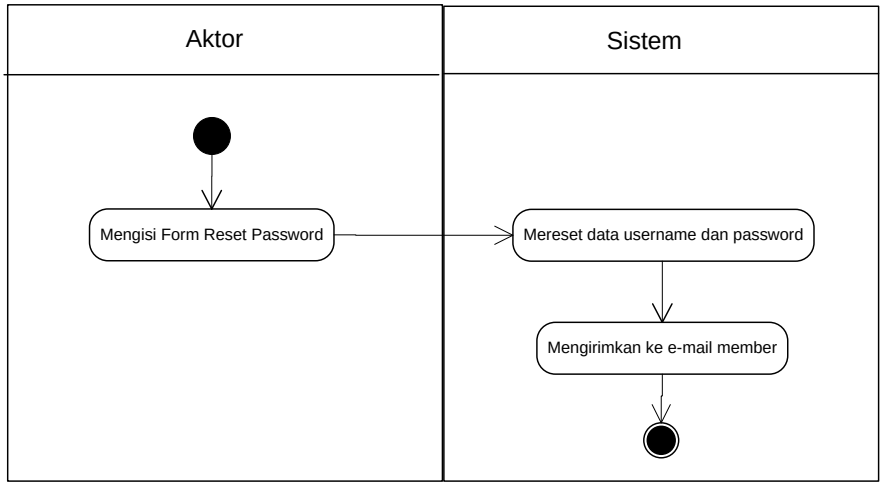
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

3.7.4.1 *Activty* Diagram Login



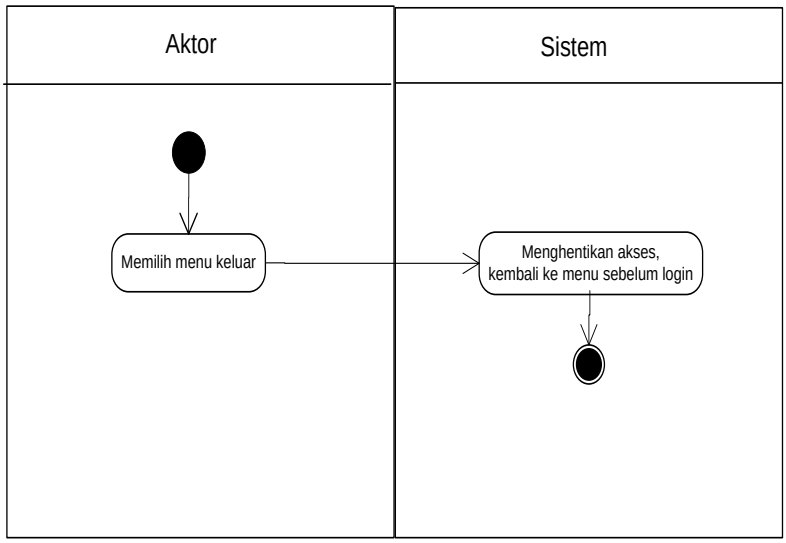
Gambar 3.51 *Activty* Diagram Login

3.7.4.2 *Activty* Diagram Reset Password



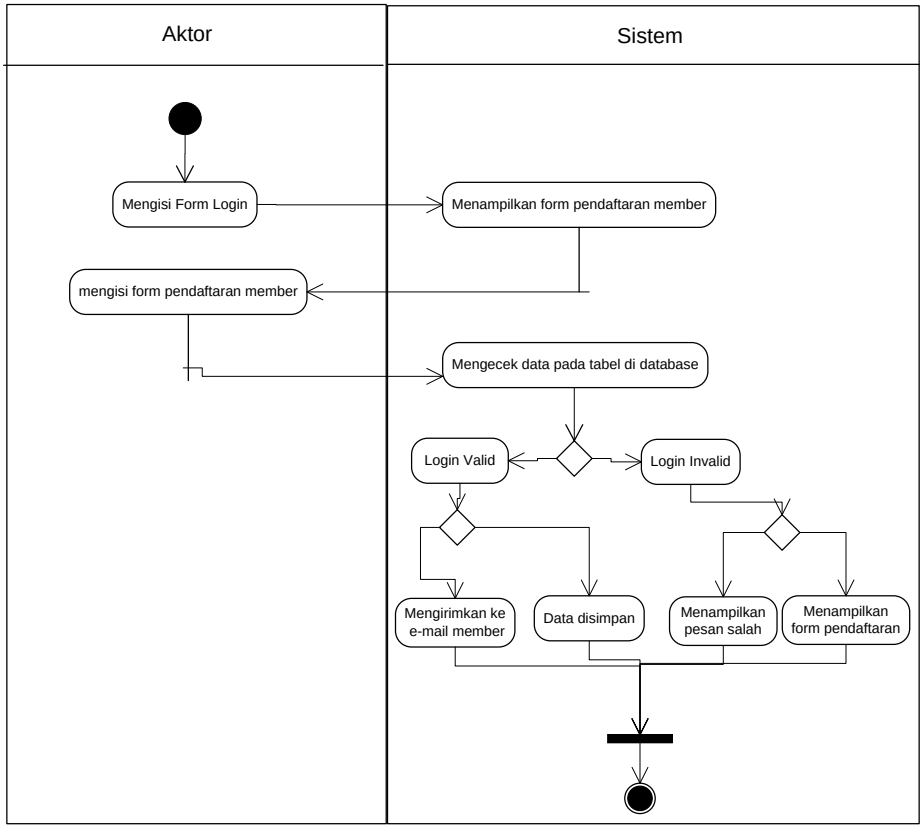
Gambar 3.52 *Activty* Diagram Reset Password

3.7.4.3 *Activty* Diagram Logout



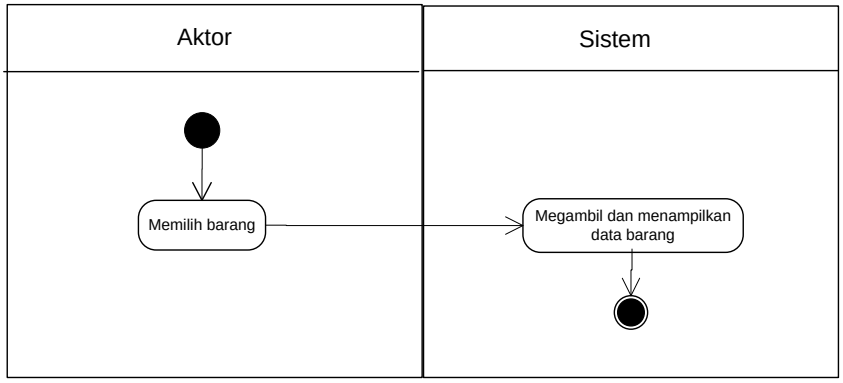
Gambar 3.53 *Activty* Diagram Logout

3.7.4.4 *Activty* Diagram Pendaftaran Member



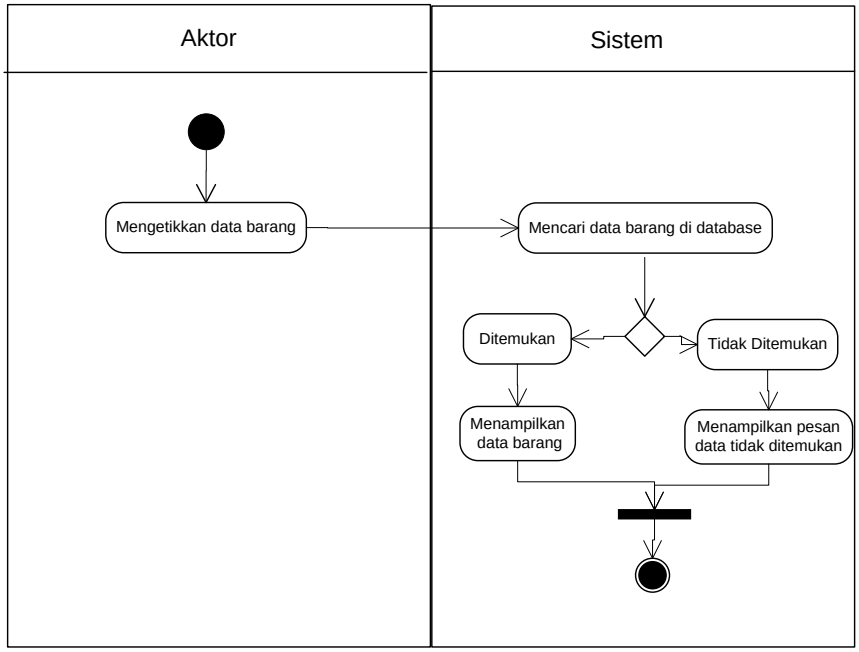
Gambar 3.54 *Activty* Diagram Pendaftaran Member

3.7.4.5 *Activty* Diagram Lihat Barang



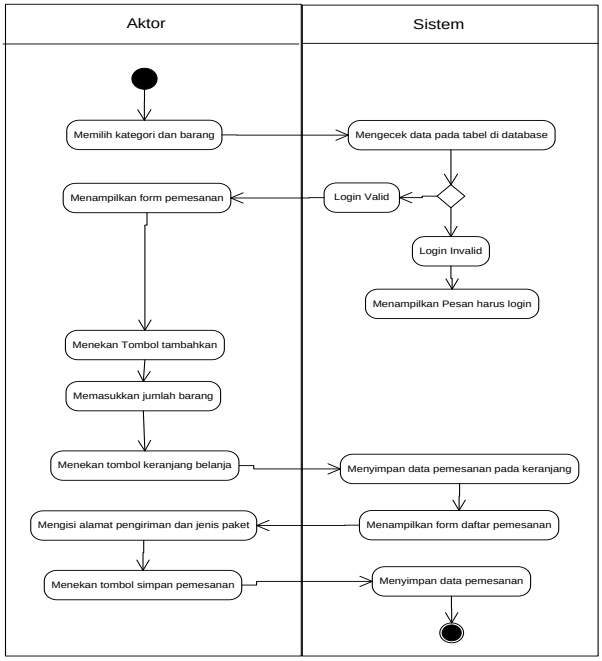
Gambar 3.55 *Activty* Diagram Lihat Barang

3.7.4.6 *Activty* Diagram Pencarian Barang



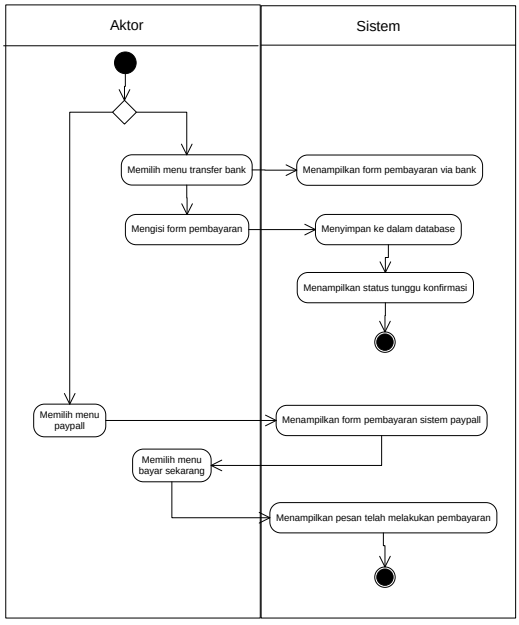
Gambar 3.56 *Activty* Diagram Pencarian Barang

3.7.4.7 *Activty* Diagram Pemesanan Barang



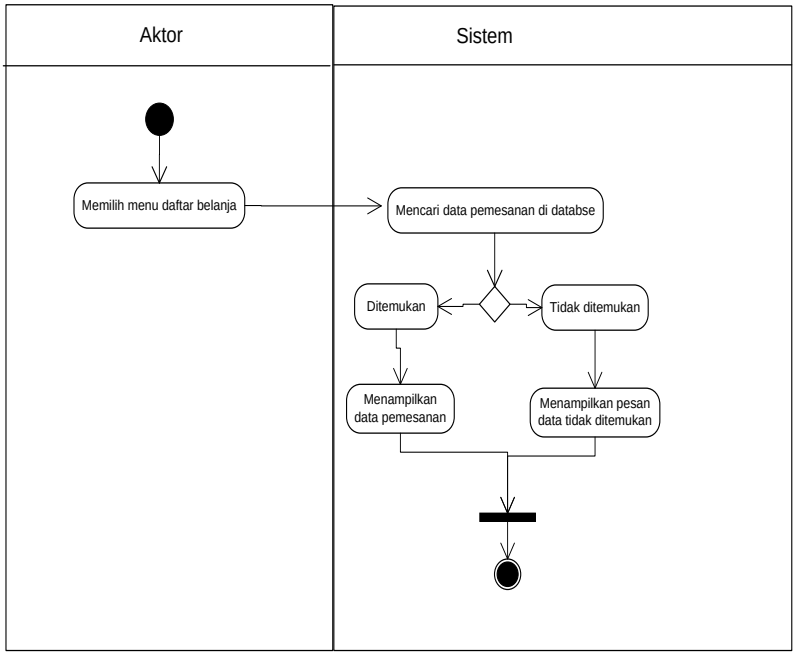
Gambar 3.57 *Activty* Diagram Pemesanan Barang

3.7.4.8 *Activty* Diagram Konfirmasi Pembayaran



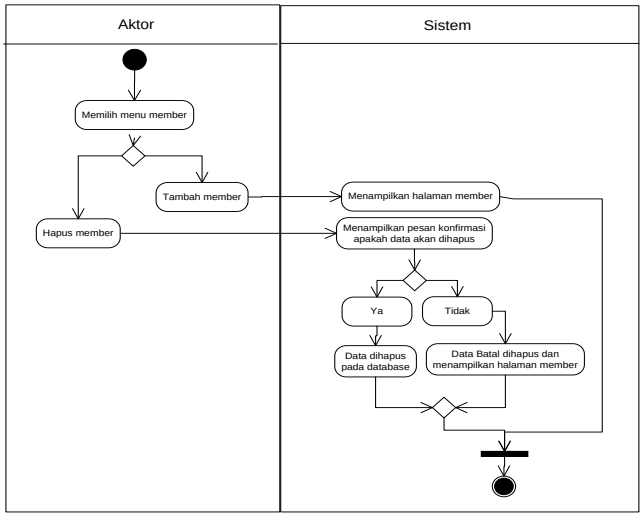
Gambar 3.58 *Activty* Diagram Konfirmasi Pembayaran

3.7.4.9 *Activty* Diagram Cek Status Order



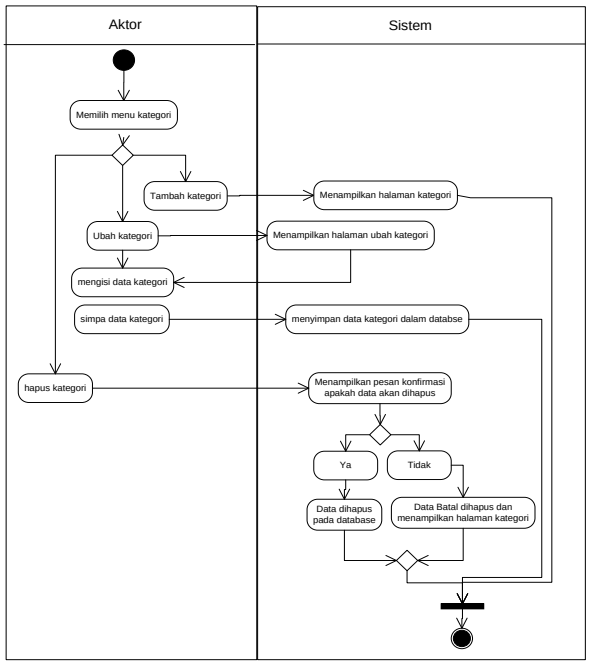
Gambar 3.59 *Activty* Diagram Pemesanan Barang

3.7.4.10 Diagram Pengolahan Data Member



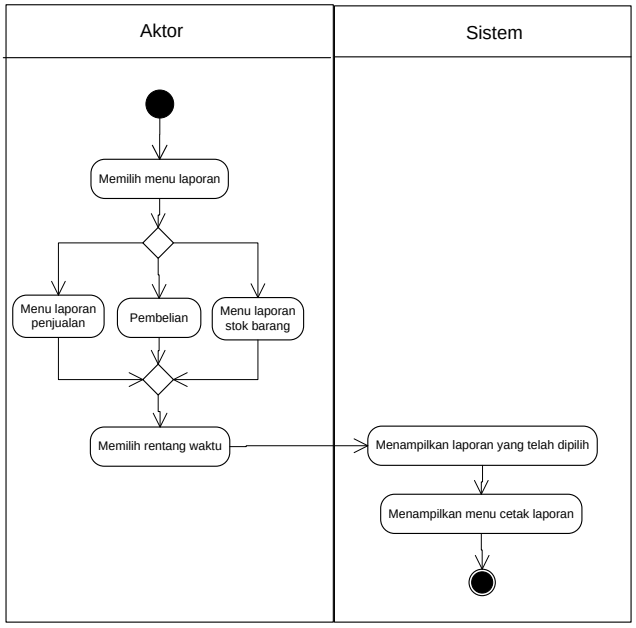
Gambar 3.60 *Activty* Diagram Diagram Pengolahan Data Member

3.7.4.11 *Activty* Diagram Pengolahan Data Kategori



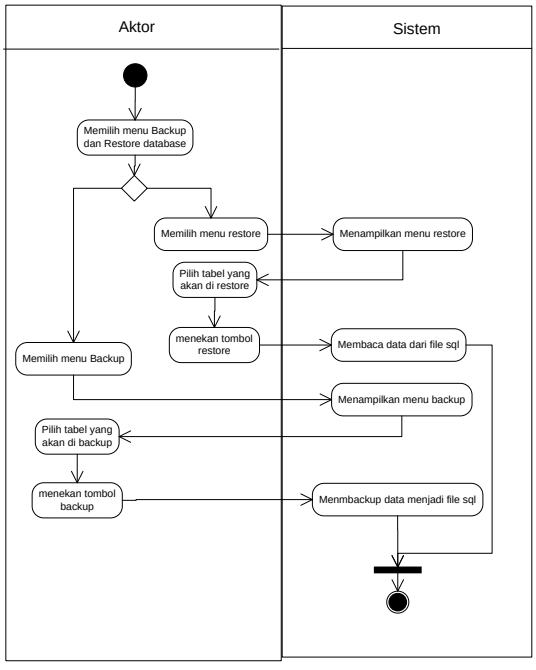
Gambar 3.61 *Activty* Diagram Diagram Pengolahan Data Kategori

3.7.4.12 *Activty* Diagram Pengolahan Data Laporan



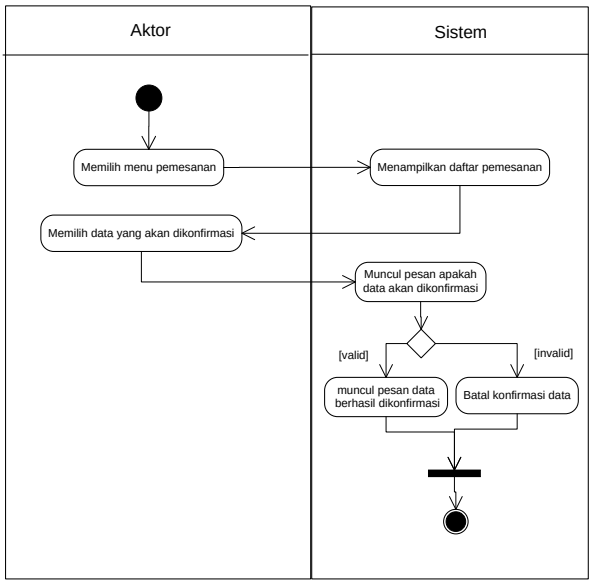
Gambar 3.62 *Activty* Diagram Pengolahan Data Laporan

3.7.4.13 *Activty* Diagram Backup dan Restore Database



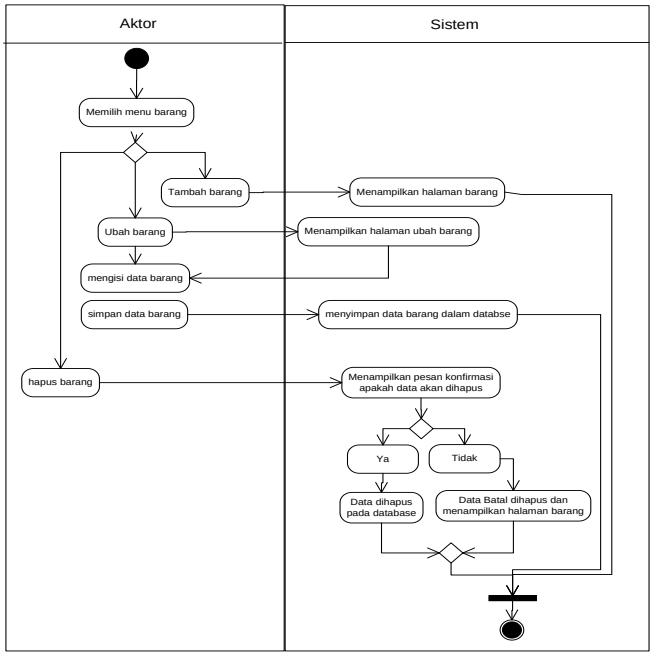
Gambar 3.63 *Activty* Diagram Backup dan Restore Database

3.7.4.14 *Activty* Diagram Konfirmasi Pembayaran



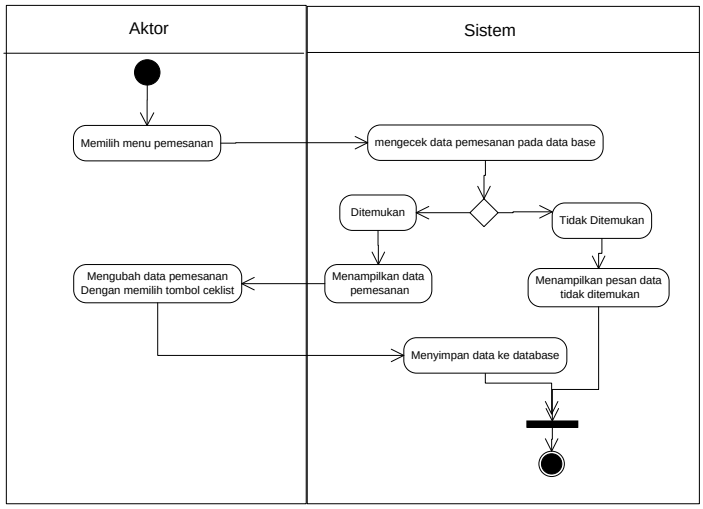
Gambar 3.64 *Activty* Diagram Konfirmasi Pembayaran

3.7.4.15 Diagram Pengolahan Data Barang



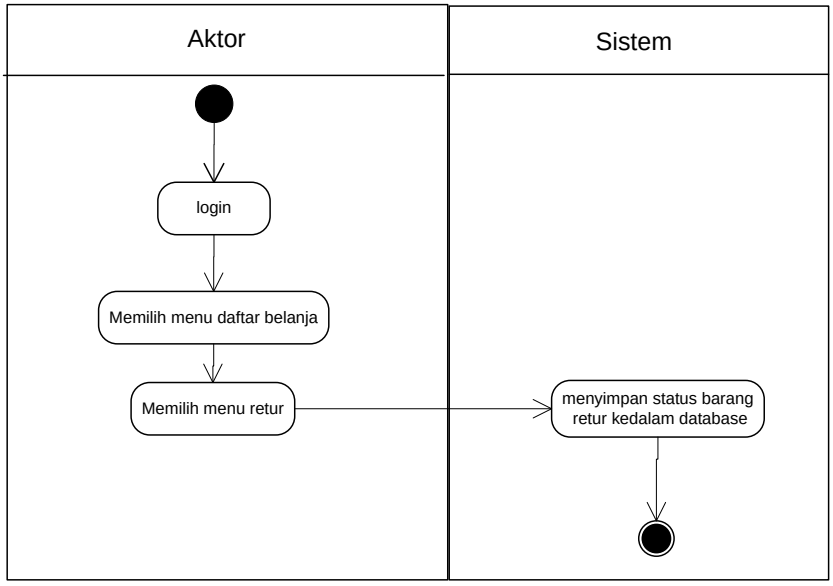
Gambar 3.65 *Activty* Diagram Diagram Pengolahan Data Barang

3.7.4.16 *Activty Activty* Diagram Pengolahan Data Pemesanan



Gambar 3.66 *Activty* Diagram Pengolahan Data Pemesanan

3.7.4.17 *Activty* Diagram Retur Barang

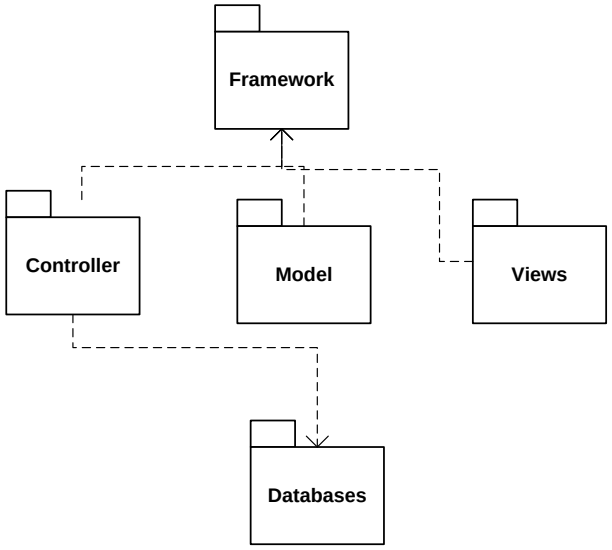


Gambar 3.67 *Activty* Diagram Retur Barang

3.7.5 Package Diagram

Package diagram adalah sebuah bentuk pengelompokkan yang memungkinkan untuk mengambil setiap bentuk di UML dan mengelompokkan

elemen-elemennya dalam tingkatan unit yang lebih tinggi. *Package diagram* untuk pengelompokkan *class* pada *E-commerce* Mitra Teknik dapat dilihat pada gambar 3.68

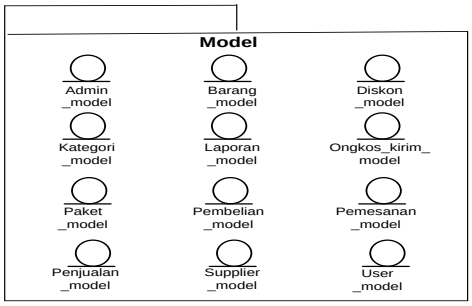


Gambar 3.68 *Package* Diagram *E-commerce* Mitra Teknik

Deskripsi untuk *package* diagram adalah sebagai berikut :

1. Package Model

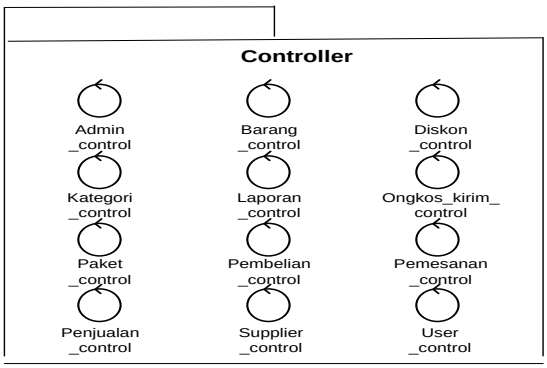
Package Model berisi *class-class* model yang mengakses langsung pada *database* untuk dapat mengolah data. Isi dari *package* Models dapat dilihat pada gambar 3.69



Gambar 3.69 *Package* Model

2. Package Controller

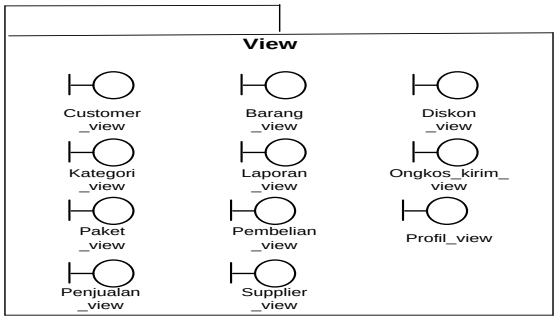
Package Controller berisi *class-class control* yang merupakan pengatur dalam setiap proses yang ada. Isi dari *package Controller* dapat dilihat pada gambar 3.70



Gambar 3.70 *Package* Controller

3. Package View

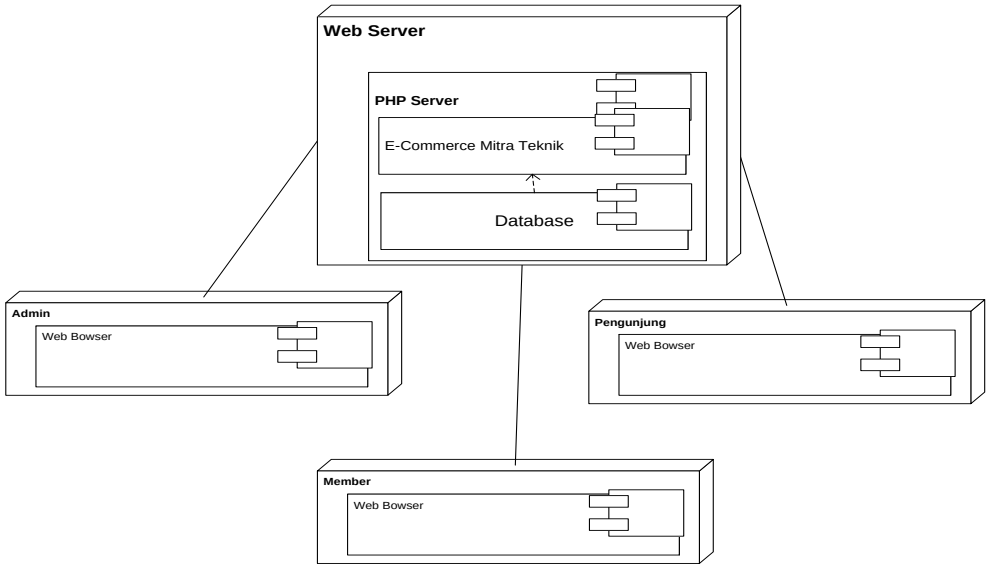
Package View berisi *class-class* view yang digunakan untuk menampilkan interface yang dapat berupa tampilan tabel atau form. Isi dari *Package* View dapat dilihat pada gambar 3.71



Gambar 3.71 *Package* View

3.7.6 Deployment Diagram

Deployment diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras mana. *Deployment* diagram *E-commerce* pada Mitra Teknik dapat dilihat pada gambar 3.51



Gambar 3.72 *Deployment* diagram

3.7.7 Perancangan Sistem

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi. Tahapan ini meliputi mengkonfigurasi

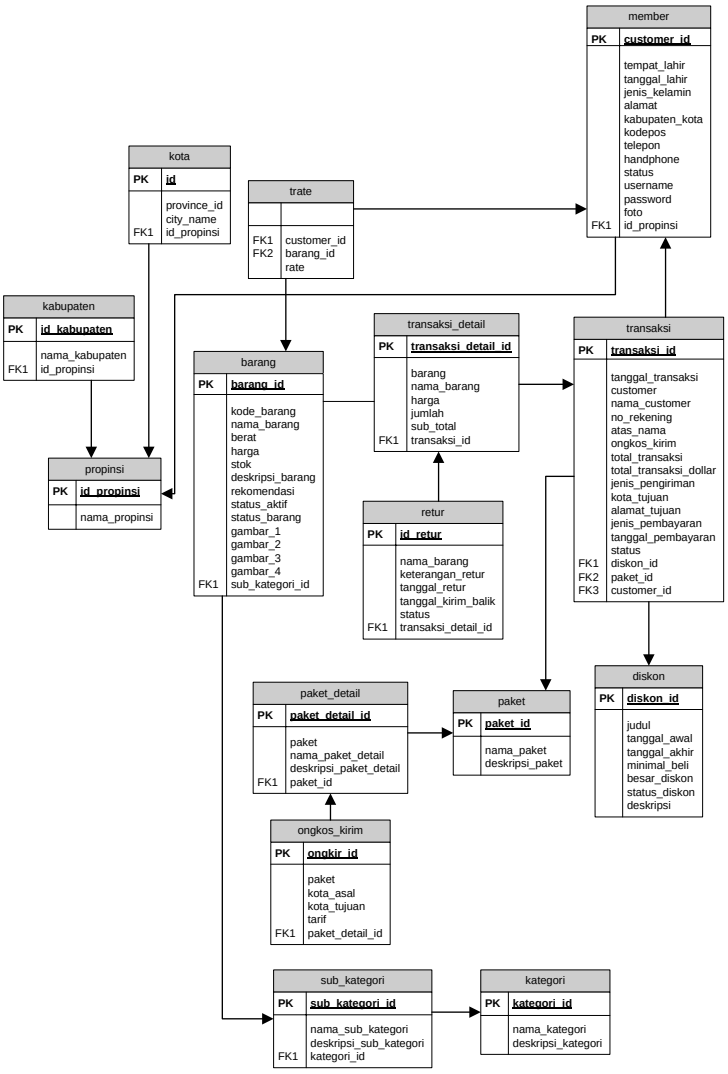
komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi sistem akan mendapatkan hasil yang baik berdasarkan rancang bangun yang telah ditetapkan pada tahap akhir analisis sistem.

3.7.8 Perancangan Data

Perancangan data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai.

3.7.8.1Skema Relasi

Skema relasi adalah hubungan antar data yang digunakan dalam sistem yang akan dibangun.



Gambar 3.73 Diagram relasi *e-commerce* Mitra Teknik

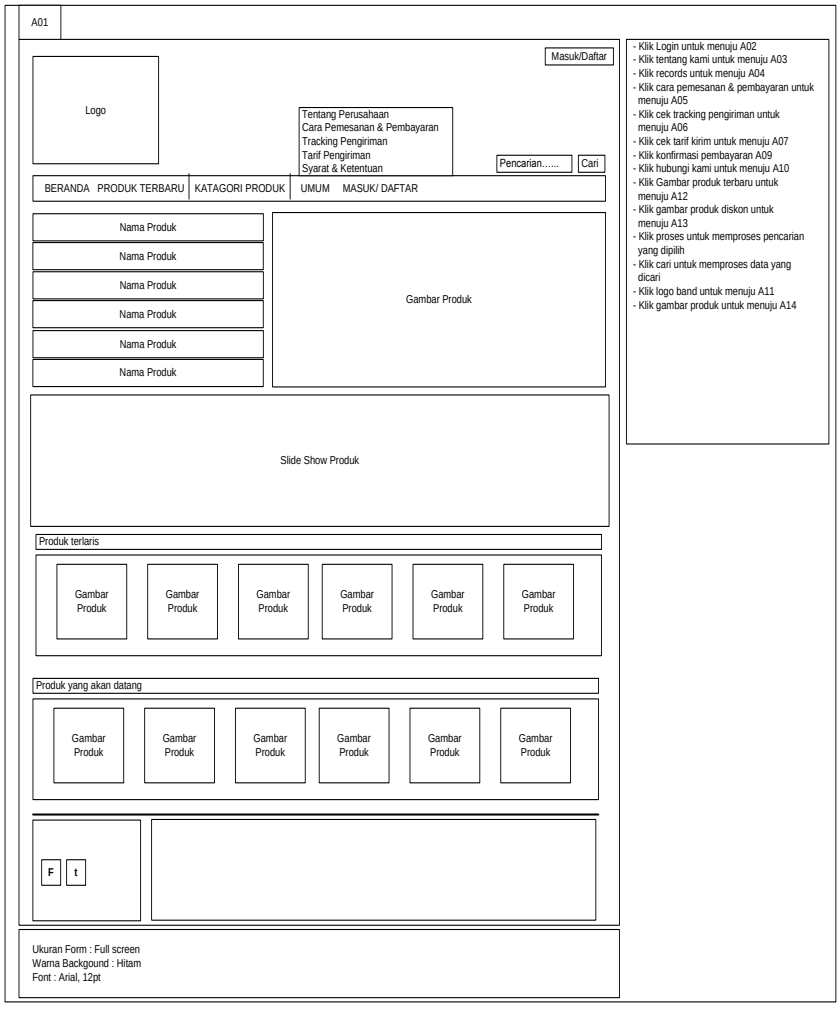
3.7.9 Arsitektur Menu

Menu dirancang untuk menggambarkan segala macam bentuk kegiatan yang bisa dilakukan dalam program aplikasi, seperti *input* data, cari data dan membuat report berdasarkan data *input*. Struktur menu yang dirancang sesuai dengan level pengguna sistem. Terdapat 3 pengguna aplikasi *e-commerce* pada Mitra Teknik yaitu admin, member dan pengunjung. Adapun struktur menu dati setiap level pengguna adalah sebagai berikut :

3.7.10 Perancangan Antarmuka

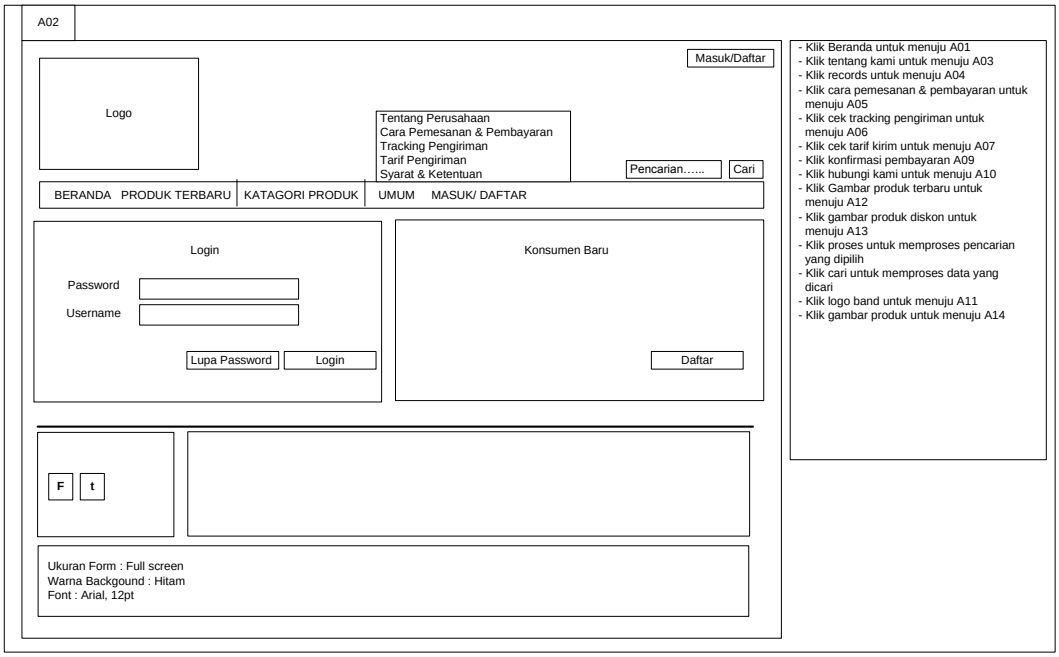
Interface atau antar muka merupakan tampilan dari suau program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan *user*. Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *user*. Untuk memudahkan proses pembuatan aplikasi maka terlebih dahulu membuat rancangan desain tampilannya. Perancangan ini meliputi rancanagan sebagai berikut :

1. Perancangan Halaman Beranda (A01)



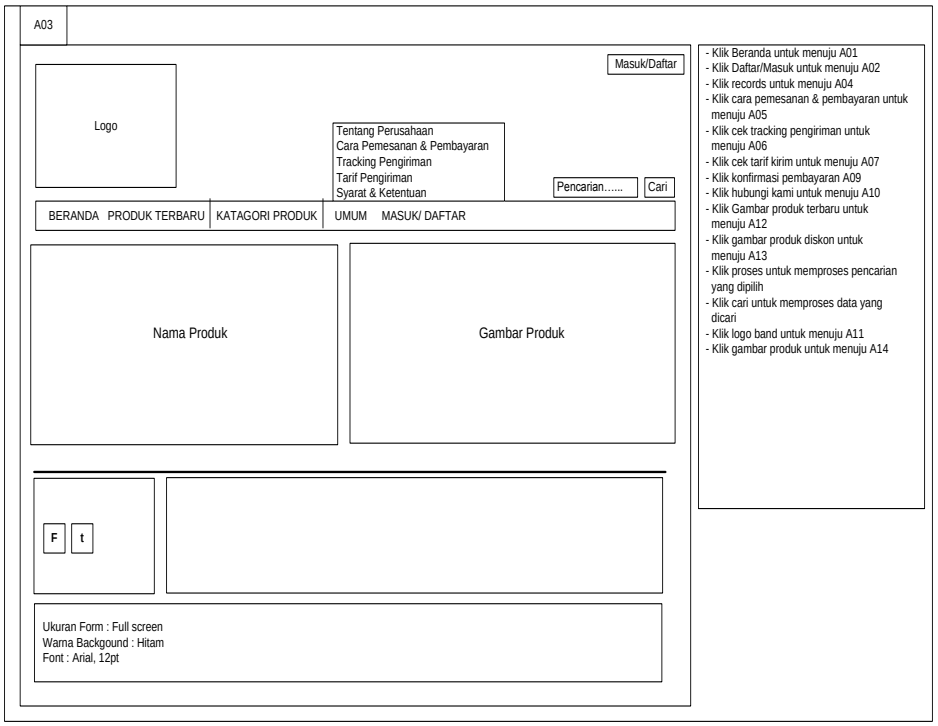
Gambar 3.74 Perancangan Tampilan Halaman Beranda

2. Perancangan Halaman Daftar/ Masuk (A02)



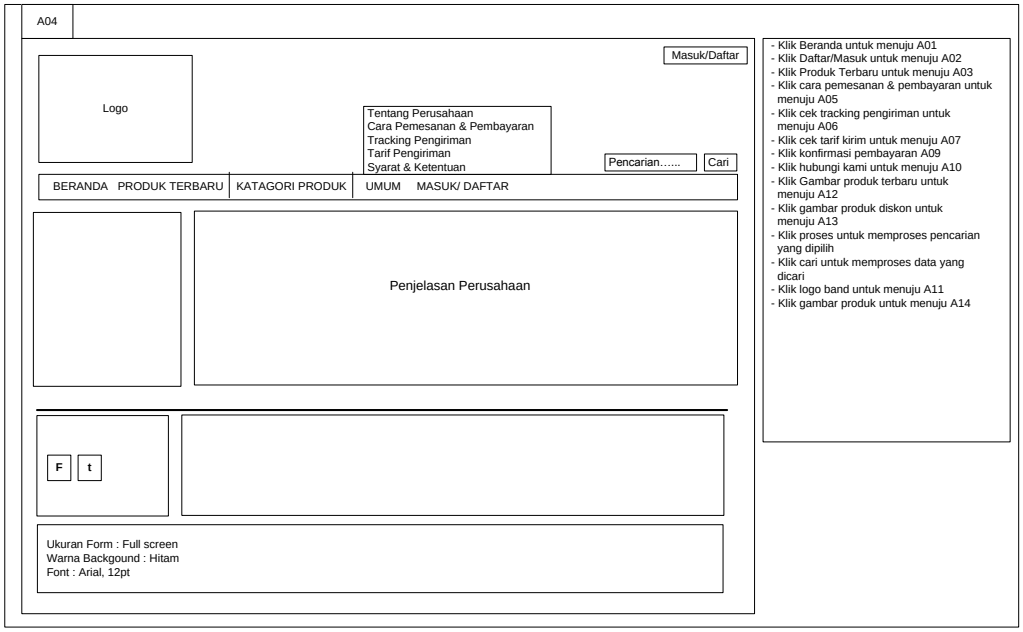
Gambar 3.75 Perancangan Tampilan Halaman Daftar/ Masuk

3. Perancangan Halaman Barang Terbaru (A03)



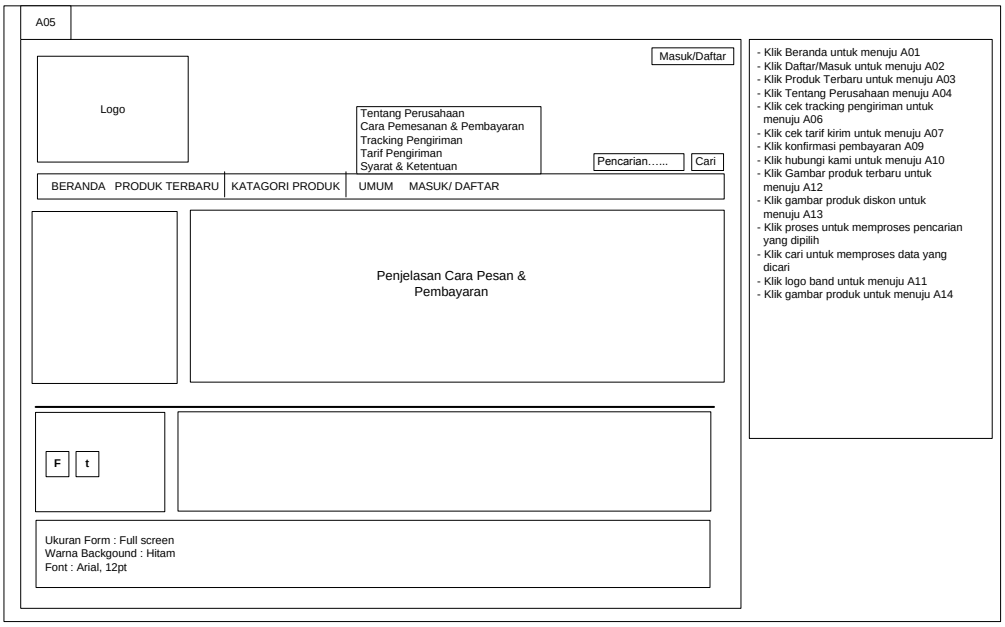
Gambar 3.76 Perancangan Tampilan Halaman Barang Terbaru

4. Perancangan Halaman Tentang Perusahaan (A04)



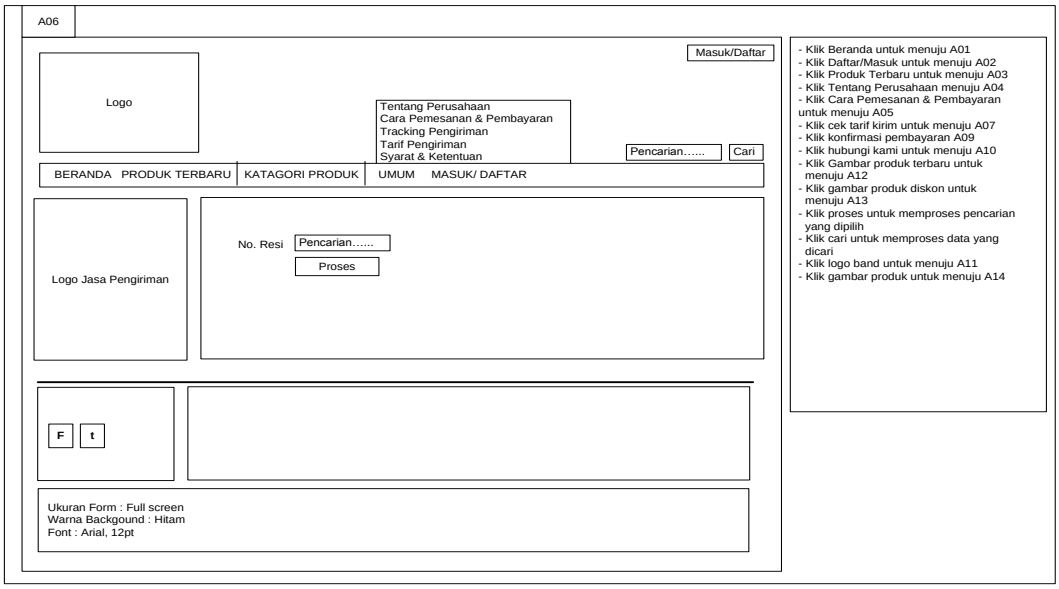
Gambar 3.77 Perancangan Tampilan Halaman Tentang Perusahaan

5. Perancangan Halaman Cara Pemesanan & Pembayaran (A05)



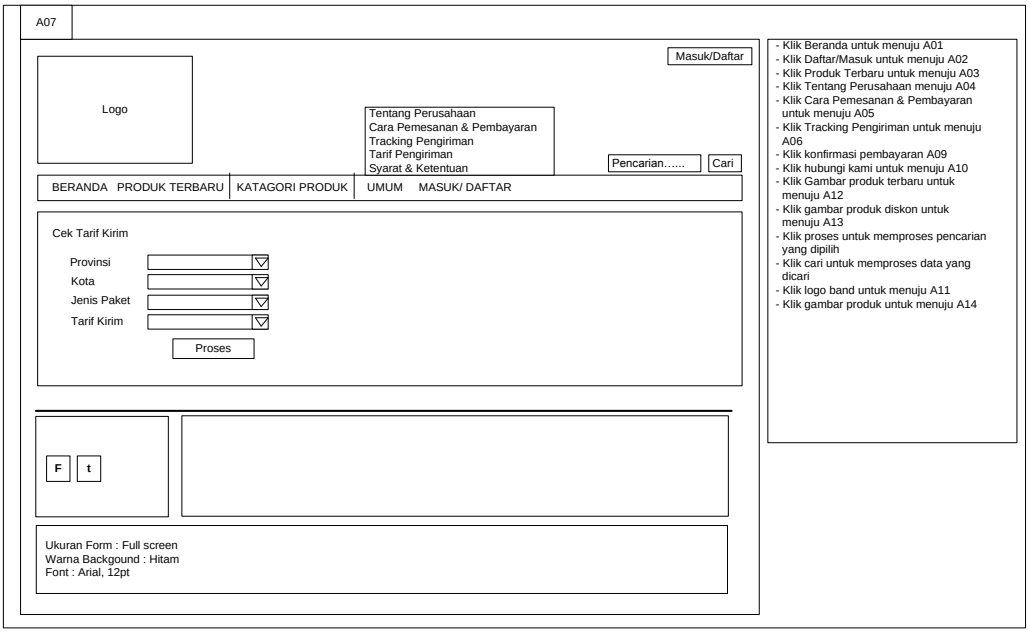
Gambar 3.78 Perancangan Tampilan Cara Pemesanan & Pembayaran

6. Perancangan Halaman *Tracking* Pengiriman (A06)



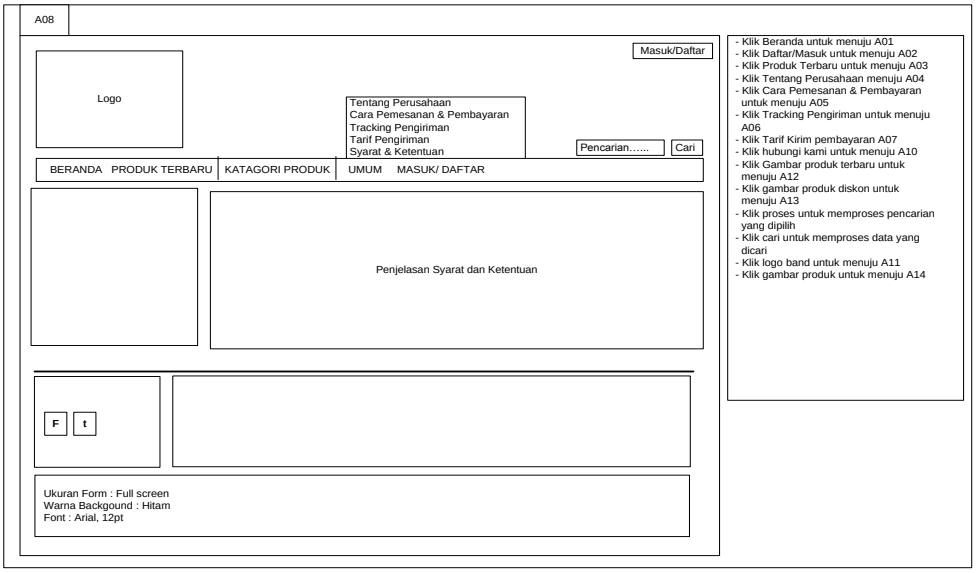
Gambar 3.79 Perancangan Tampilan *Tracking* Pengiriman

7. Perancangan Halaman Tarif Pengiriman (A07)



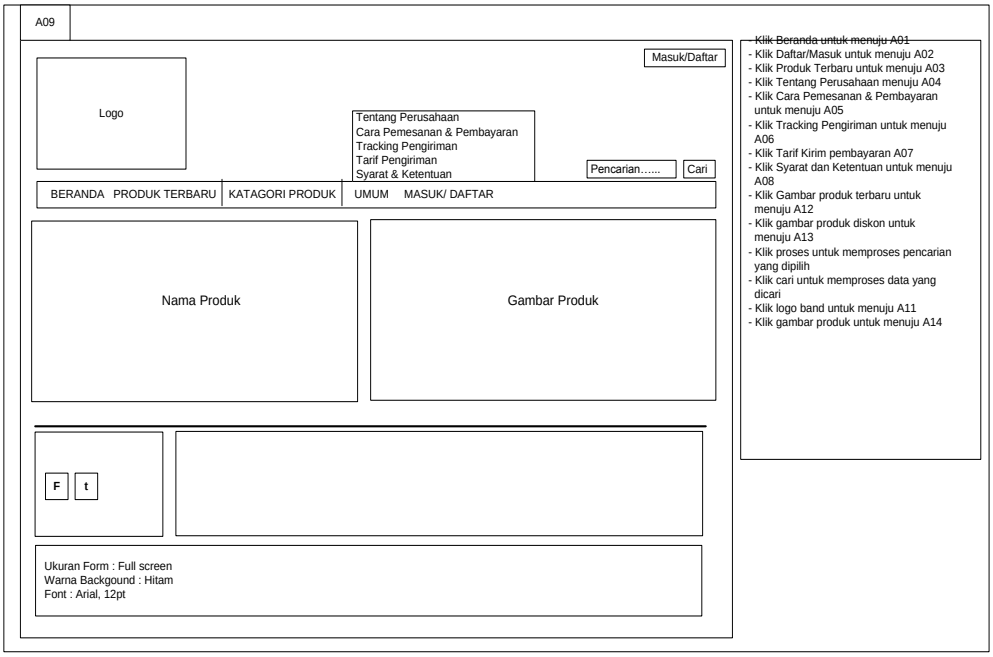
Gambar 3.80 Perancangan Tampilan Tarif Pengiriman

8. Perancangan Halaman Syarat & Ketentuan (A08)



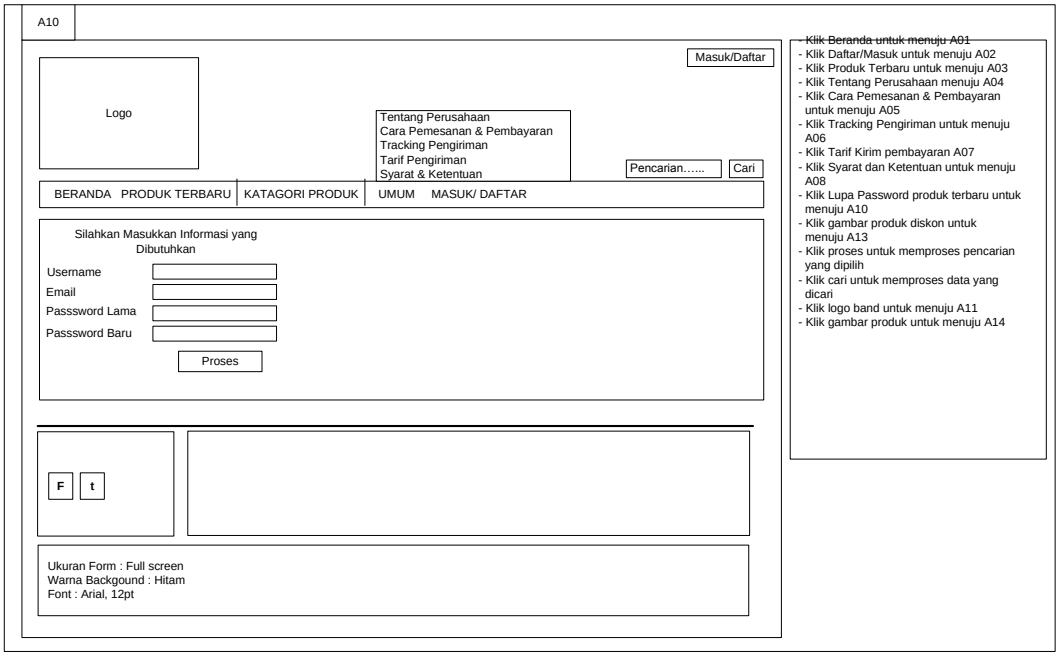
Gambar 3.81 Perancangan Tampilan Syarat & Ketentuan

9. Perancangan Halaman Katagori Barang (A09)



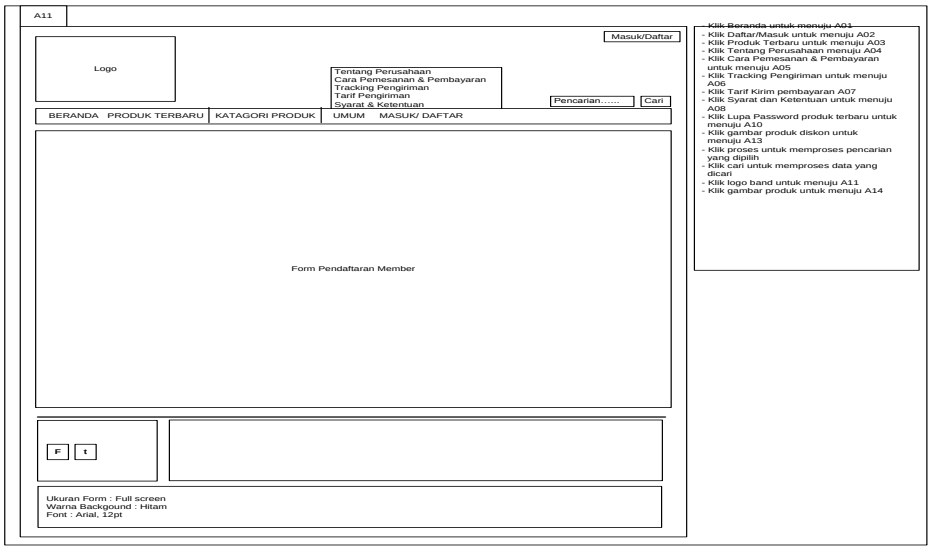
Gambar 3.82 Perancangan Tampilan Katagori Barang

10. Perancangan Halaman Lupa Password (A10)



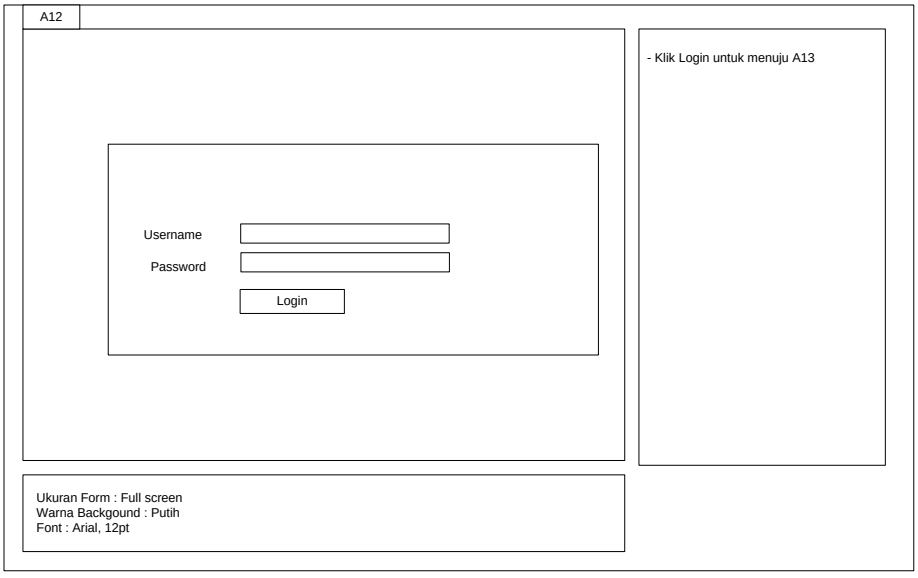
Gambar 3.83 Perancangan Tampilan Lupa Password

11. Perancangan Halaman Pendaftaran (A11)



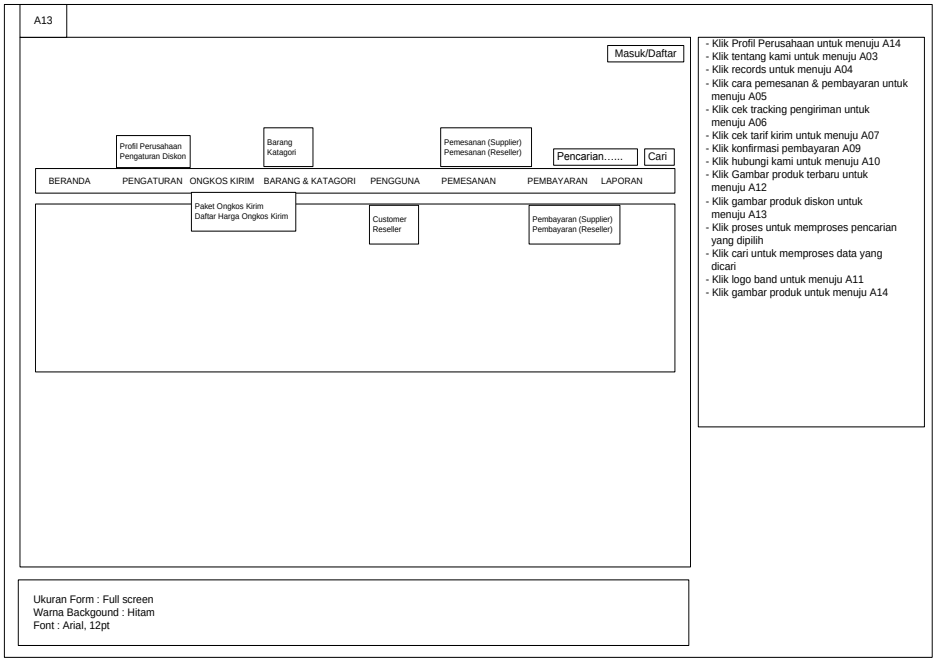
Gambar 3.84 Perancangan Tampilan Pendaftaran

12. Perancangan Login Admin (A12)



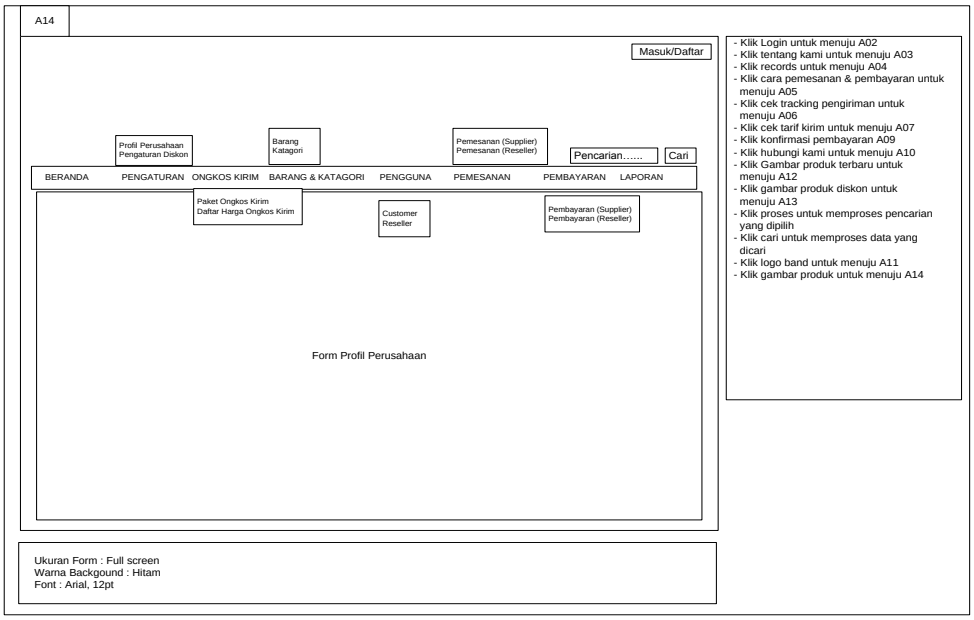
Gambar 3.85 Perancangan Tampilan Login Admin

13. Perancangan Beranda Admin (A13)



Gambar 3.86 Perancangan Tampilan Beranda Admin

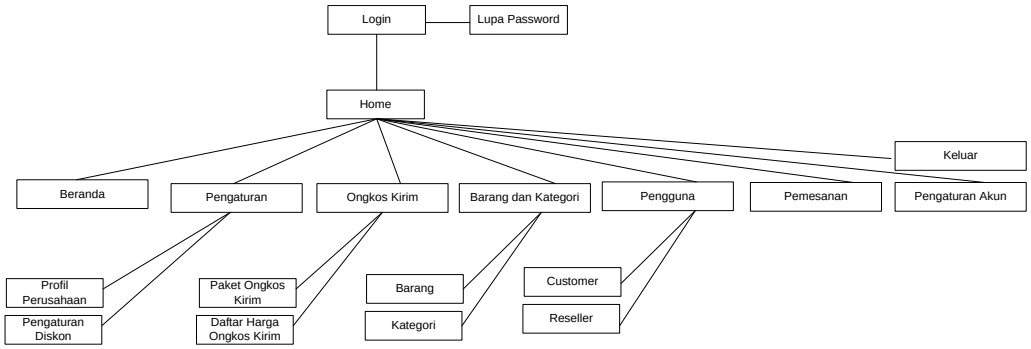
14. Perancangan Tampilan Profil Perusahaan (A14)



Gambar 3.87 Perancangan Tampilan Profil Perusahaan

1. Perancangan struktur menu *admin*

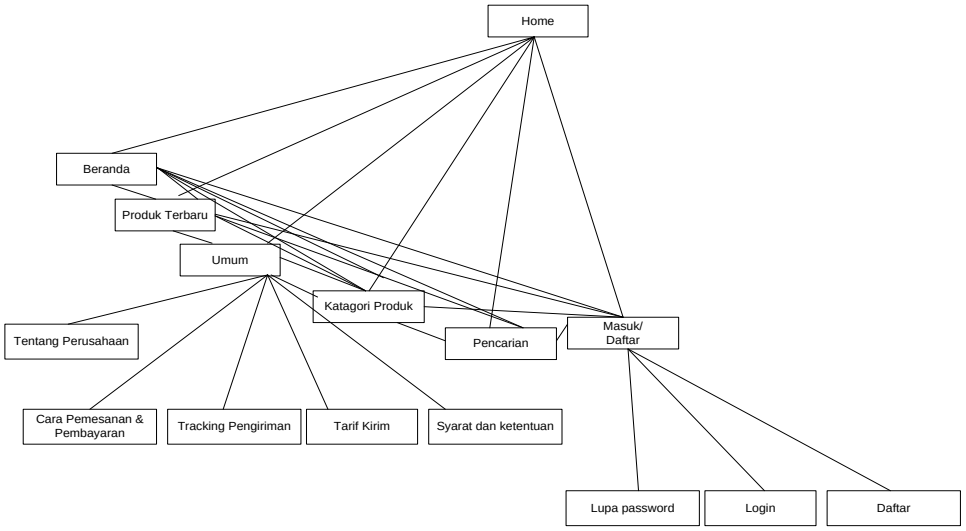
Perancangan struktur menu admin dapat dilihat pada gambar 3.89



Gambar 3.88 struktur menu *admin*

2. Perancangan Struktur Menu Pengunjung

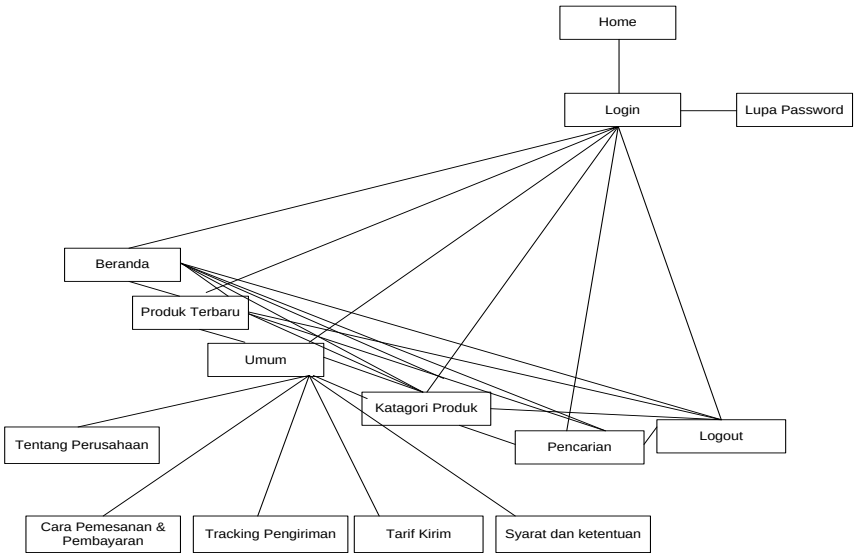
Perancangan struktur menu Pengunjung dapat dilihat pada gambar 3..90



Gambar 3.89 struktur menu pengunjung

2. Perancangan Struktur Menu Member

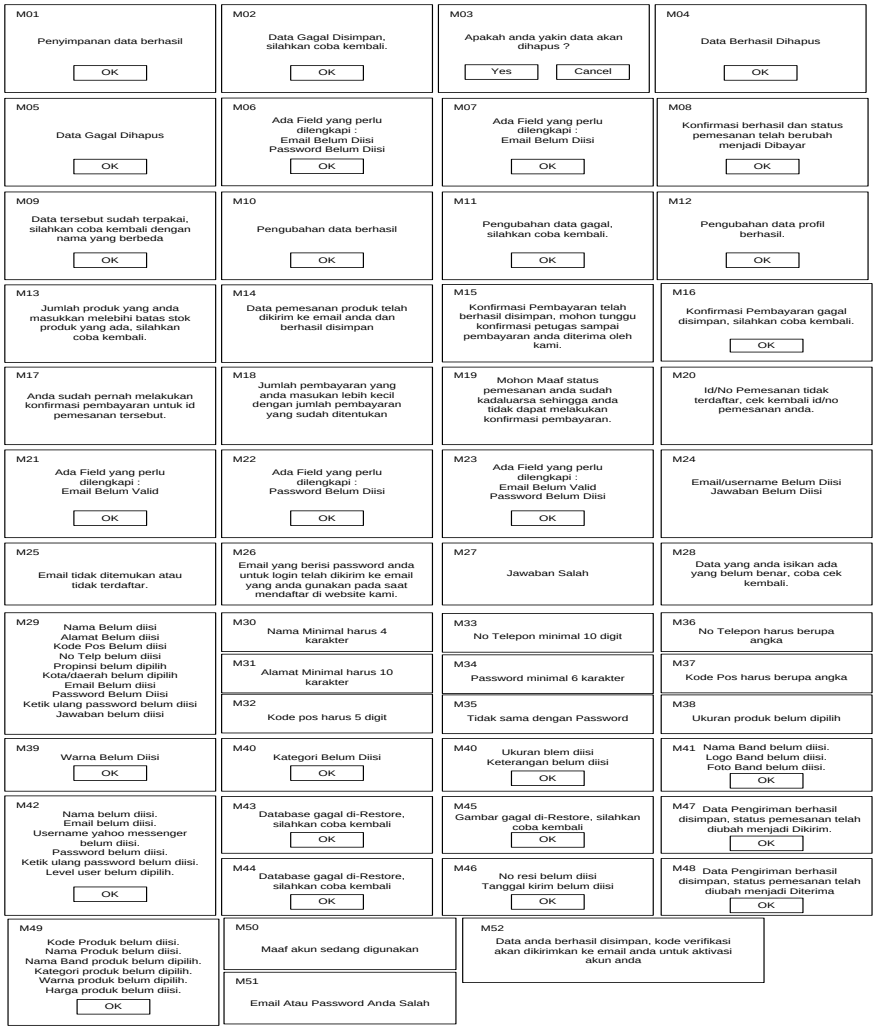
Perancangan struktur menu Pengunjung dapat dilihat pada gambar 3.91



Gambar 3.90 struktur menu pengunjung

3.7.11 Perancangan Pesan

Perancangan pesan merupakan bahasa yang disampaikan oleh sistem kepada pengguna untuk menyampaikan informasi sesuai yang dilakukan oleh *pengguna.* Tampilan pesan yang ada pada aplikasi *e-commerce* pada Mitra Teknik adalah sebagai berikut : Perancangan pesan dapat dilihat pada gambar pada gambar 3.70



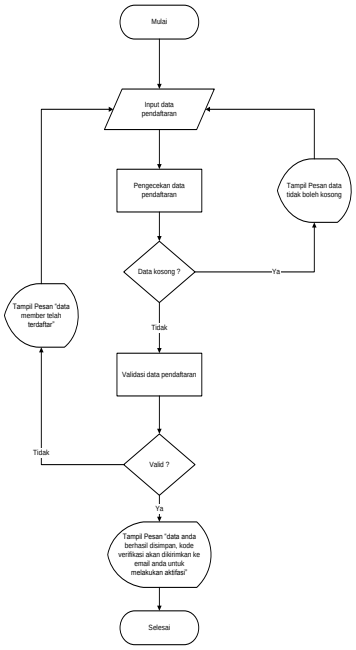
Gambar 3.91 Perancangan Pesan

3.7.12 Perancangan Prosedural

Perancangan prosedural adalah tata cara atau urutan langkahlangkah untuk melakukan suatu proses. Prosedural ini akan digunakan sebagai algoritma dasar dalam mengkodekan prosedur yang ada. *Tools* yang digunakan adalah flowchart program. Adapun perancangan prosedural adalah sebagai berikut:

3.7.12.1 *Flowchart* Pendaftaran Member

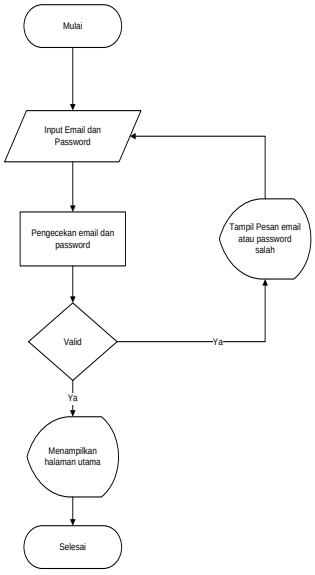
Prosedur pendaftaran member merupakan proses pengisian data member beserta pengecekan data yang dimasukan untuk mendaftar menjadi seorang *member*.*Flowchart* pendaftaran *member* dapat dilihat pada gambar 3.93



Gambar 3.92 *Flowchart* pendaftaran *member*

3.7.12.2 Flowchart Login

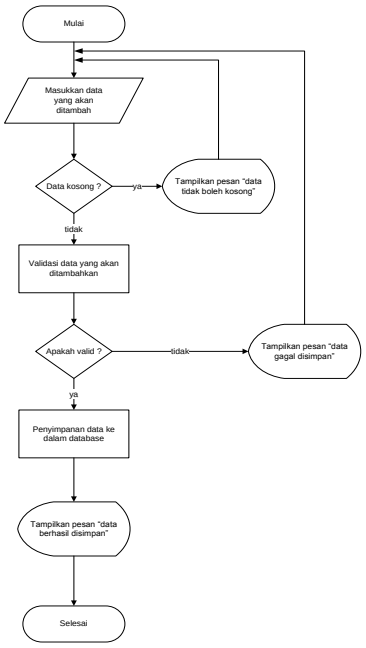
Diagram alir pada prosedur *login* adalah sebagai berikut :



Gambar 3.93 Flowchart login

3.7.12.3 *Flowchart* Tambah Data

Diagram alir pada prosedur tambah data adalah sebagai berikut :



Gambar 3.94 *Flowchart* Tambah Data