BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Berdasarkan penelitian yang telah dilakukan di Pabrik Baju 12Titan, ditemukan masalah-masalah sebagai berikut:

1. Proses pemesanan dilakukan secara langsung atau telepon sehingga pemesan dari luar kota merasa kesulitan untuk melakukan pemesanan dan memberikan desain yang sesuai dengan keinginan sehingga pihak pabrik juga merasa kesulitan dalam menentukan harga.
2. Data pesanan diolah secara manual sehingga sulit dalam pengorganisasian untuk proses selanjutnya.

3.1.1 Analisis Sistem yang Berjalan

Prosedur merupakan urutan kegiatan yang tepat dan tahapan-tahapan yang menerangkan mengenai proses apa yang dikerjakan, siapa yang mengerjakan proses tersebut, bagaimana proses itu dapat dikerjakan dan dokumen apa yang dilibatkan.

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa prosedur yang sedang berjalan, yaitu prosedur pemesanan, prosedur pembelian bahan, prosedur produksi pesanan, dan prosedur pembayaran.

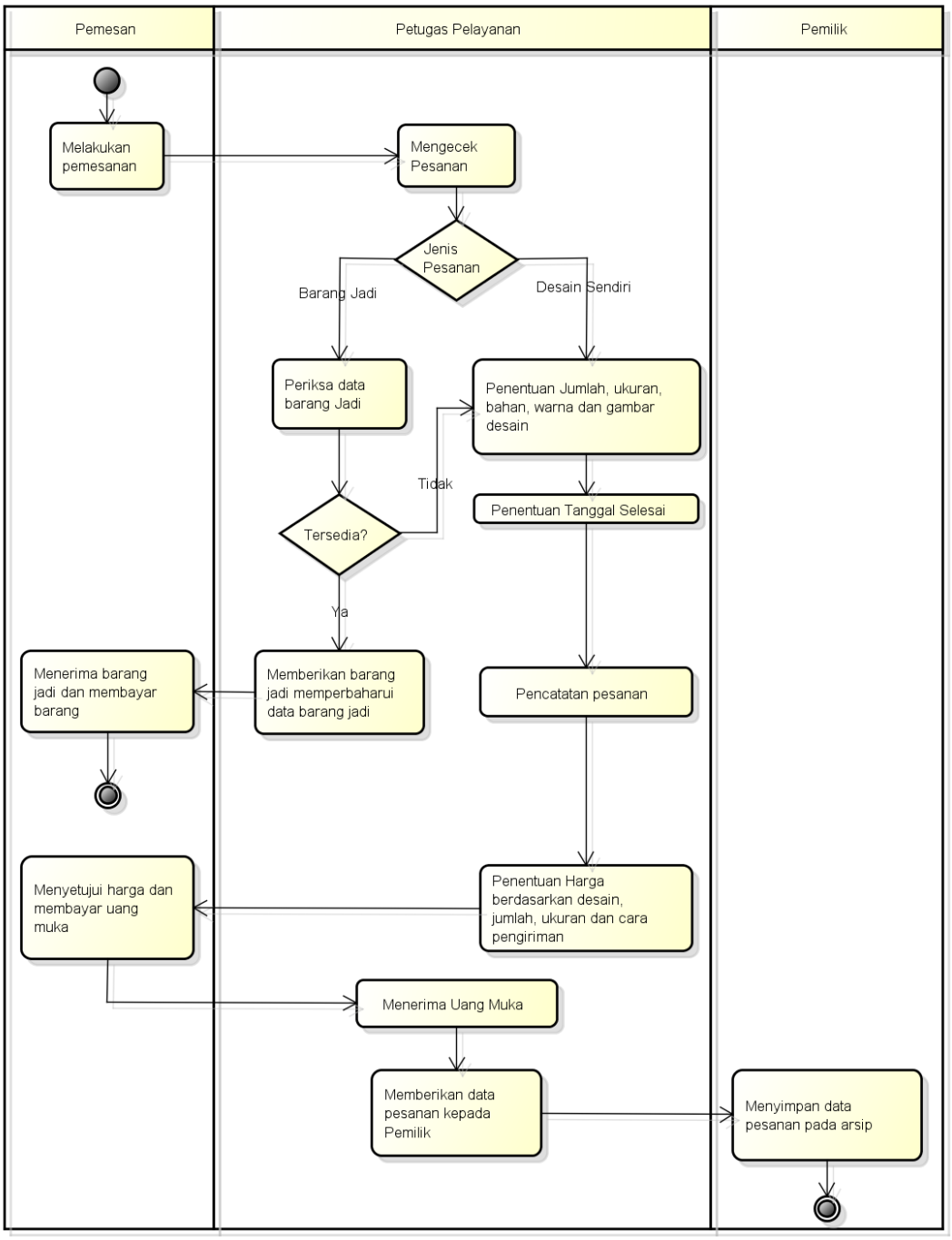
3.1.1.1 Prosedur Pemesanan Barang

Prosedur pemesanan yang sedang berjalan yaitu sebagai berikut :

1. Pemesan datang kepada petugas pelayanan untuk melakukan pesanan.

2. Petugas menanyakan apakah akan memesan barang jadi atau memesan sesuai desain sendiri. Jika pemesan akan membeli barang jadi maka petugas akan mengecek data barang jadi apakah barang tersedia. Jika barang tersedia sesuai jumlah permintaan maka barang akan langsung diberikan kepada pemesan dan memperbaharui data barang jadi, setelah itu pemesan melakukan pembayaran barang. Jika barang sudah habis atau kurang dari jumlah permintaan, maka akan dilakukan pemesanan dan dicatat sebagai pesanan baru.
3. Pemesan menentukan jumlah barang beserta ukuran, bahan, warna, dan cara pengiriman barang yang akan dipesan.
4. Petugas menghitung estimasi tanggal selesai pesanan.
5. Petugas mencatat pesanan dan menentukan harga berdasarkan jumlah, bahan, warna, cara pengiriman dan desain dari barang yang dipesan.
6. Pemesan menyetujui harga dan memberikan uang muka.
7. Petugas menerima uang muka dan memberikan data pesanan kepada pemilik.
8. Pemilik menyimpan data pesanan sebagai arsip.

Activity diagram pemesanan barang dapat dilihat pada gambar 3.1.



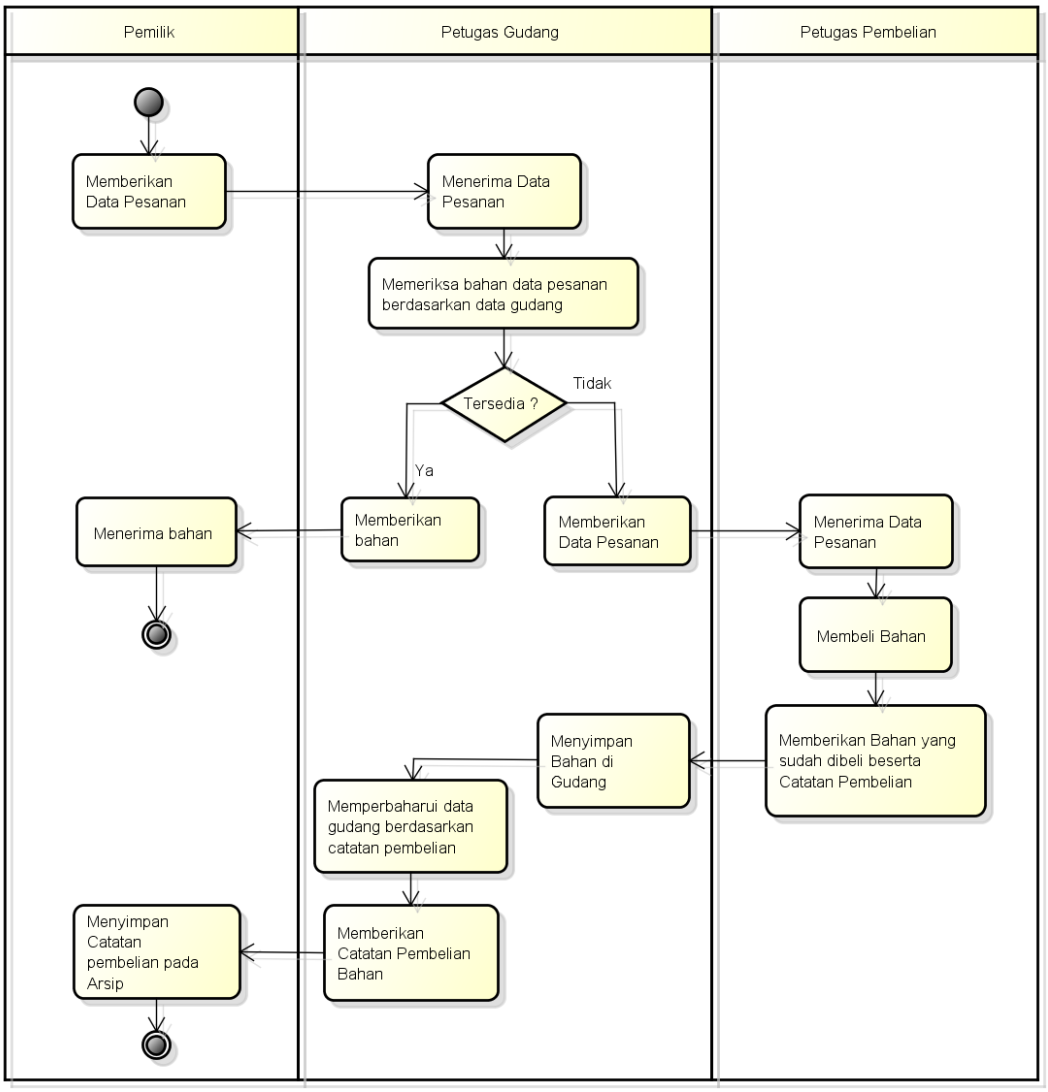
Gambar 3.1 *Activity* Diagram Pemesanan Barang

3.1.1.2 Prosedur Pengadaan Bahan

Prosedur pengadaan bahan merupakan proses untuk membeli bahan baku yang dibutuhkan untuk membuat pesanan. Prosedur pengadaan bahan yang sedang berjalan adalah sebagai berikut :

1. Pemilik memberikan data pesanan kepada petugas gudang.
2. Petugas gudang akan memeriksa pada data gudang apakah bahan yang diperlukan tersedia atau tidak. Jika ada maka petugas akan memberikan bahan kepada pemilik. Jika bahan tidak ada maka petugas gudang akan memberikan data pesanan kepada petugas pembelian.
3. Petugas pembelian menerima data pesanan dan membeli bahan yang diperlukan.
4. Petugas pembelian memberikan bahan yang sudah dibeli, catatan pembelian kepada petugas gudang.
5. Petugas gudang memperbaharui data gudang berdasarkan catatan pembelian.
6. Petugas gudang memberikan catatan pembelian kepada pemilik.
7. Pemilik menyimpan catatan pembelian pada arsip.

Activity diagram pengadaan bahan dapat dilihat pada gambar 3.2.



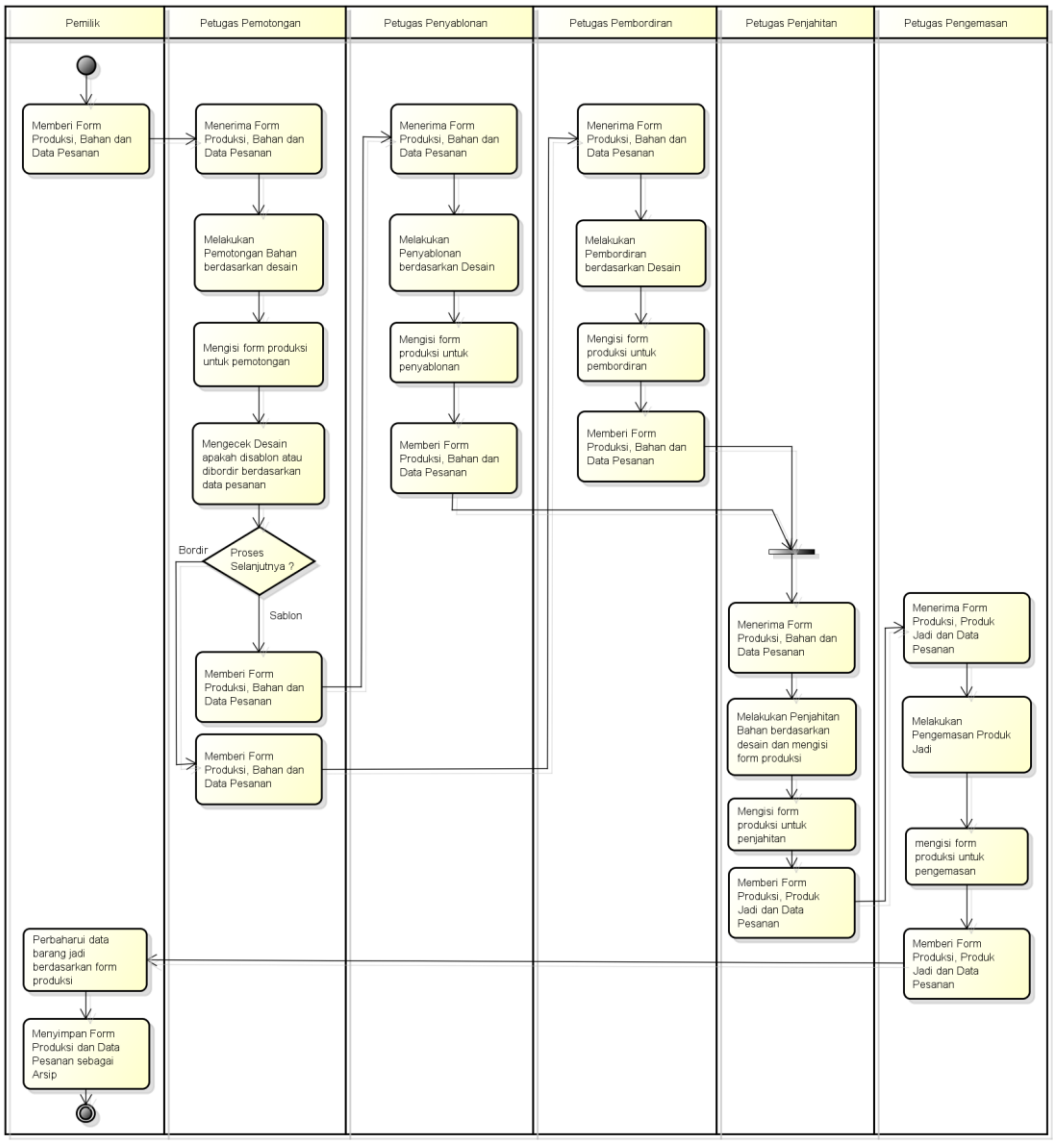
Gambar 3.2 *Activity* Diagram Pengadaan Bahan

3.1.1.3 Prosedur Produksi Pesanan

Prsoedur produksi pesanan adalah proses dimana pemilik memberikan tugas kepada bagian produksi yang mencakup pemotongan, penyablonan, pembordiran, penjahitan dan pengemasan. Pada akhirnya pemilik akan menerima informasi bahwa barang pesanan sudah selesai dikemas dan siap dikirim. Prosedur produksi yang sedang berjalan adalah sebagai berikut:

1. Pemilik memberikan form produksi, desain dan bahan ke petugas pemotongan.
2. Petugas pemotongan melakukan proses pemotongan sesuai desain.
3. Petugas pemotongan mengisi form produksi untuk menandai bahwa proses pemotongan telah selesai dilakukan.
4. Petugas pemotongan mengecek apakah desain pesanan akan disablon atau dibordir. Jika pesanan akan disablon maka akan diberikan kepada petugas sablon atau jika desain pesanan akan dibordir maka akan diberikan kepada petugas bordir.
5. Petugas sablon atau petugas bordir akan melakukan tugasnya yaitu menyablon atau membordir sesuai desain dan mengisi form produksi.
6. Petugas sablon atau bordir akan menyerahkan barang pesanan, form produksi dan desain kepada petugas penjahitan.
7. Petugas penjahitan melakukan proses penjahitan dan mengisi form produksi untuk menandai bahwa proses penjahitan sudah dilakukan.
8. Petugas penjahitan memberikan form produksi, bahan dan desain kepada petugas pengemasan.
9. Petugas pengemasan akan mengemas barang pesanan dan mengisi form produksi.
10. Petugas pengemasan akan memberikan barang yang sudah dikemas dan form produksi serta desain kepada pemilik.
11. Pemilik memperbaharui data barang jadi berdasarkan form produksi dan menyimpannya sebagai arsip.

Activity diagram Produksi pesanan dapat dilihat pada gambar 3.3.



Gambar 3.3 *Activity* Diagram Produksi Pesanan

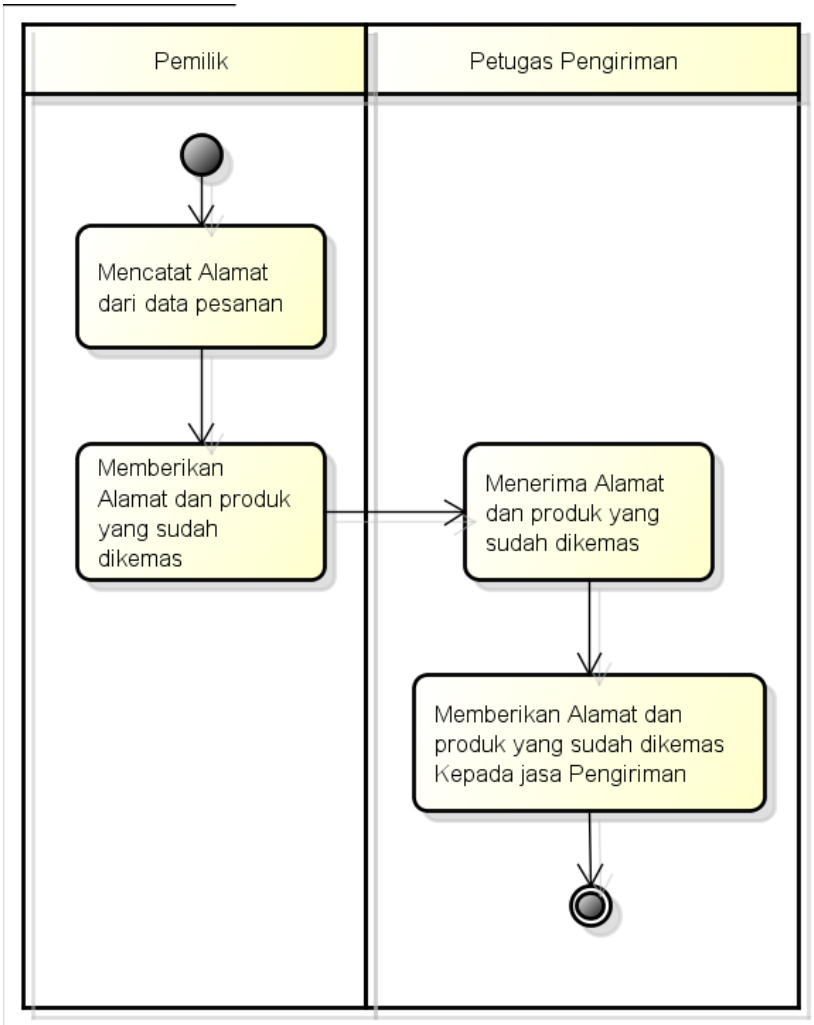
3.1.1.4 Prosedur Pengiriman Barang

Prosedur pengiriman merupakan proses mengirimkan barang kepada konsumen yang memesan.

1. Pemilik yang telah menerima konfirmasi pembayaran memeriksa alamat pada data pesanan kemana barang akan dikirim.

2. Pemilik memberikan barang yang sudah dikemas dan alamat tujuan kepada petugas pengiriman.
3. Petugas pengiriman memberikan barang dan alamat kepada jasa pengiriman barang.

Activity diagram pengiriman barang dapat dilihat pada gambar 3.4.



Gambar 3.4 *Activity* Diagram Pengiriman Barang

3.1.2 Analisis Estimasi Pemesanan

Analisis kasus yang dilakukan yaitu pada proses estimasi penentuan tanggal selesai dari sebuah pemesanan. Penentuan tanggal ini dipengaruhi oleh beberapa faktor, yaitu:

1. Kategori yang dipesan.
2. Jumlah maksimum produksi per kategori dalam satu hari.
3. Jumlah maksimum produksi setiap pesanan dalam satu hari.
4. Jumlah pesanan.

Adapun ketentuan batas jumlah pesanan yang ditetapkan untuk setiap kategori yaitu :

a. Jumlah pesanan minimum untuk setiap kategori adalah 21 potong. Alasannya adalah karena setiap pesanan memiliki desain yang berbeda dengan biaya produksi yang berbeda, dengan jumlah ini biaya tersebut dapat disamaratakan dan juga dapat memenuhi biaya produksi keseluruhan sehingga perusahaan tidak merugi.
b. Jumlah pesanan maksimum untuk setiap kategori adalah 300 potong. Jumlah ini disesuaikan dengan jumlah rata-rata produksi setiap setiap pesanan yaitu 10-20 potong per hari dan untuk menghindari waktu pengerjaan yang terlalu lama.

Estimasi pemesanan akan dilakukan ketika terjadi pemesanan dan telah ditentukan jenis kategori dan jumlah pesanan. Langkah – langkah untuk estimasi pemesanan dijelaskan sebagai berikut:

1. Lihat jenis kategori yang dipilih yang akan menghasilkan jumlah maksimum produksi per kategori dalam satu hari.
2. Lihat jumlah pesanan, lalu bagi dengan jumlah maksimum produksi setiap pesanan dalam satu hari, ini akan menghasilkan lama pengerjaan normal yang dibutuhkan untuk menyelesaikan pesanan.
3. Cek pada data pesanan ada berapa jumlah pesanan yang sedang diproduksi pada hari ini.
4. Jika jumlah pesanan sama dengan jumlah maksimum maka akan dicek jumlah pesanan pada hari ini ditambah 1 hari, jika jumlah pesanan masih sama dengan jumlah maksimum maka akan dicek kembali jumlah pesanan pada hari ini ditambah 2 hari, begitu seterusnya hingga mendapatkan jumlah pesanan kurang dari julmah maksimum. Ini akan menghasilkan lama hari tambahan.
5. Jika kategori yang dipesan lebih dari satu jenis maka hari tambahan akan diambil dari kategori dengan lama hari tambahan yang paling besar.
6. Tambahkan lama hari tambahan dengan lama pengerjaan normal yang akan menentukan lama pengerjaan keseluruhan.
7. Hitung tanggal selesai dengan menambahkan tanggal pemesanan dengan lama pengerjaan keseluruhan.

3.1.2.1 Analisis Estimasi Pemesanan untuk Satu Kategori

Berikut ini adalah contoh perhitungan estimasi pemesanan yang dilakukan untuk satu jenis kategori T-Shirt jika diketahui :

a. Jumlah produksi maksimum untuk kategori T-Shirt dalam satu hari adalah 30 potong.
b. Dari jumlah produksi maksimum tersebut akan dibagi menjadi 10 potong untuk setiap pesanan.
c. Jika jumlah produksi sudah mencapai jumlah produksi maksimum, maka pemesanan yang masuk akan diproses pada hari berikutnya ketika jumlah produksi kurang dari jumlah produksi maksimum.

Sebagai contoh, berikut adalah list pesanan yang masuk pada tanggal 1 Juni 2011 yang dapat dilihat pada tabel 3.1.

Tabel 3.1 List Pesanan

Pemesanan	Tgl Pemesanan	Jml Pesanan
I	1 Juni 2011	30
II	1 Juni 2011	20
III	1 Juni 2011	40
IV	1 Juni 2011	30

Langkah – langkah dalam melakukan perhitungan estimasi pemesanan yaitu sebagai berikut :

1. **Pemesanan I**

   Tanggal pemesanan : **1 Juni 2011**

   Jumlah Pesanan : **30 potong**

   Perhitungan untuk menentukan tanggal selesainya dapat dilihat pada tabel 3.2.

Tabel 3.2 Estimasi Pemesanan I

Pemesanan	Tanggal								
	1	2	3	4	5	6	7	8	9
I		10	10	10					
Jumlah		**10**	**10**	**10**					

Pesanan akan mulai dikerjakan dari tanggal 2 Juni selama 3 hari, sehingga akan diselesaikan pada tanggal **4 Juni 2011.**

2. Pemesanan II

Tanggal pemesanan : **1 Juni 2011**

Jumlah Pesanan : **20 potong**

Perhitungan untuk menentukan tanggal selesainya dapat dilihat pada tabel 3.3.

Tabel 3.3 Estimasi Pemesanan II

Pemesanan	Tanggal								
	1	2	3	4	5	6	7	8	9
I		10	10	10					
II		10	10						
Jumlah		**10**	**10**	**10**					

Pesanan akan mulai dikerjakan dari tanggal 2 Juni selama 2 hari, sehingga akan diselesaikan pada tanggal **3 Juni 2011.**

3. Pemesanan III

Tanggal pemesanan : **1 Juni 2011**

Jumlah Pesanan : **40 potong**

Perhitungan untuk menentukan tanggal selesainya dapat dilihat pada tabel 3.4.

Tabel 3.4 Estimasi Pemesanan III

Pemesanan	Tanggal								
	1	2	3	4	5	6	7	8	9
I		10	10	10					
II		10	10						
III		10	10	10	10				
Jumlah		**30**	**30**	**30**	**10**				

Pesanan akan mulai dikerjakan dari tanggal 2 Juni selama 4 hari, sehingga akan diselesaikan pada tanggal **5 Juni 2011.**

4. Pemesanan IV

Tanggal pemesanan : **1 Juni 2011**

Jumlah Pesanan : **30 potong**

Perhitungan untuk menentukan tanggal selesainya dapat dilihat pada tabel 3.5.

Tabel 3.5 Estimasi Pemesanan IV

Pemesanan	Tanggal								
	1	2	3	4	5	6	7	8	9
I		10	10	10					
II		10	10						
III		10	10	10	10				
IV				10	10	10			
Jumlah		**30**	**30**	**30**	**20**	**20**			

Karena jumlah pesanan yang sedang diproduksi pada tanggal 2 Juni sudah mencapai batas maksimum produksi yaitu 30 maka akan dicek pada hari berikutnya dimana jumlah pesanan yang diproduksi kurang dari jumlah produksi maksimum.

Pesanan akan mulai dikerjakan dari tanggal **4 Juni** selama 3 hari, sehingga akan diselesaikan pada tanggal **6 Juni 2011.** Proses perhitungan ini akan dilakukan kembali ketika ada pemesanan baru yang masuk.

3.1.2.2 Analisis Estimasi Pemesanan untuk Banyak Kategori

Pada pemesanan dengan banyak kategori, member dimungkinkan untuk memesan lebih dari satu jenis produk dengan kategori yang sama atau yang berbeda.

Setiap pemesanan dengan banyak kategori akan dikerjakan terpisah sesuai dengan kategori. Perhitungan estimasi akan dilakukan per kategori yang akan melihat kondisi dari proses produksi yang sedang dilakukan untuk kategori tersebut. Tanggal penyelesaian ditentukan dari kategori dengan waktu pengerjaan yang paling lama.

Berikut adalah contoh perhitungan estimasi untuk pemesanan dengan banyak kategori. List pemesanan yang masuk dapat dilihat pada tabel 3.6.

Tabel 3.6 List Pemesanan dengan Banyak Kategori

Pemesanan	Tgl Pemesanan	Kategori	Qty	Jml Pesanan
V	2 Juni 2011	T-Shirt	100	220
		Jacket	50	
		Polo Shirt	70	
VI	2 Juni 2011	T-Shirt	40	150
		Jacket	40	
		Polo Shirt	40	
		Sweater	30	
VII	2 Juni 2011	Jacket	50	110
		Sweater	60	

Jumlah maksimum produksi perhari untuk setiap kategori dapat dilihat pada tabel 3.7.

Tabel 3.7 Daftar Maksimum Produksi per Kategori

Kategori	Jml Maks per hari [a]	Jml Maks per pesanan/order [b]	Jml Pesanan/order per hari [a/b]
T-Shirt	30	10	3
Polo Shirt	45	15	3
Jacket	40	20	2
Sweater	30	10	3

Langkah-langkah perhitungan estimasi untuk pemesanan pada tabel 3.6 adalah sebagai berikut :

1. **Pemesanan V**

   Tanggal pemesanan : **2 Juni 2011**

   Kategori : - **T-Shirt (100 potong)**

- **Jacket (50 potong)**
- **Polo Shirt (70 potong)**

Jumlah Pesanan : **220 potong**

a. Kategori T-Shirt

Kondisi proses produksi untuk kategori T-Shirt dapat dilihat pada tabel 3.8.

Tabel 3.8 Kondisi Produksi T-Shirt Sebelum Ditambah Pemesanan V

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
IV		10	10	10														
Jumlah		**10**	**10**	**10**														

Jika ditambahkan dengan pemesanan V sebanyak **100 potong** maka proses produksinya akan menjadi seperti pada tabel 3.9.

Tabel 3.9 Kondisi Produksi T-Shirt Setelah Ditambah Pemesanan V

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
IV		10	10	10														
V			10	10	10	10	10	10	10	10	10	10	10					
Jumlah		**10**	**20**	**20**	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**					

Sehingga didapatkan tanggal selesai untuk pemesanan V kategori T-Shirt adalah **13 Juni 2011.**

b. Kategori Jacket

Kondisi proses produksi untuk kategori Jacket adalah kosong atau tidak ada proses produksi. Jika ditambahkan dengan pemesanan V sebanyak 50 potong maka proses produksinya akan menjadi seperti pada tabel 3.10.

Tabel 3.10 Kondisi Produksi Jacket Setelah Ditambah Pemesanan V

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
V			20	20	10													
Jumlah			**20**	**20**	**10**													

Sehingga didapatkan tanggal selesai untuk pemesanan V kategori Jacket adalah **5 Juni 2011.**

c. Kategori Polo Shirt

Kondisi proses produksi untuk kategori Polo Shirt adalah kosong atau tidak ada proses produksi. Jika ditambahkan dengan pemesanan V sebanyak 70 potong maka proses produksinya akan menjadi seperti pada tabel 3.11.

Tabel 3.11 Kondisi Produksi Polo Shirt Setelah Ditambah Pemesanan V

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
V			15	15	15	15	15	15	10									
Jumlah			**15**	**15**	**15**	**15**	**15**	**15**	**10**									

Sehingga didapatkan tanggal selesai untuk pemesanan V kategori Polo Shirt adalah **9 Juni 2011.**

Berdasarkan hasil perhitungan yang telah dilakukan maka didapatkan tanggal penyelesaian untuk pemesanan V adalah **13 Juni 2011** berdasarkan waktu pengerjaan yang paling lama yaitu kategori T-Shirt.

2. Pemesanan VI

Tanggal pemesanan : **2 Juni 2011**

Kategori : - **T-Shirt (40 potong)**

- **Jacket (40 potong)**
- **Polo Shirt (40 potong)**
- **Sweater (30 potong)**

Jumlah Pesanan : **150 potong**

a. Kategori T-Shirt

Kondisi proses produksi untuk kategori T-Shirt dapat dilihat pada tabel 3.12.

Tabel 3.12 Kondisi Produksi T-Shirt Sebelum Ditambah Pemesanan VI

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
IV		10	10	10														
V			10	10	10	10	10	10	10	10	10	10	10					
Jumlah		**10**	**20**	**20**	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**					

Jika ditambahkan dengan pemesanan VI sebanyak **40 potong** maka proses produksinya akan menjadi seperti pada tabel 3.13.

Tabel 3.13 Kondisi Produksi T-Shirt Setelah Ditambah Pemesanan VI

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
IV		10	10	10														
V			10	10	10	10	10	10	10	10	10	10	10					
VI			10	10	10	10												
Jumlah		**10**	**30**	**30**	**20**	**20**	**10**	**10**	**10**	**10**	**10**	**10**	**10**					

Sehingga didapatkan tanggal selesai untuk pemesanan VI kategori T-Shirt adalah **6 Juni 2011.**

b. Kategori Jacket

Kondisi proses produksi untuk kategori Jacket dapat dilihat pada tabel 3.14.

Tabel 3.14 Kondisi Produksi Jacket Sebelum Ditambah Pemesanan VI

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
V			20	20	10													
Jumlah			**20**	**20**	**10**													

Jika ditambahkan dengan pemesanan VI sebanyak **40 potong** maka proses produksinya akan menjadi seperti pada tabel 3.15.

Tabel 3.15 Kondisi Produksi Jacket Setelah Ditambah Pemesanan VI

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
V			20	20	10													
VI			20	20														
Jumlah			**40**	**40**	**10**													

Sehingga didapatkan tanggal selesai untuk pemesanan VI kategori Jacket adalah **4 Juni 2011.**

c. Kategori Polo Shirt

Kondisi proses produksi untuk kategori Polo Shirt dapat dilihat pada tabel 3.16.

Tabel 3.16 Kondisi Produksi Polo Shirt Sebelum Ditambah Pemesanan VI

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
V			15	15	15	15	15	15	10									
Jumlah			**15**	**15**	**15**	**15**	**15**	**15**	**10**									

Jika ditambahkan dengan pemesanan VI sebanyak **40 potong** maka proses produksinya akan menjadi seperti pada tabel 3.17.

Tabel 3.17 Kondisi Produksi Polo Shirt Setelah Ditambah Pemesanan VI

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
V			15	15	15	15	15	15	10									
VI			15	15	10													
Jumlah			**30**	**30**	**25**	**15**	**15**	**15**	**10**									

Sehingga didapatkan tanggal selesai untuk pemesanan VI kategori Polo Shirt adalah **5 Juni 2011.**

d. Kategori Sweater

Kondisi proses produksi untuk kategori Sweater adalah kosong atau tidak ada proses produksi. Jika ditambahkan dengan pemesanan VI sebanyak **30 potong** maka proses produksinya akan menjadi seperti pada tabel 3.18.

Tabel 3.18 Kondisi Produksi Sweater Setelah Ditambah Pemesanan VI

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
VI			10	10	10													
Jumlah			**10**	**10**	**10**													

Sehingga didapatkan tanggal selesai untuk pemesanan V kategori Jacket adalah **5 Juni 2011.**

Berdasarkan hasil perhitungan yang telah dilakukan maka didapatkan tanggal penyelesaian untuk pemesanan VI adalah **6 Juni 2011** berdasarkan waktu pengerjaan yang paling lama yaitu kategori T-Shirt.

3. Pemesanan VII

Tanggal pemesanan : **2 Juni 2011**

Kategori : - **Jacket (50 potong)**

- **Sweater (60 potong)**

Jumlah Pesanan : **110 potong**

a. Kategori Jacket

Kondisi proses produksi untuk kategori Jacket dapat dilihat pada tabel 3.19.

Tabel 3.19 Kondisi Produksi Jacket Sebelum Ditambah Pemesanan VII

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
V			20	20	10													
VI			20	20														
Jumlah			**40**	**40**	**10**													

Jika ditambahkan dengan pemesanan VII sebanyak **50 potong** maka proses produksinya akan menjadi seperti pada tabel 3.20.

Tabel 3.20 Kondisi Produksi Jacket Setelah Ditambah Pemesanan VII

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
V			20	20	10													
VI			20	20														
VII					20	20	10											
Jumlah			**40**	**40**	**30**	**20**	**10**											

Sehingga didapatkan tanggal selesai untuk pemesanan VII kategori Jacket adalah **7 Juni 2011.**

b. Kategori Sweater

Kondisi proses produksi untuk kategori Sweater dapat dilihat pada tabel 3.21.

Tabel 3.21 Kondisi Produksi Sweater Sebelum Ditambah Pemesanan VII

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
VI			10	10	10													
Jumlah			**10**	**10**	**10**													

Jika ditambahkan dengan pemesanan VII sebanyak **60 potong** maka proses produksinya akan menjadi seperti pada tabel 3.22.

Tabel 3.22 Kondisi Produksi Sweater Setelah Ditambah Pemesanan VII

Pemesanan	Tanggal																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
VI			10	10	10													
VII			10	10	10	10	10	10										
Jumlah			**20**	**20**	**20**	**10**	**10**	**10**										

Sehingga didapatkan tanggal selesai untuk pemesanan V kategori Jacket adalah **8 Juni 2011.**

Berdasarkan hasil perhitungan yang telah dilakukan maka didapatkan tanggal penyelesaian untuk pemesanan VII adalah **8 Juni 2011** berdasarkan waktu pengerjaan yang paling lama yaitu kategori Sweater.

3.1.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah analisis yang dilakukan untuk mengetahui kebutuhan apa saja yang dibutuhkan yang meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan sehingga aplikasi yang akan dibangun dapat bekerja secara optimal. Analisis kebutuhan non fungsional mencakup analisis perangkat keras, perangkat lunak dan perangkat pikir (pengguna).

3.1.3.1 Analisis Perangkat Keras

Perangkat keras yang terdapat saat ini pada Pabrik Baju 12Titan yaitu satu unit komputer pada kantor pemilik dengan spesifikasi sebagai berkut:

a. Proccessor : AMD X2 555 3.0 GHz
b. Memory : 2GB
c. Harddisk : 320 GB
d. Monitor : LG 19”
e. Jaringan : Internet
f. Scanner
g. Printer
h. *Keyboard* dan *Mouse*

Perangkat keras yang dibutuhkan untuk menjalankan aplikasi *E-commerce* ini adalah 2 buah komputer dengan spesifikasi minimal sebagai berikut:

1. Komputer Admin diletakan di ruang pemilik dengan spesifikasi sebagai berikut:
   a. Proccessor : Kecepatan 1.8 GHz
   b. Memory : 256 MB
   c. Harddisk : 40 GB (ruang kosong 2 GB)
   d. Monitor : ukuran 14” Resolusi 1024x768 px
   e. Jaringan : LAN dan koneksi internet
   f. *Keyboard* dan *Mouse*
2. Komputer pada bagian pelayanan
   a. Proccessor : Kecepatan 1.8 GHz

b. Memory : 256 MB

c. Harddisk : 40 GB (ruang kosong 2 GB)

d. Monitor : ukuran 14” Resolusi 1024x768 px

e. Jaringan : LAN dan koneksi internet

Perangkat keras yang ada pada saat ini belum memenuhi kebutuhan untuk dapat menjalankan aplikasi yang akan dibangun.

3.1.3.2 Analisis Perangkat Lunak

Perangkat lunak yang digunakan pada saat ini yaitu:

1. Sistem Operasi Windows XP
2. *Web Browser* Google Chrome

Sedangkan perangkat lunak yang dibutuhkan untuk membangun aplikasi ini adalah :

1. WAMP*Server*, yang mencakup *Web Server* (*Apache*), MySQL sebagai *database*, dan PHP.
2. PHP Designer 7.2.4 sebagai aplikasi pembangun.
3. Web Browser seperti Mozilla Firefox atau Google Chrome.

3.1.3.3 Analisis Perangkat Pikir (Pengguna)

Analisis pengguna dilakukan untuk mengetahui tugas dan karakteristik pengguna yang akan menggunakan sistem yang akan dibangun. Adapun perangkat pikir (pengguna) yang dimiliki saat ini adalah Pemilik yang memiliki rincian sebagai berikut :

1. **Direktur Utama / Pemilik**

   Umur : 24 Tahun

   Pendidikan Terakhir : Strata I (S1)

   Kemampuan yang dimiliki : Mampu menggunakan perangkat lunak *Microsoft Office* dan aplikasi pengolahan data dan grafis.

2. **Bagian Pelayanan**

   Umur : 23 Tahum

   Pendidikan Terakhir : Strata I (S1)

   Kemampuan yang dimiliki : Mampu menggunakan *Microsoft Office*.

Pengguna yang akan menggunakan aplikasi yang akan dibangun ini terdiri dari administrator, operator, petugas gudang, *member* dan pengunjung, jenis pengguna akan dijelaskan sebagai berikut :

1. Admin

   Admin adalah pengguna aplikasi yang memiliki hak akses lebih dibandingkan pengguna lain, dengan karakteristik sebagai berikut :

   a. Mempunyai kemampuan dasar di bidang komputer.
   b. Memiliki pengalaman lebih lama dalam menggunakan komputer.
   c. Memiliki latar belakang pendidikan minimal SMA / Sederajat.
   d. Memiliki pengalaman menggunakan aplikasi berbasis web / internet.

2. Operator

   Operator adalah pengguna aplikasi yang memiliki hak akses untuk mengolah data produk dan data pesanan. Operator memiliki karakteristik sebagai berikut :

   a. Memiliki kemampuan dasar dalam menggunakan komputer.
   b. Memiliki latar belakang pendidikan minimal SMA / Sederajat.
   c. Dapat menggunakan web browser dan pernah menggunakan aplikasi berbasis web / internet.

3. Petugas Gudang

   Petugas gudang adalah pengguna aplikasi yang memiliki hak akses untuk menentukan status pemesanan apakah sedang dalam proses atau telah selesai diproses. Petugas gudang minimal memiliki kemampuan dalam menggunakan komputer dan internet.

4. *Member*

   Member adalah pengguna dapat melakukan pemesanan dan pembelian, memiliki akun khusus sehingga dapat melihat *history* pemesanan yang sudah diakukan dan tidak perlu lagi memasukan data jika melakukan pemesanan. *Member* memiliki karakteristik sebagai berikut :

   a. Mempunyai kemampuan dasar di bidang komputer.
   b. Memiliki pengalaman menggunakan internet.

5. Pengunjung

   Pengunjung adalah pengguna yang hanya melakukan aktivitas melihat produk pada situs, dapat memesan barang dan menjadi *member* dengan karakteristik memiliki pengalaman menggunakan komputer dan internet.

Berdasarkan perbandingan antara pengguna yang ada dan yang akan menggunakan aplikasi, dapat disimpulkan bahwa pemilik akan menjadi berstatus sebagai admin yang memiliki hak akses penuh terhadap sistem, sedangkan bagian pelayanan akan menjadi operator yang hanya memiliki hak akses untuk mengolah data produk dan pesanan.

3.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk menganalisis proses yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan proses bisnis pada Pabrik Baju 12Titan.

Analisis fungsional dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahapan pemodelan dalam analisis tersebut antara lain mengidentifikasi aktor, pembuatan *use case diagram*, *use case scenario, sequence diagram*, *class diagram, package diagram, dan deployment diagram*.

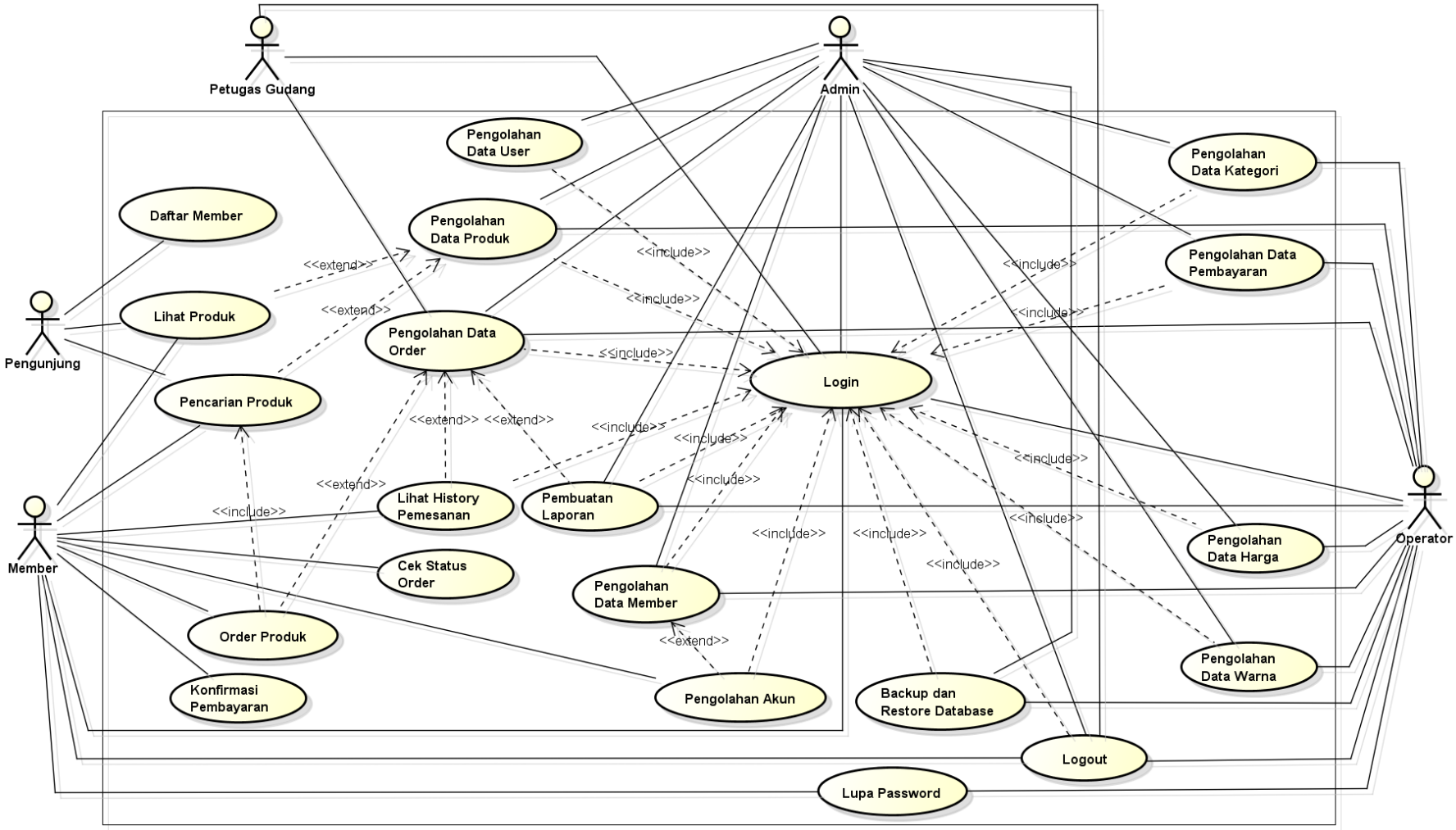
3.1.4.1 Identifikasi Aktor pada Pabrik Baju 12Titan

Aktor yang diidentifikasi dalam *E-commerce* yang akan dibangun yaitu admin, operator, petugas gudang, *member* dan pengunjung.

3.1.4.2 *Use* Case Diagram

Use case adalah interaksi atau dialog antara sistem dan aktor, termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem. *Use case* diagram menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar.

Gambar 3.5 adalah *use case* diagram untuk *E-commerce* pada Pabrik Baju12Titan.



Gambar 3.5 *Use Case* Diagram *E-commerce* Pabrik Baju 12Titan

3.1.4.3 Use Case Skenario

Use Case Skenario menjelaskan interaksi antara aktor dan use case dalam melakukan sebuah proses tertemtu. Berikut adalah use case skenario berdasarkan use case pada *E-commerce* Pabrik Baju 12Titan.

1. Use Case Skenario Login

Interaksi antara aktor admin, operator dan member dengan use case login dapat dilihat pada tabel 3.23.

Tabel 3.23 Use Case Skenario Login

Identifikasi			
Nama	Login		
Tujuan	Masuk ke dalam sistem		
Deskripsi	Proses Login merupakan proses autentikasi untuk menggunakan hak akses sebagai administrator atau operator dalam menggunakan sistem.		
Aktor	Admin, Operator, Member		
Use Case yang Berkaitan	-		
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**	
		1	Menampilkan pesan bahwa data login salah.
2	Mengisi kembali form login	3	Mengecek data user di *database*.
		4	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 sesuai dengan hak akses		

2. Use Case Skenario Pengolahan Data Kategori

Interaksi antara aktor admin dan operator dengan use case pengolahan data kategori dapat dilihat pada tabel 3.24.

Tabel 3.24 Use Case Skenario Pengolahan Data Ketegori

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 dan cari data.		
Aktor	Admin, Operator		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan data kategori dan menu pengolahan kategori		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu tambah data kategori	2	Menampilkan form tambah data 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 kategori 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 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
Skenario Alternatif (Cari Data Kategori)			
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 Kategori telah berhasil diolah		

3. Use Case Skenario Pengolahan Data Produk

Interaksi antara aktor admin dan operator dengan use case pengolahan data produk dapat dilihat pada tabel 3.25.

Tabel 3.25 Use Case Skenario Pengolahan Data Produk

Identifikasi			
Nama	Pengolahan Data Produk		
Tujuan	Mengelola data produk pada sistem		
Deskripsi	Proses pengolahan data produk merupakan proses untuk mengatur dan mengolah data produk pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data dan cari data.		
Aktor	Admin, Operator		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan data produk dan menu pengolahan produk		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu tambah data produk	2	Menampilkan form tambah data produk
3	Mengisi form tambah data produk dan menekan tombol simpan	4	Memvalidasi data masukan, jika data valid maka data disimpan pada *database* dan mengarahkan ke halaman produk, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data produk
Skenario Alternatif (Ubah Data Produk)			
5	Memilih menu ubah data pada data produk 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 produk, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data produk
Skenario Alternatif (Hapus Data Produk)			
9	Memilih data produk yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data produk pada *database*
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman produk

Tabel 3.25 Use Case Skenario Pengolahan Data Produk (lanjutan)

Skenario Alternatif (Cari Data Produk)			
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 Produk telah berhasil diolah		

4. Use Case Skenario Pengolahan Data Order

Interaksi antara aktor admin dan operator dengan use case pengolahan data order dapat dilihat pada tabel 3.26.

Tabel 3.26 Use Case Skenario Pengolahan Data Order

Identifikasi			
Nama	Pengolahan Data Order		
Tujuan	Mengelola data order 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, Operator		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan data order dan menu pengolahan data order		
Aksi Aktor		**Reaksi Sistem**	
1	Memasukan data cari dan memilih tombol cari	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)			
5	Memilih menu ubah data pada data order 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 order, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data order
Skenario Alternatif (Hapus Data Order)			
9	Memilih data order yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan	12	Menghapus data order pada *database*

Tabel 3.26 Use Case Skenario Pengolahan Data Order (lanjutan)

	konfirmasi		
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman order
Kondisi Akhir	Data Order telah berhasil diolah		

5. Use Case Skenario Pengolahan Data Member

Interaksi antara aktor admin dan operator dengan use case pengolahan data member dapat dilihat pada tabel 3.27.

Tabel 3.27 Use Case Skenario Pengolahan Data Member

Identifikasi			
Nama	Pengolahan Data 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.		
Aktor	Admin, Operator		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan data member		
Aksi Aktor		**Reaksi Sistem**	
1	Memasukan data cari dan memilih tombol cari	2	Mencari data pada *database*, jika data ditemukan maka data akan ditampilkan, jika tidak ditemukan maka akan menampilkan pesan data tidak ditemukan.
Skenario Alternatif (Hapus Data Member)			
3	Memilih data member yang akan dihapus	4	Menampilkan pesan konfirmasi apakah data akan dihapus
5	Menekan tombol ya pada pesan konfirmasi	6	Menghapus data member pada *database*
7	Menekan tombol tidak pada pesan konfirmasi	8	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman member
Kondisi Akhir	Data Member telah berhasil diolah		

6. Use Case Skenario Pengolahan Data Pembayaran

Interaksi antara aktor admin dan operator dengan use case pengolahan data pembayaran dapat dilihat pada tabel 3.28.

Tabel 3.28 Use Case Skenario Pengolahan Data Pembayaran

Identifikasi			
Nama	Pengolahan Data Pembayaran		
Tujuan	Mengelola data pembayaran pada sistem		
Deskripsi	Proses pengolahan data pembayaran merupakan proses untuk mengatur dan mengolah data pembayaran pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data dan cari data pembayaran.		
Aktor	Admin, Operator		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan data pembayaran dan menu pengolahan data pembayaran		
Aksi Aktor		**Reaksi Sistem**	
1	Memasukan data cari dan memilih tombol cari	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 Pembayaran)			
5	Memilih menu ubah data pada data pembayaran 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 pembayaran, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data pembayaran
Skenario Alternatif (Hapus Data Pembayaran)			
9	Memilih data pembayaran yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data pembayaran pada *database*
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman pembayaran
Kondisi Akhir	Data Pembayaran berhasil diolah		

7. Use Case Skenario Pengolahan Data User

Interaksi antara aktor admin dengan use case pengolahan data user dapat dilihat pada tabel 3.29.

Tabel 3.29 Use Case Skenario Pengolahan Data User

Identifikasi			
Nama	Pengolahan Data user		
Tujuan	Mengelola Data user pada sistem		
Deskripsi	Proses pengolahan data user merupakan proses untuk mengatur dan mengolah data user untuk menentukan level login dan data user pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data dan cari data user.		
Aktor	Admin		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan data user dan menu pengolahan data user		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu tambah data user	2	Menampilkan form tambah data user
3	Mengisi form tambah data user dan menekan tombol simpan	4	Memvalidasi data masukan, jika data valid maka data disimpan pada *database* dan mengarahkan ke halaman pengiriman, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data user
Skenario Alternatif (Ubah Data user)			
5	Memilih menu ubah data pada data user 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 pengiriman, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data user
Skenario Alternatif (Hapus Data user)			
9	Memilih data user yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data user pada *database*
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman pengiriman
Skenario Alternatif (Cari Data user)			
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 user berhasil diolah		

8. Use Case Skenario Pembuatan Laporan

Interaksi antara aktor admin dan operator dengan use case pembuatan laporan dapat dilihat pada tabel 3.30.

Tabel 3.30 Use Case Skenario 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 produk, order, dan pembayaran.		
Aktor	Admin, Operator		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan menu pilihan laporan		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih data yang akan dijadikan 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		

9. Use Case Skenario Logout

Interaksi antara aktor admin, operator, dan member dengan use case logout dapat dilihat pada tabel 3.31.

Tabel 3.31 Use Case Skenario Logout

Identifikasi			
Nama	Logout		
Tujuan	Keluar dari sistem		
Deskripsi	Merupakan proses untuk berhenti menggunakan dan berhenti dari sesi login		
Aktor	Admin, Operator, Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan menu utama sesuai hak akses		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu logout	2	Menghentikan akses dan mengarahkan pada form login
Kondisi Akhir	berhasil logout dari sistem		

10. Use Case Skenario Lihat Produk

Interaksi antara aktor pengunjung dan member dengan use case lihat produk dapat dilihat pada tabel 3.32.

Tabel 3.32 Use Case Skenario Lihat Produk

Identifikasi			
Nama	Lihat Produk		
Tujuan	Melihat info produk		
Deskripsi	Merupakan proses untuk melihat list dan info produk yang tersedia		
Aktor	Pengunjung, Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan halaman awal (*home*)		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih tampil produk	2	Mengambil data produk dari *database* menampilkannya pada halaman produk
Kondisi Akhir	Halaman Produk berhasil ditampilkan		

11. Use Case Skenario Pencarian Produk

Interaksi antara aktor pengunjung dan member dengan use case pencarian produk dapat dilihat pada tabel 3.33.

Tabel 3.33 Use Case Skenario Pencarian Produk

Identifikasi			
Nama	Pencarian Produk		
Tujuan	Mencari dan Melihat info produk		
Deskripsi	Proses untuk mencari data produk dan menampilkannya		
Aktor	Pengunjung, Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan halaman awal (*home*)		
Aksi Aktor		**Reaksi Sistem**	
1	Mengisi data cari produk berdasarkan kode/nama/kategori produk.	2	Mencari data produk pada *database*, jika data ditemukan maka akan menampilkan list data produk, jika data tidak ditemukan maka akan menampilkan pesan data tidak ditemukan
Kondisi Akhir	Menampilkan hasil pencarian produk		

12. Use Case Skenario Order Produk

Interaksi antara aktor member dengan use case order produk dapat dilihat pada tabel 3.34.

Tabel 3.34 Use Case Skenario Order Produk

Identifikasi			
Nama	Order Produk		
Tujuan	Melakukan pemesanan produk		
Deskripsi	Proses untuk memilih produk yang akan dipesan		
Aktor	Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan halaman awal (beranda)		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu order	2	Menampilkan katalog produk
3	Memilih kategori dan produk yang akan dipesan	4	Mengecek apakah member sudah login, jika sudah login maka akan menampilkan form order, jika belum login maka akan menampilkan pesan harus login dan menampilkan form login
5	Mengisi form order		
6	Mengisi desain dari produk		

Tabel 3.34 Use Case Skenario Order Produk (lanjutan)

7	Memilih ukuran dan mengisi jumlah ukuran yang aka dipesan	8	Menghitung harga berdasarkan kategori produk dan jumlah ukuran yang dipesan. Jika jumlah pesanan kurang dari 21 atau lebih dari 300 potong maka akan menampilkan pesan bahwa pesanan minimal 21 dan maksimal 300 potong.
9	Menekan tombol tambah ke keranjang	10	Mengecek pada *database* jumlah order yang sedang diproses pada hari ini. Jika order yang sedang diproses kurang dari jumlah kapasitas maksimum produksi maka tanggal penyeselesaian sesuai dengan jumlah pesanan. Jika jumlah order yang sedang diproses melebihi jumlah kapasitas maksimum produksi maka akan dicari pada hari berikutnya dimana jumlah order kurang dari kapasitas maksimum, sehingga tanggal penyelesaiannya dihitung berdasarkan jumlah pesanan ditambah dengan selisih hari ini dengan hari dimana order kurang dari kapasitas maksimum produksi
		11	Menyimpan data order pada keranjang
		12	Menampilkan daftar pesanan dari keranjang
13	Menekan tombol *checkout*	14	Mengambil alamat dari data member dan menampilkan sebagai alamat pengiriman
15	Menekan tombol ‘Lanjutkan’	16	Menampilkan rangkuman order yang telah dipesan , ongkos kirim sesuai alamat pengiriman, total yang harus dibayar dan alamat pengiriman.
17	Menekan tombol ‘Konfirmasi Order’	18	Menyimpan data order pada *database*
		19	Mengirimkan e-mail konfirmasi order kepada member
Kondisi Akhir	Pemesan berhasil melakukan order produk		

13. Use Case Skenario Daftar Member

Interaksi antara aktor pengunjung dengan use case daftar member dapat dilihat pada tabel 3.35.

Tabel 3.35 Use Case Skenario Daftar Member

Identifikasi			
Nama	Daftar Member		
Tujuan	Melakukan pendaftaran member		
Deskripsi	Proses untuk mendaftar/registrasi menjadi member		
Aktor	Pengunjung		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan halaman awal (*home*)		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu daftar member	2	Menampilkan halaman registrasi dan 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 e-mail konfirmasi registrasi berhasil.
Kondisi Akhir	Pengunjung berhasil mendaftar sebagai member		

14. Use Case Skenario Lihat History Pemesanan

Interaksi antara aktor member dengan use case lihat history pemesanan dapat dilihat pada tabel 3.36.

Tabel 3.36 Use Case Skenario Lihat History Pemesanan

Identifikasi			
Nama	Lihat *History* Pemesanan		
Tujuan	Melihat pemesanan yang pernah dilakukan member		
Deskripsi	Proses untuk melihat daftar pemesanan yang sudah dilakukan oleh member		
Aktor	Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan menu utama member		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu history pemesanan	2	Mencari data pesanan pada *database* berdasarkan id member. Jika data ditemukan maka data akan ditampilkan, jika tidak akan tampil pesan data history kosong
Kondisi Akhir	Member dapat melihat history pemesanan		

15. Use Case Skenario Pengolahan Akun

Interaksi antara aktor member dengan use case pengolahan akun dapat dilihat pada tabel 3.37.

Tabel 3.37 Use Case Skenario Pengolahan Akun

Identifikasi			
Nama	Pengolahan Akun		
Tujuan	Mengolah data akun member		
Deskripsi	Proses untuk mengolah data akun member		
Aktor	Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan menu utama member		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu pengolahan akun	2	Mencari data akun berdasarkan id member dan menampilkannya.
3	Memilih menu ubah data akun	4	Menampilkan form ubah data akun
5	Mengisi data ubah akun	6	Memvalidasi data masukan, jika data tidak valid maka akan tampil pesan kesalahan, jika data valid maka data akan disimpan pada *database*
Kondisi Akhir	Member berhasil mengolah data akun		

16. Use Case Skenario Pengolahan Data Warna

Interaksi antara aktor admin dan operator dengan use case pengolahan data warna dapat dilihat pada tabel 3.38.

Tabel 3.38 Use Case Skenario Pengolahan Data Warna

Identifikasi	
Nama	Pengolahan Data Warna
Tujuan	Mengelola data warna pada sistem
Deskripsi	Proses pengolahan data warna merupakan proses untuk mengatur dan mengolah data warna pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data dan cari data warna.
Aktor	Admin, Operator
Use Case yang Berkaitan	-

Tabel 3.38 Use Case Skenario Pengolahan Data Warna (lanjutan)

Skenario Utama			
Kondisi Awal	Menampilkan data warna dan menu pengolahan data warna		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu tambah data warna	2	Menampilkan form tambah data warna
3	Mengisi form tambah data warna dan menekan tombol simpan	4	Memvalidasi data masukan, jika data valid maka data disimpan pada *database* dan mengarahkan ke halaman warna, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data warna
Skenario Alternatif (Ubah Data Warna)			
5	Memilih menu ubah data pada data warna 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 warna, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data warna
Skenario Alternatif (Hapus Data Warna)			
9	Memilih data warna yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data warna pada *database*
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman warna
Skenario Alternatif (Cari Data Warna)			
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 Warna telah berhasil diolah		

17. Use Case Skenario Pengolahan Data Harga

Interaksi antara aktor admin dan operator dengan use case pengolahan data harga dapat dilihat pada tabel 3.39.

Tabel 3.39 Use Case Skenario Pengolahan Data Harga

Identifikasi			
Nama	Pengolahan Data Harga		
Tujuan	Mengelola data harga pada sistem		
Deskripsi	Proses pengolahan data harga merupakan proses untuk mengatur dan mengolah data harga pada sistem yang mencakup tampil data, tambah data, ubah data, hapus data dan cari data harga.		
Aktor	Admin, Operator		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan data harga dan menu pengolahan data harga		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih menu tambah data harga	2	Menampilkan form tambah data harga
3	Mengisi form tambah data harga dan menekan tombol simpan	4	Memvalidasi data masukan, jika data valid maka data disimpan pada *database* dan mengarahkan ke halaman harga, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data harga
Skenario Alternatif (Ubah Data Harga)			
5	Memilih menu ubah data pada data harga 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 harga, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data harga
Skenario Alternatif (Hapus Data Harga)			
9	Memilih data harga yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data harga pada *database*
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke halaman harga
Skenario Alternatif (Cari Data Harga)			
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 Harga telah berhasil diolah		

18. Use Case Skenario Lupa Password

Interaksi antara aktor admin, operator, dan member dengan use case lupa password dapat dilihat pada tabel 3.40.

Tabel 3.40 Use Case Skenario Lupa Password

Identifikasi			
Nama	Lupa Password		
Tujuan	Mereset password aktor		
Deskripsi	Proses untuk mereset password, memberitahu admin dan memberikan password baru melalui email yang didaftarkan.		
Aktor	Admin, Operator, Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Memilih menu lupa password		
Aksi Aktor		**Reaksi Sistem**	
1	Mengisi form username atau email yang sebelumnya didaftarkan	2	Mencari data berdasarkan username atau email yang diisikan, kemudian mereset password dan mengirimkannya ke email yang ada pada *database*.
Kondisi Akhir	Password telah direset dan dikirm ke email.		

19. Use Case Skenario Backup dan Restore *Database*

Interaksi antara aktor admin dan operator dengan use case Backup dan Restore *Database* dapat dilihat pada tabel 3.41.

Tabel 3.41 Use Case Skenario 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, Operator
Use Case yang Berkaitan	-

Tabel 3.38 Case Skenario Backup dan Restore *Database* (lanjutan)

Skenario Utama			
Kondisi Awal	Menampilkan menu backup dan restore *database*		
Aksi Aktor		**Reaksi Sistem**	
1	Menekan tombol Backup *database*	2	Membaca semua data pada *database*
		3	Menyalin pada file berformat SQL
		4	Menyimpan file SQL ke tempat penyimpanan file atau *hard drive*
Skenario Alternatif (Restore *Database*)			
		1	Menampilkan form restore *database*
2	Memilih file berformat SQL yang berisi data dari *database*		
3	Menekan tombol restore	4	Membaca data dari file SQL
		5	Menyalin data dari file SQL ke *database*
Kondisi Akhir	Password telah direset dan dikirm ke email.		

20. Use Case Skenario Cek Status Order

Interaksi antara aktor member dengan use case Cek Status Order dapat dilihat pada tabel 3.42.

Tabel 3.42 Use Case Skenario Cek Status Order

Identifikasi			
Nama	Cek Status Order		
Tujuan	Melihat status order tertentu		
Deskripsi	Proses mengecek data order pada *database* dan menampilkannya kepada member		
Aktor	Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan menu awal (beranda)		
Aksi Aktor		**Reaksi Sistem**	
1	Mengisi form cek status order dengan order ID	2	Mencari data order pada *database* berdasarkan order ID, Jika data ditemukan maka akan ditampilkan.
Skenario Alternatif (Data order tidak ditemukan)			
		1	Menampilkan pesan data order tidak ditemukan
2	Mengisi kembali form cek status order		Mencari data order pada *database* berdasarkan order ID, Jika data ditemukan maka akan ditampilkan.
Kondisi Akhir	Menampilkan status data order		

21. Use Case Skenario Konfirmasi Pembayaran

Interaksi antara aktor member dengan use case Konfirmasi Pembayaran dapat dilihat pada tabel 3.43.

Tabel 3.43 Use Case Skenario Konfirmasi Pembayaran

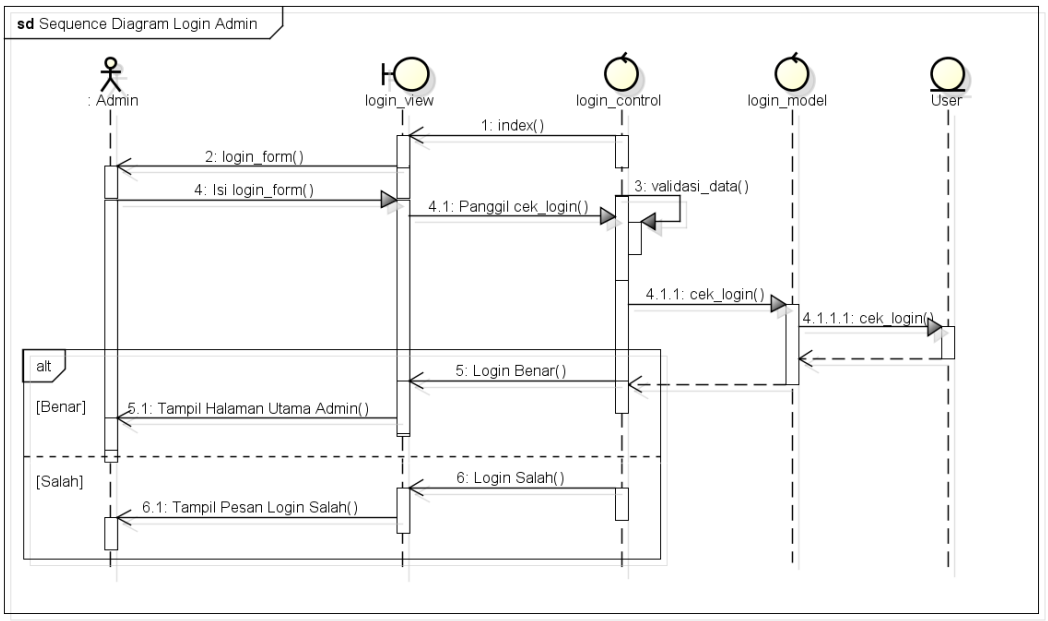
Identifikasi			
Nama	Konfirmasi Pembayaran		
Tujuan	Melakukan konfirmasi pemabayaran untuk sebuah order		
Deskripsi	Proses konfirmasi pembayaran order untuk pembayaran yang dilakukan melalui transfer bank		
Aktor	Member		
Use Case yang Berkaitan	-		
Skenario Utama			
Kondisi Awal	Menampilkan menu awal (beranda)		
Aksi Aktor		**Reaksi Sistem**	
1	Mengisi form Konfirmasi pembayaran dengan order ID	2	Mencari data order pada *database* berdasarkan order ID, Jika data ditemukan maka akan menampilkan form rincian konfirmasi pembayaran.
3	Mengisi form konfirmasi pembayaran meliputi jumlah bayar, bank tujuan, bank asal dan nama pembayar.	4	Melakukan validasi data masukan, jika data valid maka data akan disimpan pada *database*.
		5	Mengirim e-mail konfirmasi pembayaran berhasil
Skenario Alternatif (Data order tidak ditemukan)			
		1	Menampilkan pesan data order tidak ditemukan
2	Mengisi kembali form Konfirmasi Pembayaran	3	Mencari data order pada *database* berdasarkan order ID, Jika data ditemukan maka akan ditampilkan.
Kondisi Akhir	Member berhasil melakukan konfirmasi pembayaran		

3.1.4.4 *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

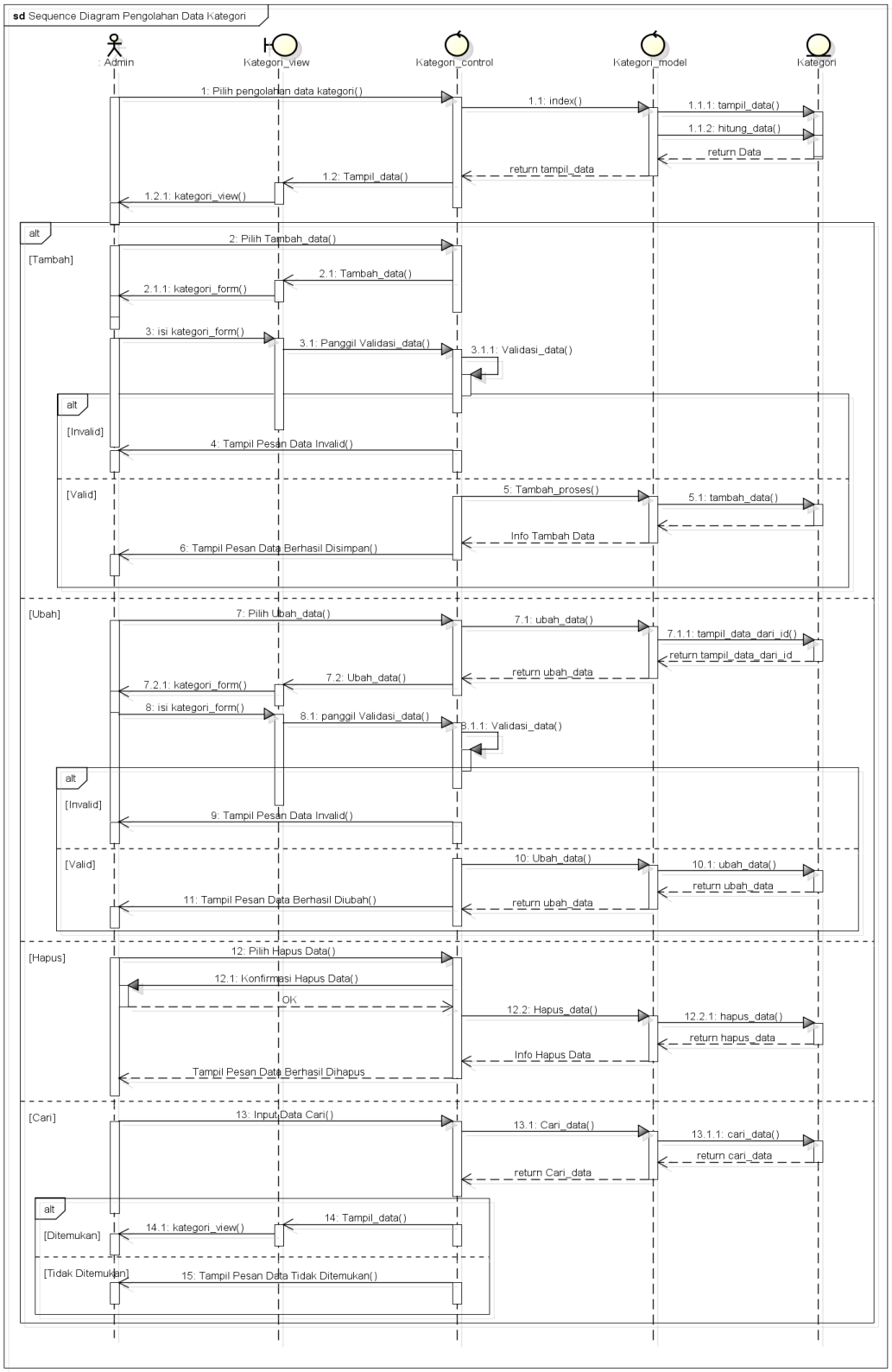
Sequence diagram login admin dapat dilihat pada gambar 3.6.



Gambar 3.6 *Sequence* Diagram Login Admin

2. *Sequence* Diagram Pengolahan Data Kategori

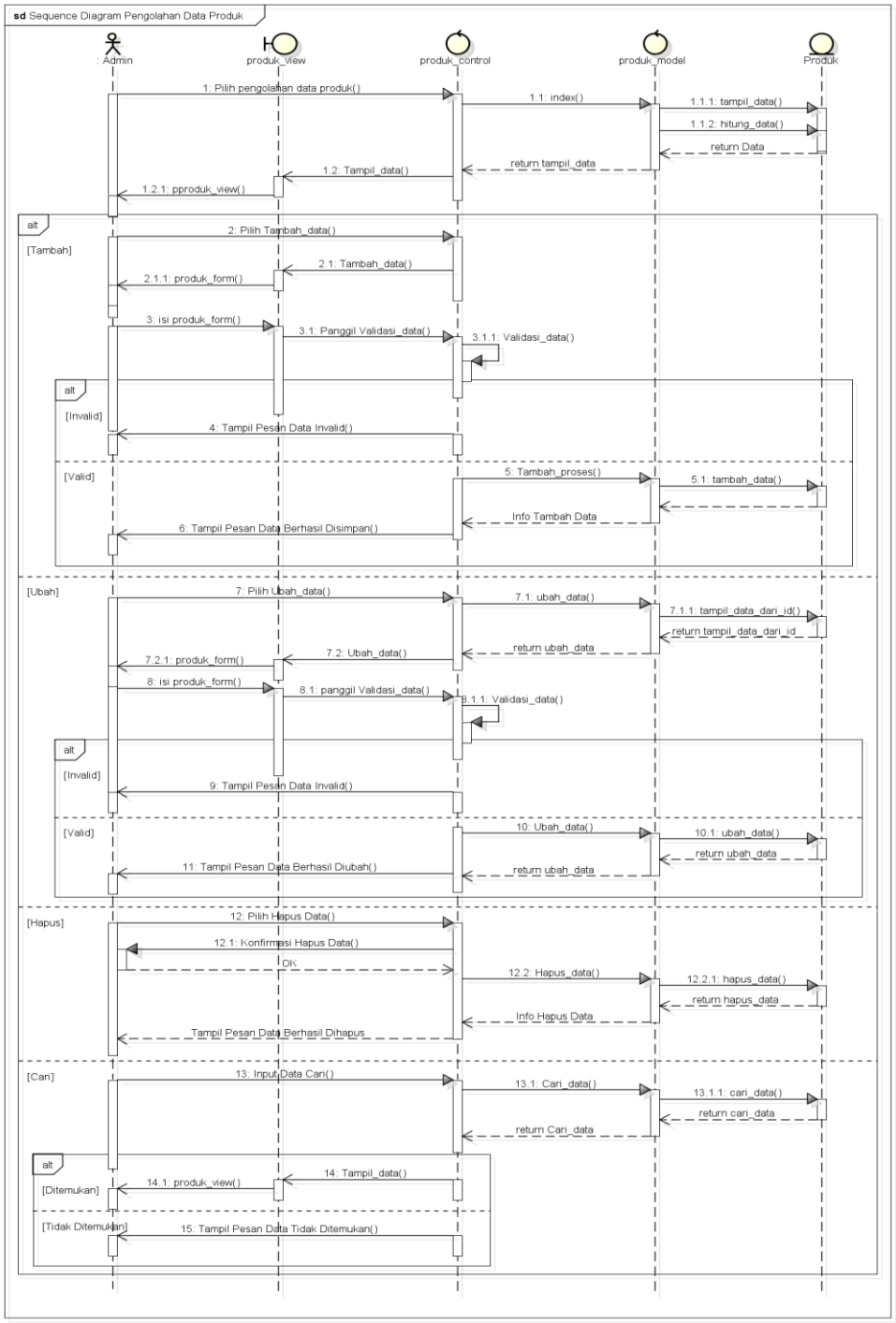
Sequence diagram pengolahan data kategori dapat dilihat pada gambar 3.7.



Gambar 3.7 *Sequence* diagram pengolahan data kategori

3. *Sequence* Diagram Pengolahan Data Produk

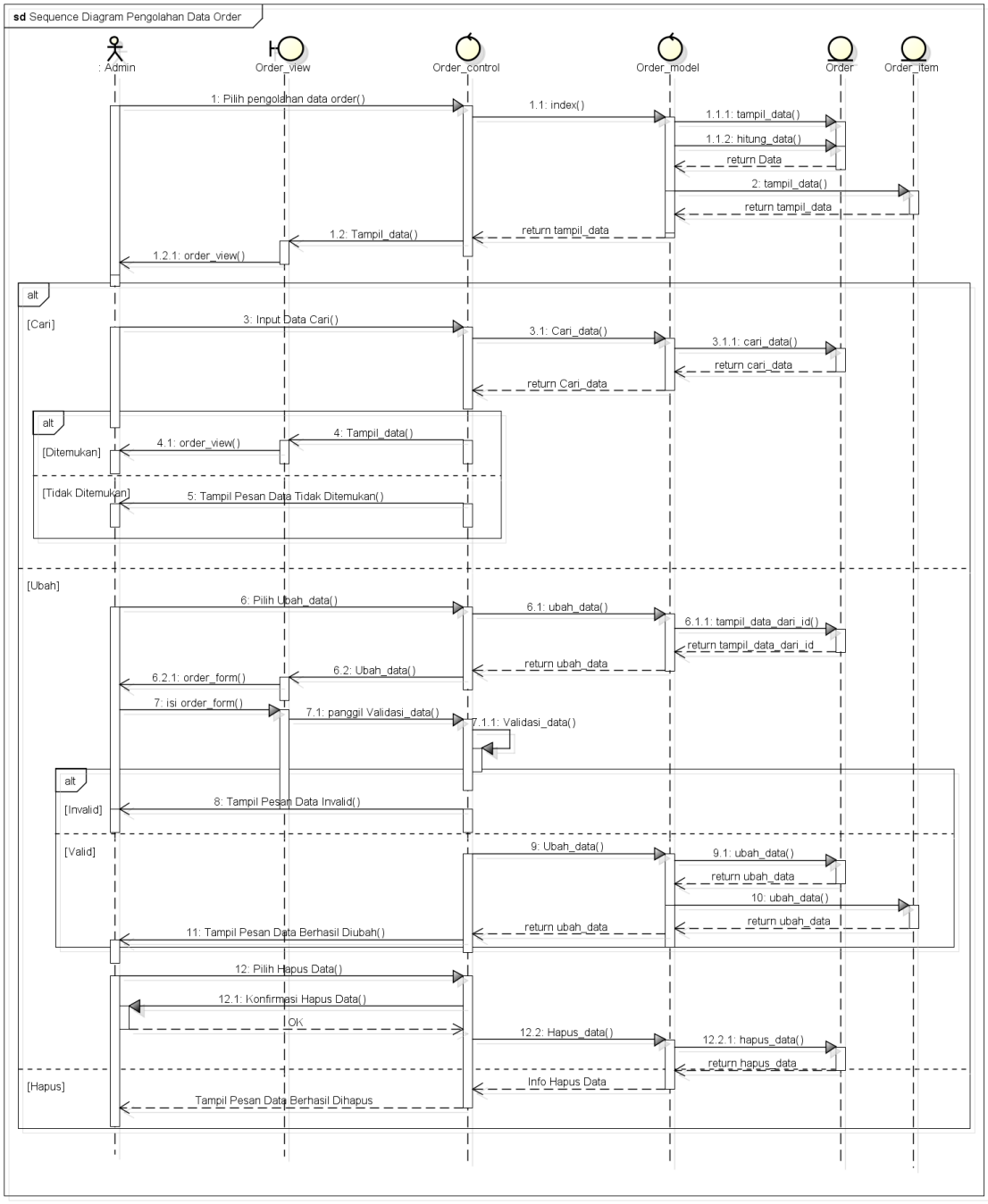
Sequence diagram pengolahan data produk dapat dilihat pada gambar 3.8.



Gambar 3.8 *Sequence* diagram pengolahan data produk

4. *Sequence* Diagram Pengolahan Data Order

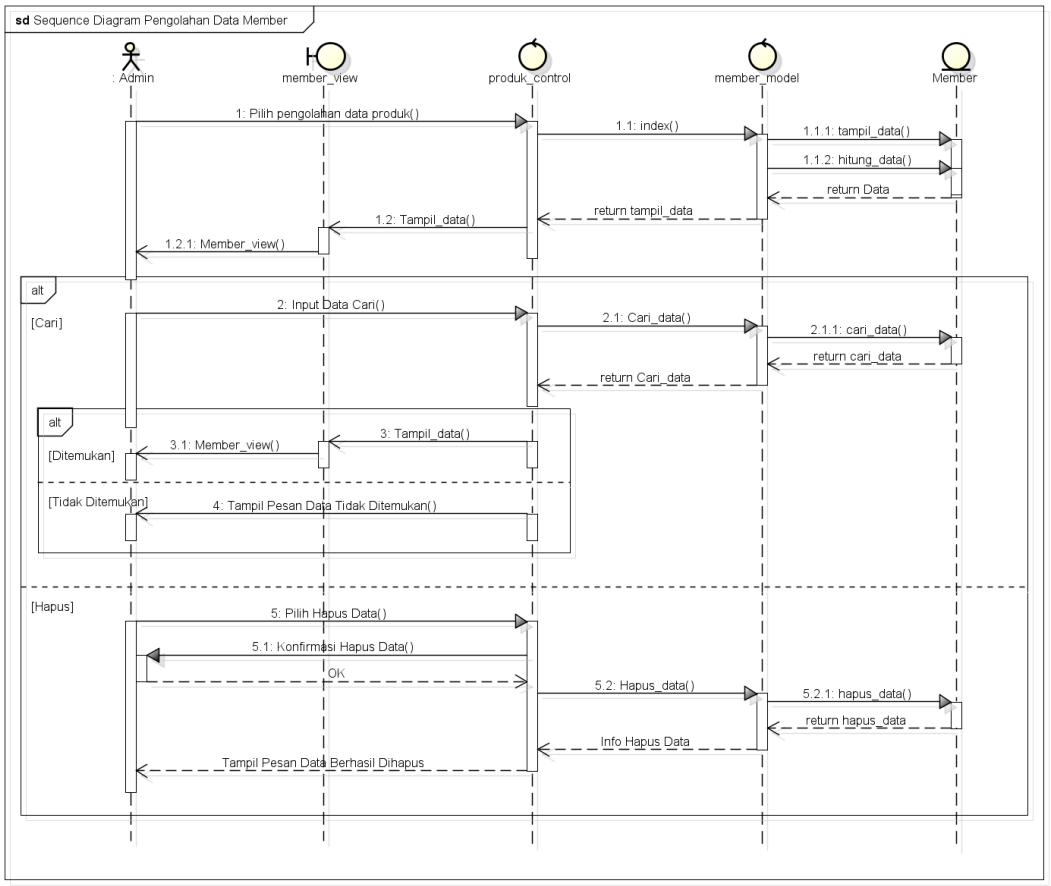
Sequence diagram pengolahan data order dapat dilihat pada gambar 3.9.



Gambar 3.9 *Sequence* Diagram Pengolahan Data Order

5. *Sequence* Diagram Pengolahan Data Member

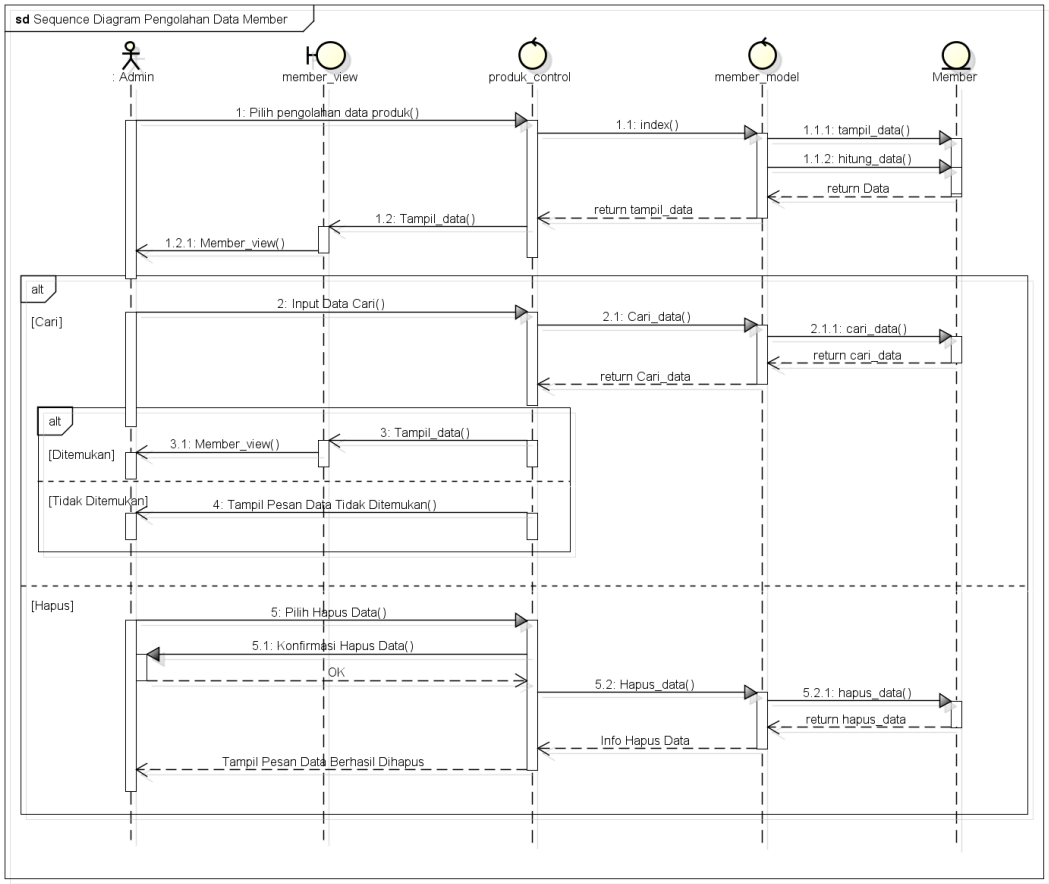
Sequence diagram pengolahan data member dapat dilihat pada gambar 3.10.



Gambar 3.10 *Sequence* Diagram Pengolahan Data Member

6. *Sequence* Diagram Pengolahan Data Pembayaran

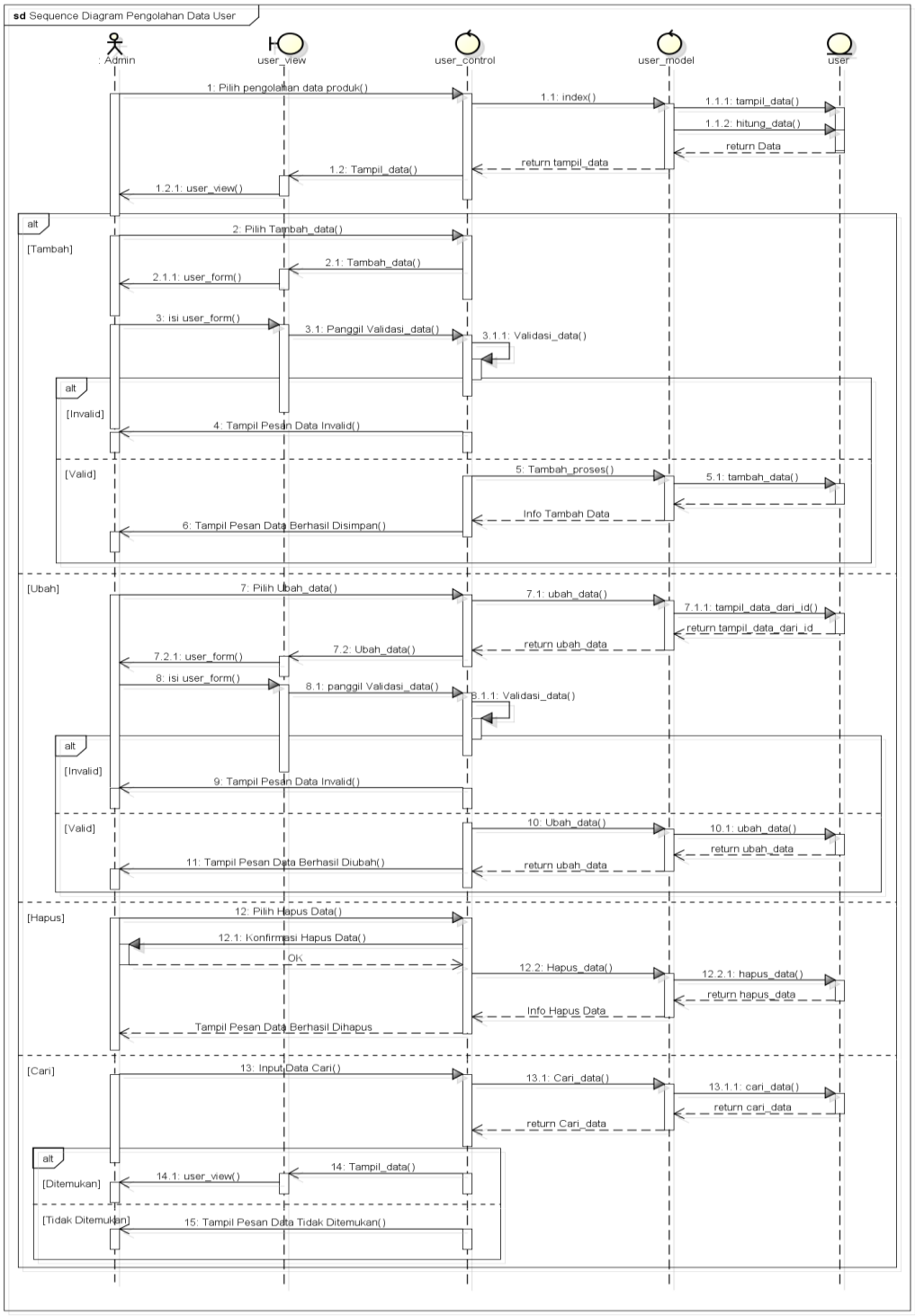
Sequence diagram pengolahan data pembayaran dapat dilihat pada gambar 3.11.



Gambar 3.11 *Sequence* Diagram Pengolahan Data Pembayaran

7. *Sequence* Diagram Pengolahan Data User

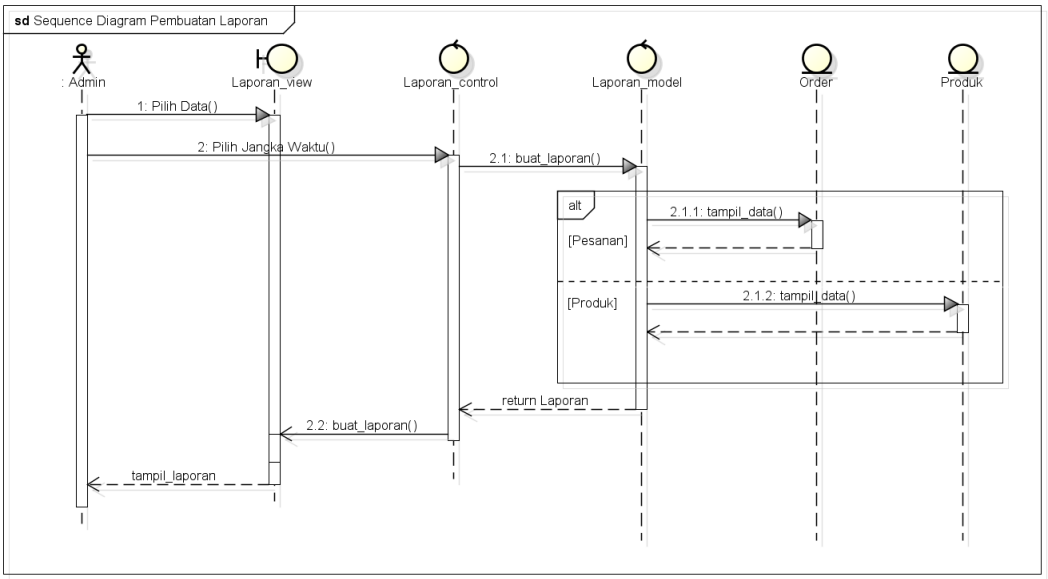
Sequence diagram pengolahan data user dapat dilihat pada gambar 3.12.



Gambar 3.12 *Sequence* Diagram Pengolahan Data User

8. *Sequence* Diagram Pembuatan Laporan

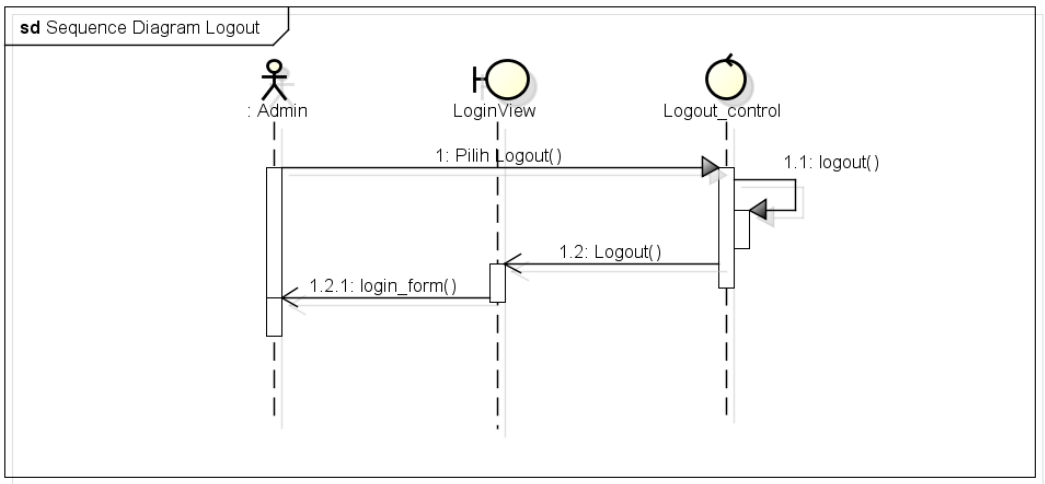
Sequence diagram pembuatan laporan dapat dilihat pada gambar 3.13.



Gambar 3.13 *Sequence* Diagram Pembuatan Laporan

9. *Sequence* Diagram Logout

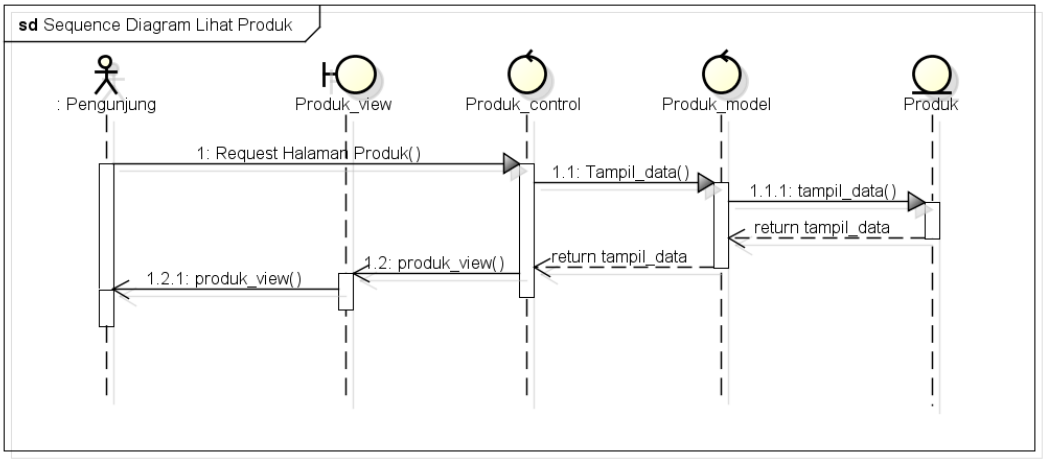
Sequence diagram pengolahan data kategori dapat dilihat pada gambar 3.14.



Gambar 3.14 *Sequence* Diagram Logout Admin

10. *Sequence* Diagram Lihat Produk

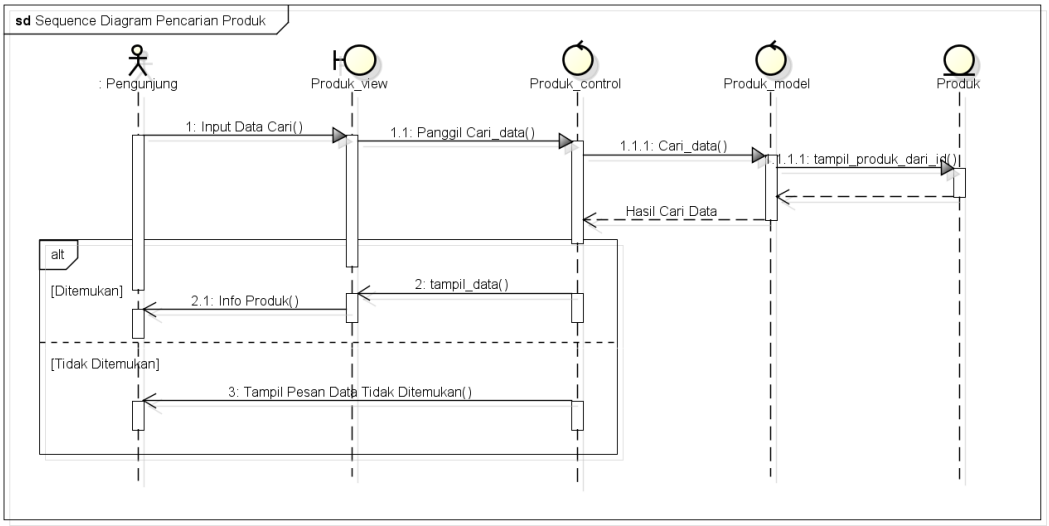
Sequence diagram lihat produk dapat dilihat pada gambar 3.15.



Gambar 3.15 *Sequence* Diagram Lihat Produk

11. *Sequence* Diagram Pencarian Produk

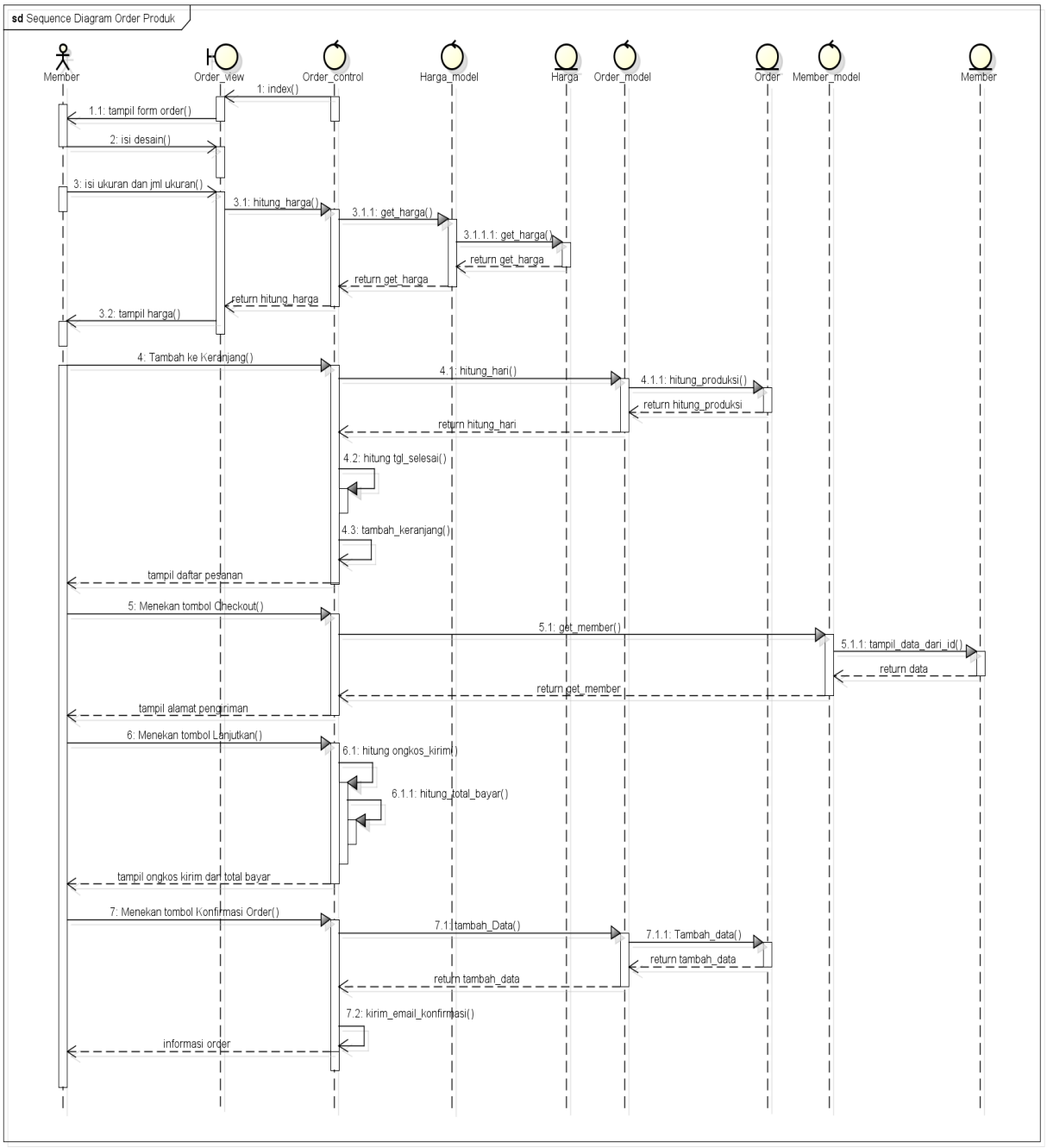
Sequence diagram pencarian produk dapat dilihat pada gambar 3.16.



Gambar 3.16 *Sequence* Diagram Pencarian Produk

12. *Sequence* Diagram Order Produk

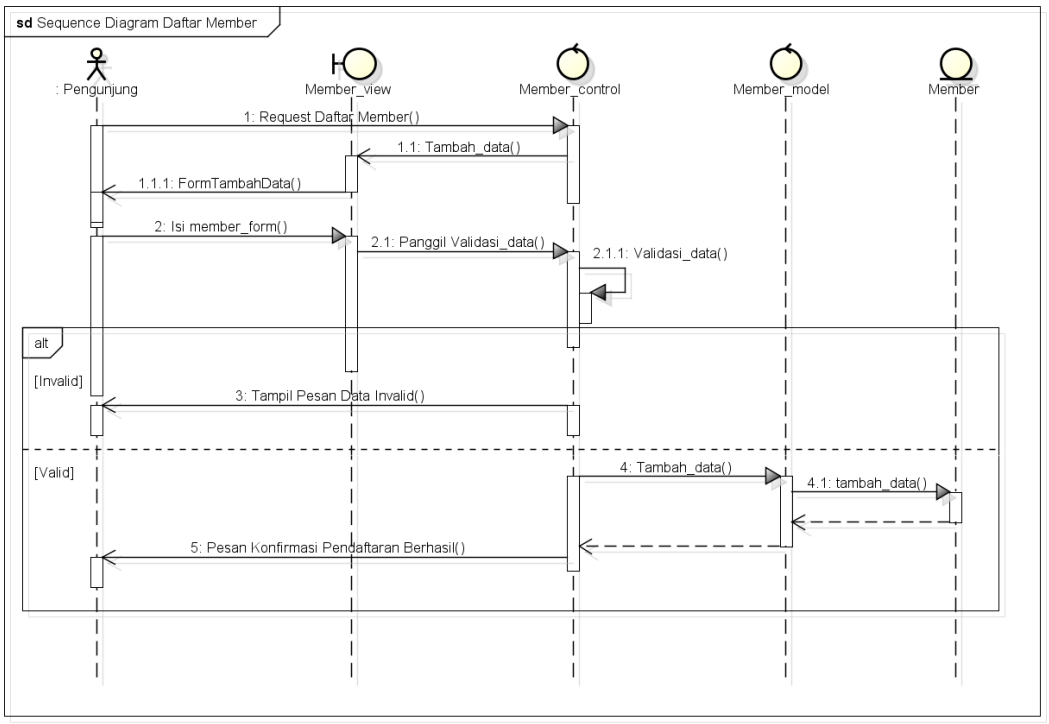
Sequence diagram order produk dapat dilihat pada gambar 3.17.



Gambar 3.17 *Sequence* Diagram Pemesanan Produk

13. *Sequence* Diagram Daftar Member

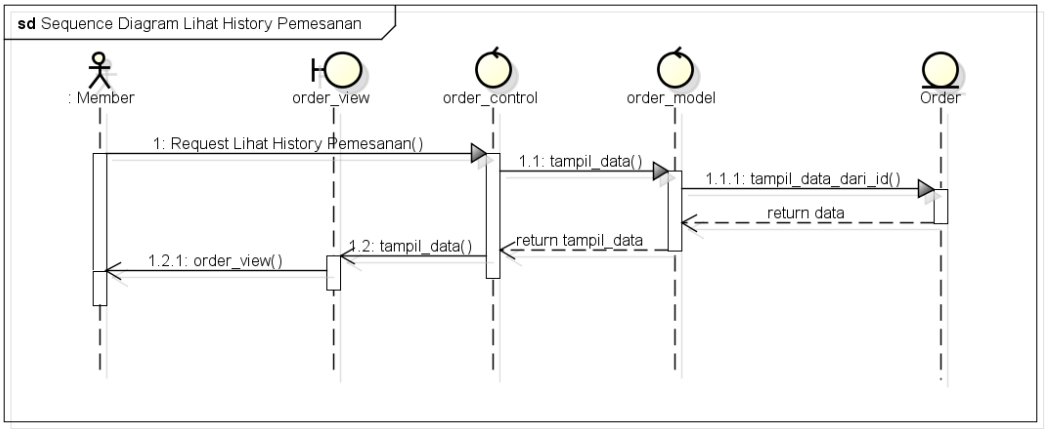
Sequence diagram daftar member dapat dilihat pada gambar 3.18.



Gambar 3.18 *Sequence* Diagram Daftar Member

14. *Sequence* Diagram Lihat History Pemesanan

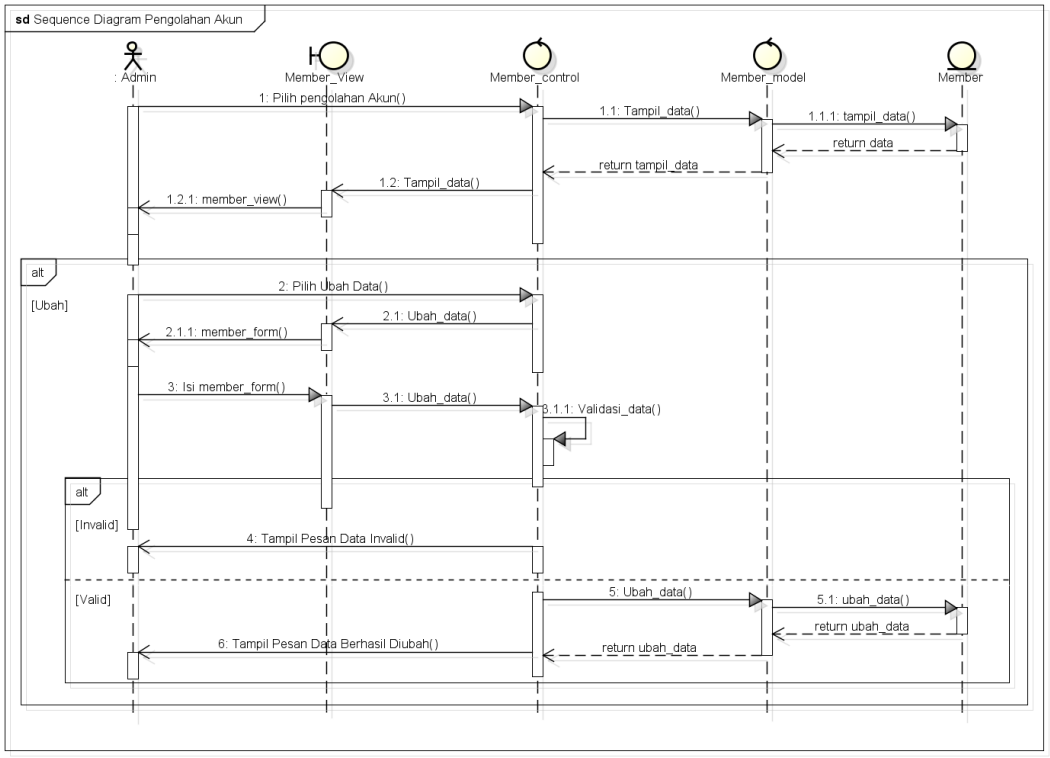
Sequence diagram lihat history pemesanan dapat dilihat pada gambar 3.19.



Gambar 3.19 *Sequence* Diagram Lihat History Pemesanan

15. *Sequence* Diagram Pengolahan Akun

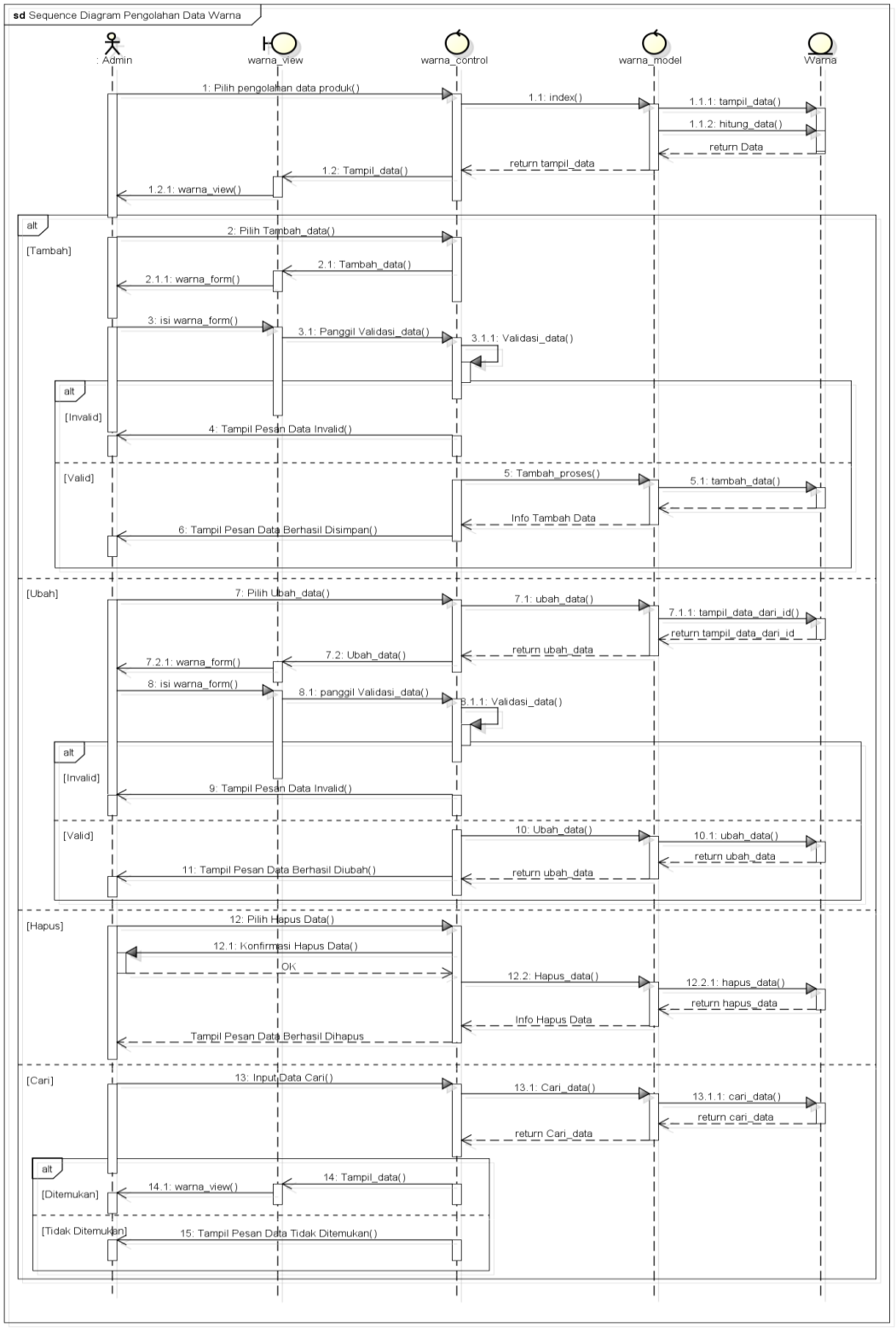
Sequence diagram pengolahan akun dapat dilihat pada gambar 3.20.



Gambar 3.20 *Sequence* Diagram Pengolahan Akun

16. *Sequence* Diagram Pengolahan Data Warna

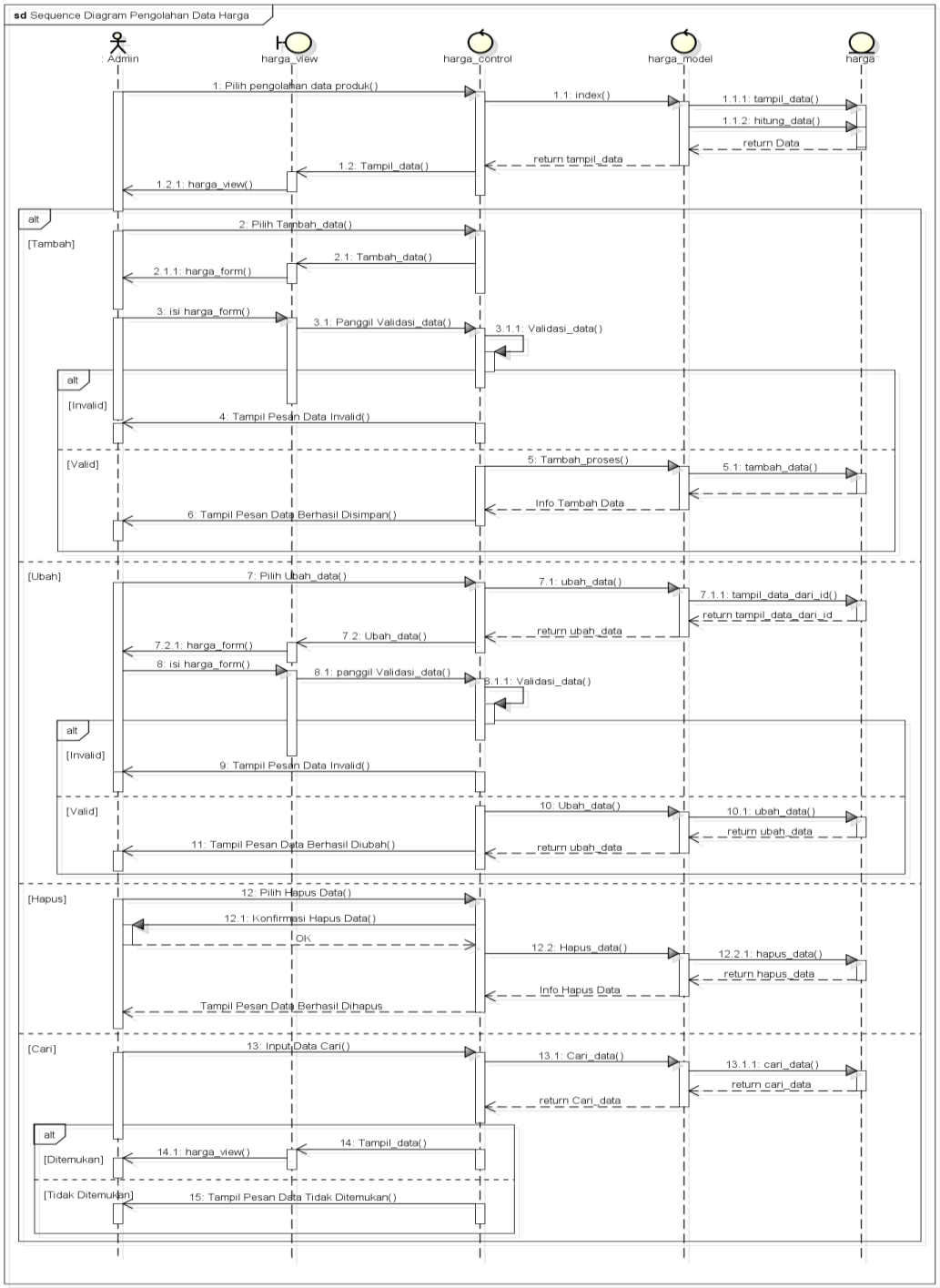
Sequence diagram pengolahan Data Warna dapat dilihat pada gambar 3.21.



Gambar 3.21 *Sequence* Diagram Pengolahan Data Warna

17. *Sequence* Diagram Pengolahan Data Harga

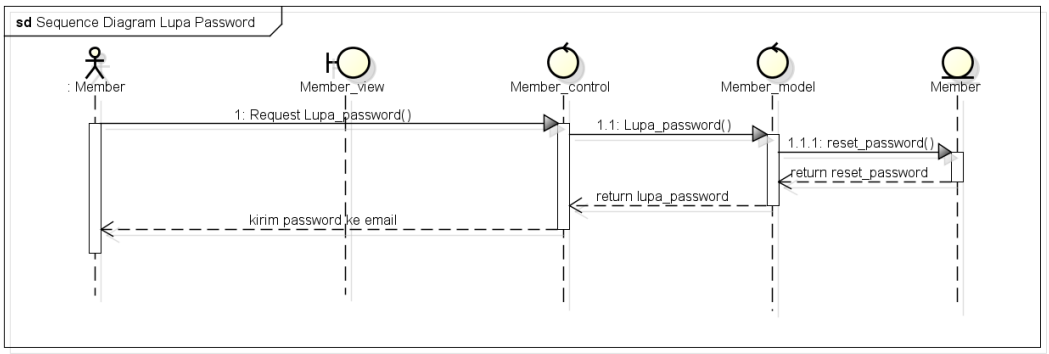
Sequence diagram pengolahan data harga dapat dilihat pada gambar 3.22.



Gambar 3.22 *Sequence* Diagram Pengolahan Data Harga

18. *Sequence* Diagram Lupa Password

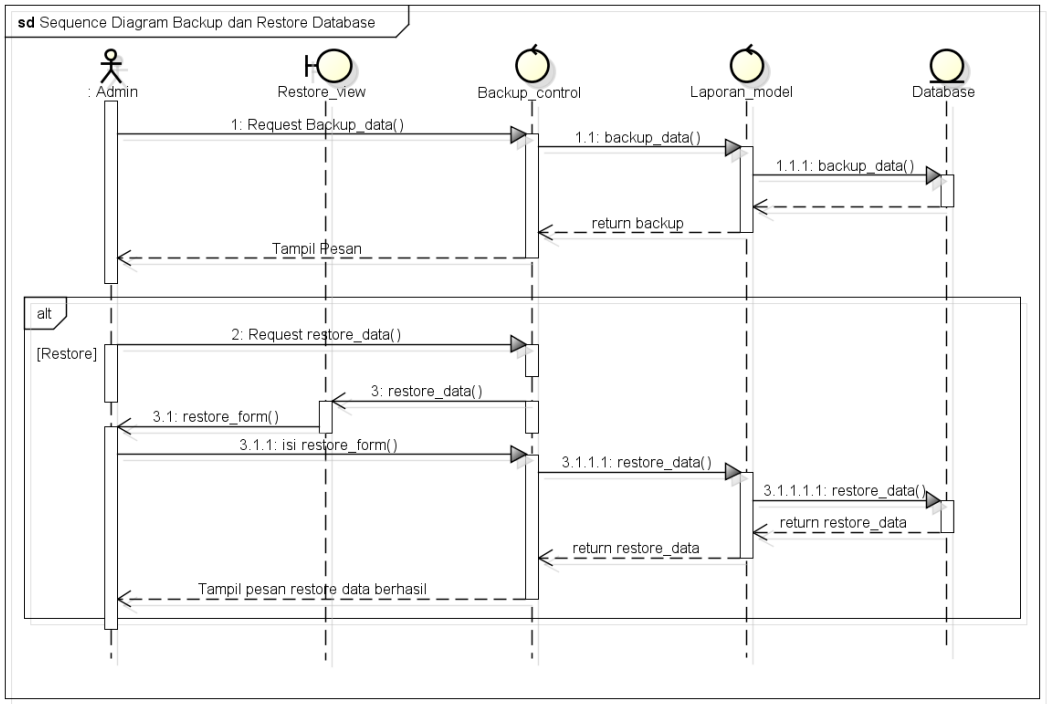
Sequence diagram lupa password dapat dilihat pada gambar 3.23.



Gambar 3.23 *Sequence* Diagram Lupa Password

19. *Sequence* Diagram Backup dan Restore *Database*

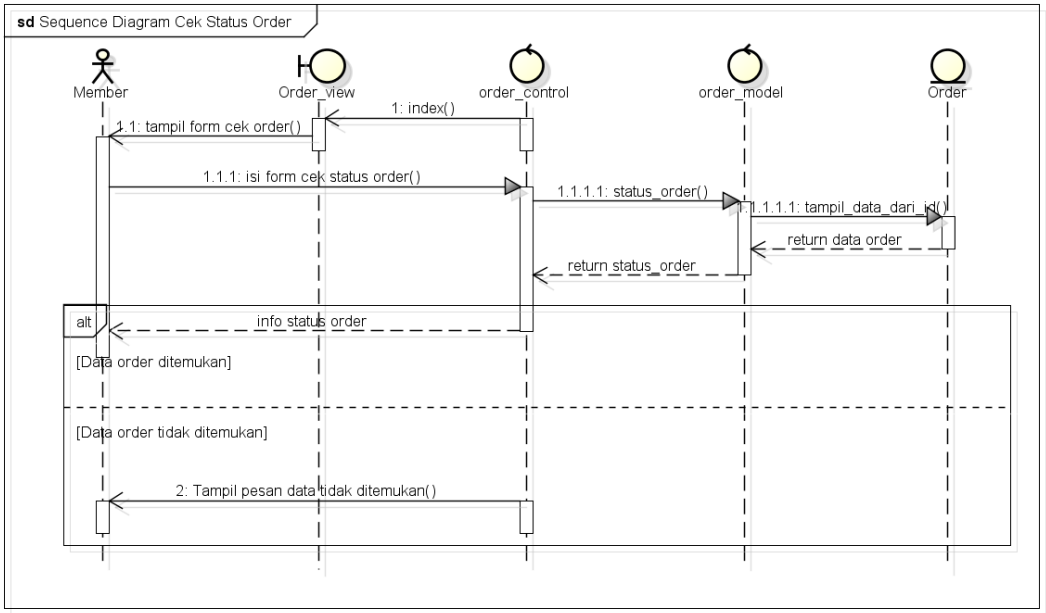
Sequence diagram backup dan restore *database* dapat dilihat pada gambar 3.24.



Gambar 3.24 *Sequence* Diagram Backup dan Restore *Database*

20. *Sequence* Diagram Cek Status Order

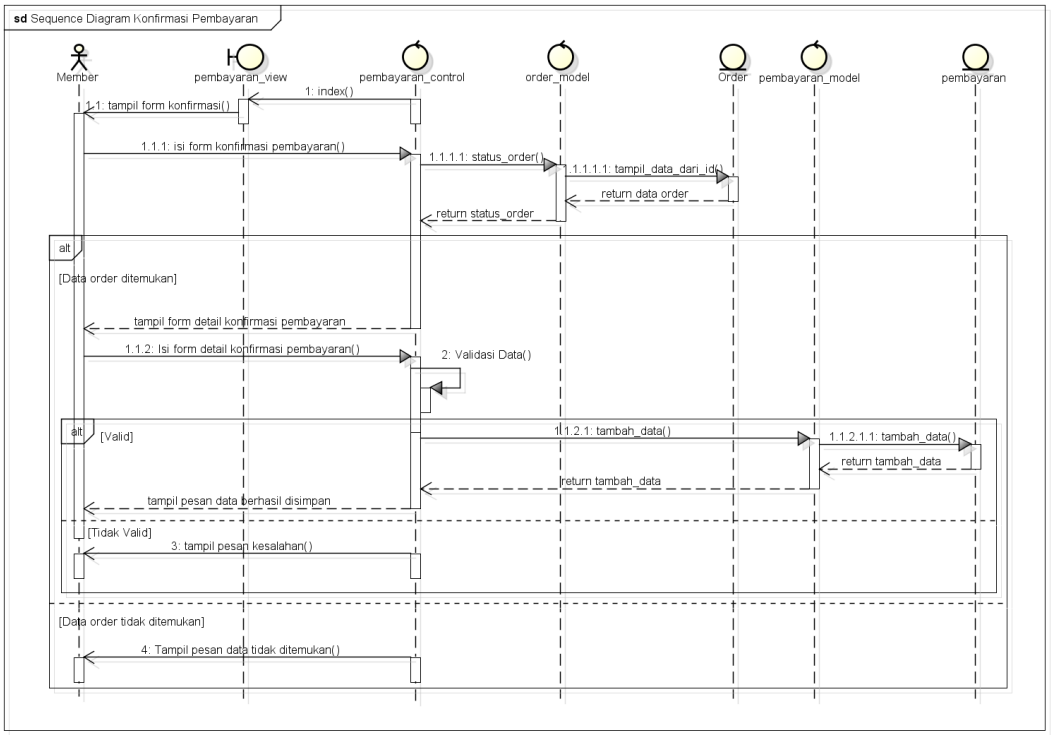
Sequence diagram cek status order dapat dilihat pada gambar 3.25.



Gambar 3.25 *Sequence* Diagram Cek Status Order

21. *Sequence* Diagram Konfirmasi Pembayaran

Sequence diagram konfirmasi pembayaran dapat dilihat pada gambar 3.26.

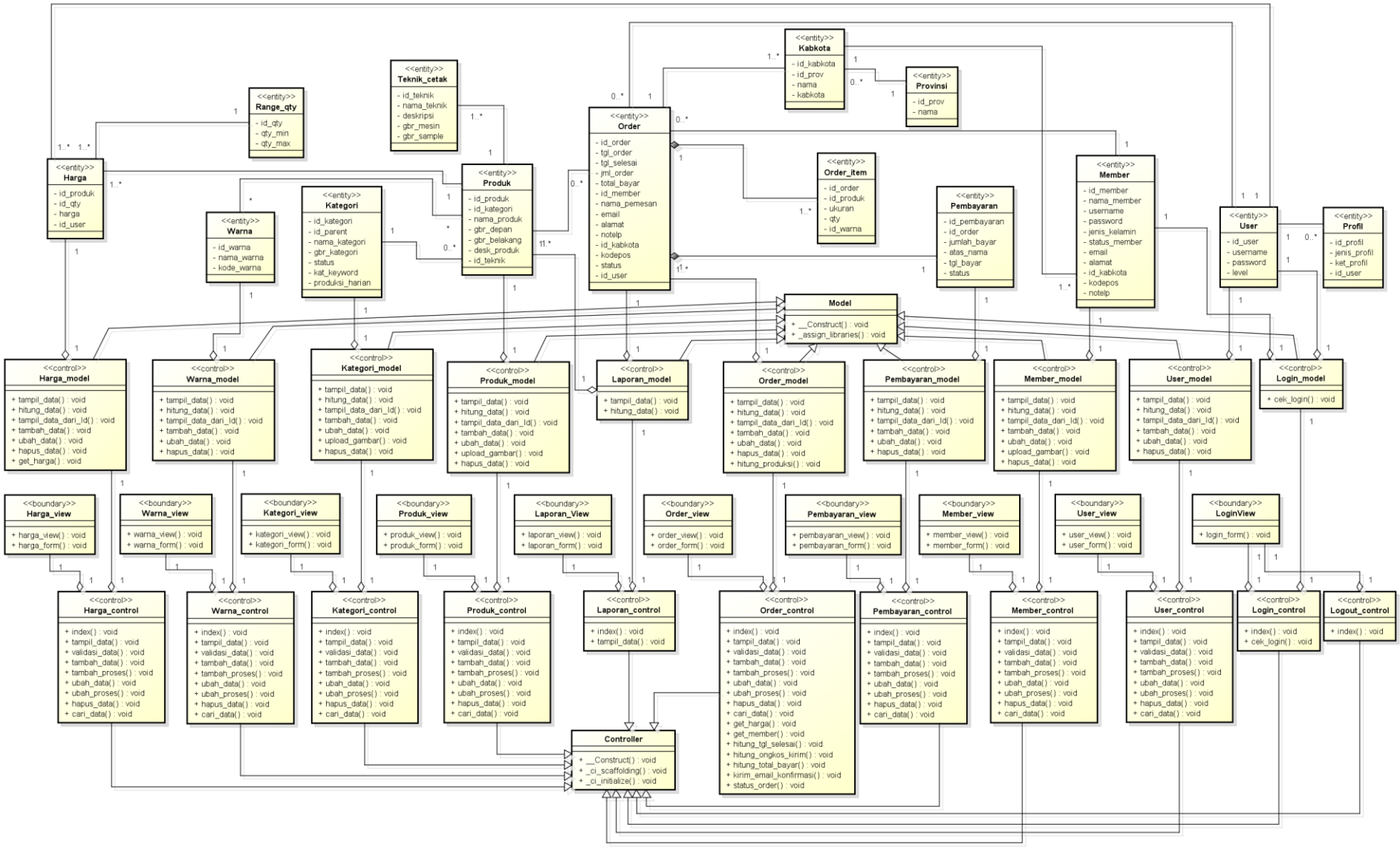


Gambar 3.26 *Sequence* Diagram Konfirmasi Pembayaran

3.1.4.5 *Class* Diagram

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara mereka. *Class* diagram juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan objek tersebut[2].

Class diagram *E-commerce* Pabrik Baju 12Titan dapat dilihat pada gambar 3.27.



Gambar 3.27 *Class* Diagram *E-commerce* Pabrik Baju 12Titan

Setiap kelas pada gambar 3.25 memiliki jenis kelasnya masing-masing seperti yang terlihat pada tabel 3.44.

Tabel 3.44 Deskrpsi *Class*

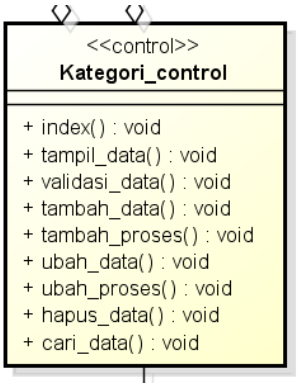
No	Nama	Jenis
1	Kategori_control	Control
2	Produk_control	Control
3	Laporan_control	Control
4	Pesanan_control	Control
5	Pembayaran_control	Control
6	User_control	Control
7	Member_control	Control
8	Login_control	Control
9	Logout_control	Control
10	Harga_control	Control
11	Warna_control	Control
12	Kategori_model	Control
13	Produk_model	Control
14	Laporan_model	Control
15	Pesanan_model	Control
16	Pembayaran_model	Control
17	User_model	Control
18	Member_model	Control
19	Login_model	Control
20	Harga_model	Control
21	Warna_model	Control
22	Kategori_view	Boundary
23	Produk_view	Boundary
24	Laporan_view	Boundary
25	Pesanan_view	Boundary
26	Pembayaran_view	Boundary
27	User_view	Boundary
28	Member_view	Boundary
29	Login_view	Boundary
30	Harga_view	Boundary
31	Warna_view	Boundary
32	Kategori	Entity
33	Produk	Entity
34	Pesanan	Entity
35	Profil	Entity
36	Pembayaran	Entity
37	User	Entity
38	Member	Entity
39	Harga	Entity
40	Range_qty	Entity
41	Teknik_cetak	Entity

42	Warna	Entity
43	Order_item	Entity
44	Kabkota	Entity
45	Provinsi	Entity

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

1. *Class* Kategori_control

Class Kategori_control merupakan *class* yang digunakan untuk mengatur proses pengolahan pada data kategori dan bertugas untuk memanggil *class-class* yang lainnya. Atribut dan operasi yang dimiliki *class* Kategori_control dapat dilihat pada gambar 3.28.



Gambar 3.28 Deskripsi *Class* Kategori_control

a. Deskripsi Atribut

Class Kategori_control tidak memiliki atribut.

b. Deskripsi Operasi

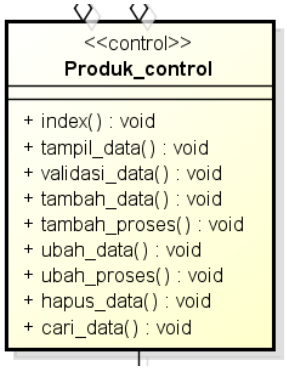
Deskripsi operasi *class* kategori_control dapat dilihat pada tabel 3.45.

Tabel 3.45 Deskripsi Operasi *Class* kategori_control

Nama	Keterangan
index() :	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, prosedur ini akan memanggil prosedur tampilData untuk menampilkan data kategori
tampil_data() : void	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* Kategori_model untuk mengambil data pada *database*.
validasi_data() : void	Prosedur yang dipanggil untuk memvalidasi data kategori yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
tambah_data() : void	Prosedur untuk melakukan tambah data. Memanggil Kategori_View untuk menampilkan form tambah.
tambah_proses() : void	Prosedur untuk melakukan proses penambahan data kategori berdasarkan data masukan dari form tambah dan akan memanggil *class* KategoriModel untuk melakukan proses tambah data kategori dan menyimpannya pada *database*.
ubah_data() : void	Prosedur untuk melakukan perubahan data kategori, akan memanggil *class* KategoriModel untuk mengambil data kemudian memanggil *class* KategoriView untuk menampilkan form ubah.
ubah_proses() : void	Prosedur untuk memproses data masukan dari form ubah, memanggil KategoriModel untuk menyimpan data yang sudah diubah.
hapus_data() : void	Prosedur untuk menghapus data kategori, akan menampilkan pesan konfirmasi apakah data kategori akan dihapus, jika ya maka akan memanggil KategoriModel untuk melakukan penghapusan data pada *database*.
cari_data() : void	Prosedur untuk mencari data kategori pada *database* dan menampilkannya.

2. *Class* Produk_control

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



Gambar 3.29 Deskripsi *Class* Produk_control

a. Deskripsi Atribut

Class Produk_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* produk_control dapat dilihat pada tabel 3.46.

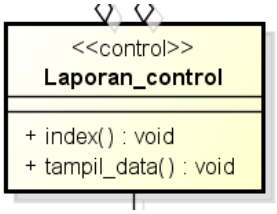
Tabel 3.46 Deskripsi Operasi *Class* produk_control

Nama	Keterangan
index() :	Prosedur yang dijalankan ketika *class* pertama kali dipanggil, prosedur ini akan memanggil prosedur tampilData untuk menampilkan data produk
tampil_data() : void	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* ProdukModel untuk mengambil data pada *database*.
validasi_data() : void	Prosedur yang dipanggil untuk memvalidasi data produk yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
tambah_data() : void	Prosedur untuk melakukan tambah data. Memanggil ProdukView untuk menampilkan form tambah.
tambah_proses() : void	Prosedur untuk melakukan proses penambahan data produk berdasarkan data masukan dari form tambah dan akan memanggil *class* ProdukModel untuk melakukan proses tambah data produk dan menyimpannya pada *database*.
ubah_data() : void	Prosedur untuk melakukan perubahan data produk, akan memanggil *class* ProdukModel untuk mengambil data kemudian memanggil *class* ProdukView untuk menampilkan form ubah.
ubah_proses() : void	Prosedur untuk memproses data masukan dari form ubah, memanggil ProdukModel untuk menyimpan data yang sudah diubah.

hapus_data() : void	Prosedur untuk menghapus data produk, akan menampilkan pesan konfirmasi apakah data produk akan dihapus, jika ya maka akan memanggil ProdukModel untuk melakukan penghapusan data pada *database*.
cari_data() : void	Prosedur untuk mencari data produk pada *database* dan menampilkannya.

3. *Class* Laporan_control

Class Laporan_control merupakan *class* yang digunakan untuk mengatur proses pembuatan laporan dan memanggil *class-class* lain 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

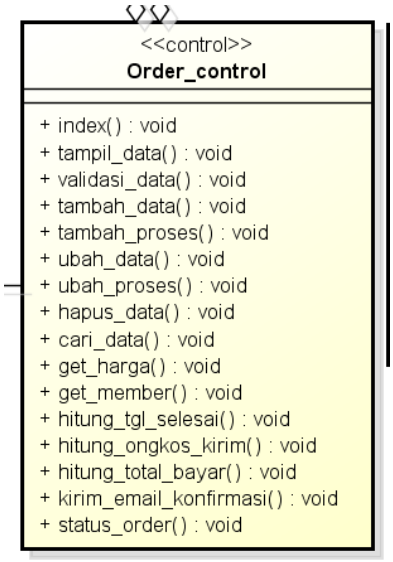
Deskripsi operasi *class* laporan_control dapat dilihat pada tabel 3.47.

Tabel 3.47 Deskripsi Operasi *Class* laporan_control

Nama	Keterangan
Index() : void	Prosedur yang pertama kali dijalankan, akan memanggil form Periode pada LaporanView.
tampil_data() : void	Prosedur untuk melakukan pembuatan laporan form yang sudah diisi sebelumnya dan kemudian menampilkannya.
cetakLaporan() : void	Prosedur untuk mencetak laporan yang sudah dibuat.

4. *Class* Order_control

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



Gambar 3.31 Deskripsi *Class* order_control

a. Deskripsi Atribut

Class Order_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* order_control dapat dilihat pada tabel 3.48.

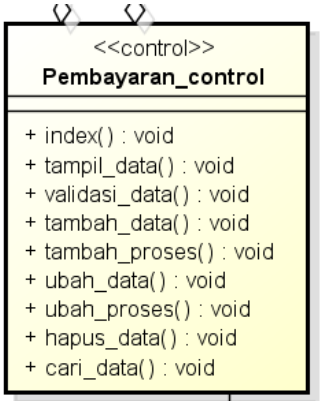
Tabel 3.48 Deskripsi Operasi *Class* order_control

Nama	Keterangan
index() :	Prosedur yang dijalankan ketika *class* pertama kali dipanggil, prosedur ini akan memanggil prosedur tampilData untuk menampilkan data order
tampil_data() : void	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* OrderModel untuk mengambil data pada *database*.
validasi_data() : void	Prosedur yang dipanggil untuk memvalidasi data order yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

tambah_data() : void	Prosedur untuk melakukan tambah data. Memanggil OrderView untuk menampilkan form tambah.
tambah_proses() : void	Prosedur untuk melakukan proses penambahan data order berdasarkan data masukan dari form tambah dan akan memanggil *class* OrderModel untuk melakukan proses tambah data order dan menyimpannya pada *database*.
ubah_data() : void	Prosedur untuk melakukan perubahan data order, akan memanggil *class* OrderModel untuk mengambil data kemudian memanggil *class* OrderView untuk menampilkan form ubah.
ubah_proses() : void	Prosedur untuk memproses data masukan dari form ubah, memanggil OrderModel untuk menyimpan data yang sudah diubah.
hapus_data() : void	Prosedur untuk menghapus data order, akan menampilkan pesan konfirmasi apakah data order akan dihapus, jika ya maka akan memanggil OrderModel untuk melakukan penghapusan data pada *database*.
cari_data() : void	Prosedur untuk mencari data order pada *database* dan menampilkannya.
Get_harga() : void	Prosedur untuk mengambil harga dari tabel harga
Get_member : void	Prosedur untuk mengambil data member
Hitung_tgl_selesai() : void	Prosedur untuk menghitung tanggal selesai
Hitung_ongkos_kirim() :void	Prosedur untuk menghitung ongkos kirim
Hitung_total_bayar() : void	Prosedur untuk menghitung total bayar
Kirim_email_konfirmasi(): void	Prosedur untuk mengirim email konfirmasi
Status_order(): void	Prosedur untuk melihat status suatu order/pesanan

5. *Class* Pembayaran_control

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



Gambar 3.32 Deskripsi *Class* Pembayaran_control

a. Deskripsi Atribut

Class Pembayaran_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* pembayaran_control dapat dilihat pada tabel 3.49.

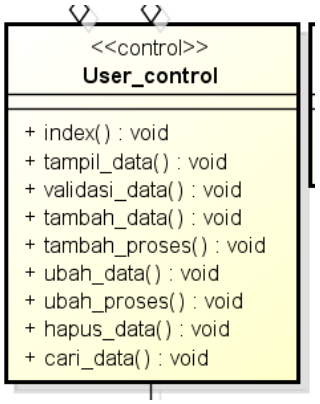
Tabel 3.49 Deskripsi Operasi *Class* pembayaran_control

Nama	Keterangan
index() :	Prosedur yang dijalankan ketika *class* pertama kali dipanggil, prosedur ini akan memanggil prosedur tampilData untuk menampilkan data pembayaran
tampil_data() : void	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* PembayaranModel untuk mengambil data pada *database*.
validasi_data() : void	Prosedur yang dipanggil untuk memvalidasi data pembayaran yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
tambah_data() : void	Prosedur untuk melakukan tambah data. Memanggil PembayaranView untuk menampilkan form tambah.
tambah_proses() : void	Prosedur untuk melakukan proses penambahan data pembayaran berdasarkan data masukan dari form tambah dan akan memanggil *class* PembayaranModel untuk melakukan proses tambah data pembayaran dan menyimpannya pada *database*.
ubah_data() : void	Prosedur untuk melakukan perubahan data pembayaran, akan memanggil *class* PembayaranModel untuk mengambil data kemudian memanggil *class* PembayaranView untuk menampilkan form ubah.
ubah_proses() : void	Prosedur untuk memproses data masukan dari form ubah,

	memanggil PembayaranModel untuk menyimpan data yang sudah diubah.
hapus_data() : void	Prosedur untuk menghapus data pembayaran, akan menampilkan pesan konfirmasi apakah data pembayaran akan dihapus, jika ya maka akan memanggil PembayaranModel untuk melakukan penghapusan data pada *database*.
cari_data() : void	Prosedur untuk mencari data pembayaran pada *database* dan menampilkannya.

6. *Class* User_control

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



Gambar 3.33 Deskripsi *Class* User_control

a. Deskripsi Atribut

Class User_control tidak memiliki atribut.

b. Deskripsi Operasi

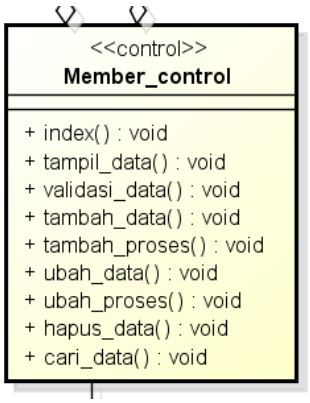
Deskripsi operasi *class* user_control dapat dilihat pada tabel 3.50.

Tabel 3.50 Deskripsi Operasi *Class* user_control

Nama	Keterangan
index() :	Prosedur yang dijalankan ketika *class* pertama kali dipanggil, prosedur ini akan memanggil prosedur tampilData untuk menampilkan data user
tampil_data() : void	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* UserModel untuk mengambil data pada *database*.
validasi_data() : void	Prosedur yang dipanggil untuk memvalidasi data user yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
tambah_data() : void	Prosedur untuk melakukan tambah data. Memanggil UserView untuk menampilkan form tambah.
tambah_proses() : void	Prosedur untuk melakukan proses penambahan data user berdasarkan data masukan dari form tambah dan akan memanggil *class* UserModel untuk melakukan proses tambah data user dan menyimpannya pada *database*.
ubah_data() : void	Prosedur untuk melakukan perubahan data user, akan memanggil *class* UserModel untuk mengambil data kemudian memanggil *class* UserView untuk menampilkan form ubah.
ubah_proses() : void	Prosedur untuk memproses data masukan dari form ubah, memanggil UserModel untuk menyimpan data yang sudah diubah.
hapus_data() : void	Prosedur untuk menghapus data user, akan menampilkan pesan konfirmasi apakah data user akan dihapus, jika ya maka akan memanggil UserModel untuk melakukan penghapusan data pada *database*.
cari_data() : void	Prosedur untuk mencari data user pada *database* dan menampilkannya.

7. *Class* Member_control

Class Member_control merupakan *class* yang digunakan untuk mengatur proses pengolahan pada data member dan bertugas untuk memanggil *class*-*class* lain yang berkaitan. Atribut dan operasi yang dimiliki *class* Member_control dapat dilihat pada gambar 3.34.



Gambar 3.34 Deskripsi *Class* Member_control

a. Deskripsi Atribut

Class Member_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* member_control dapat dilihat pada tabel 3.51.

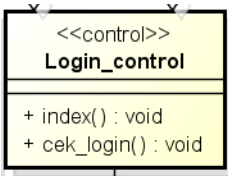
Tabel 3.51 Deskripsi Operasi *Class* member_control

Nama	Keterangan
index() :	Prosedur yang dijalankan ketika *class* pertama kali dipanggil, prosedur ini akan memanggil prosedur tampilData untuk menampilkan data member
tampil_data() : void	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* MemberModel untuk mengambil data pada *database*.
validasi_data() : void	Prosedur yang dipanggil untuk memvalidasi data member yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
tambah_data() : void	Prosedur untuk melakukan tambah data. Memanggil MemberView untuk menampilkan form tambah.
tambah_proses() : void	Prosedur untuk melakukan proses penambahan data member berdasarkan data masukan dari form tambah dan akan memanggil *class* MemberModel untuk melakukan proses tambah data member dan menyimpannya pada *database*.
ubah_data() : void	Prosedur untuk melakukan perubahan data member, akan memanggil *class* MemberModel untuk mengambil data kemudian memanggil *class* MemberView untuk menampilkan form ubah.
ubah_proses() : void	Prosedur untuk memproses data masukan dari form ubah, memanggil MemberModel untuk menyimpan data yang

	sudah diubah.
hapus_data() : void	Prosedur untuk menghapus data member, akan menampilkan pesan konfirmasi apakah data member akan dihapus, jika ya maka akan memanggil MemberModel untuk melakukan penghapusan data pada *database*.
cari_data() : void	Prosedur untuk mencari data member pada *database* dan menampilkannya.

8. *Class* Login_control

Class Login_control merupakan *class* yang digunakan untuk mengatur proses login untuk pengguna atau member dan bertugas untuk memanggil *class-class* lain yang berkaitan. Atribut dan operasi yang dimiliki *class* Login_control dapat dilihat pada gambar 3.35.



Gambar 3.35 Deskripsi *Class* Login_control

a. Deskripsi Atribut

Class Login_control tidak memiliki atribut.

b. Deskripsi Operasi

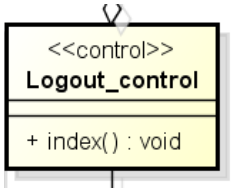
Deskripsi operasi *class* login_control dapat dilihat pada tabel 3.52.

Tabel 3.52 Deskripsi Operasi *Class* login_control

Nama	Keterangan
Index() : void	Prosedur yang pertama dijalankan dan akan memanggil Login_view untuk menampilkan form login.
Cek_login() : void	Prosedur yang dipanggil ketika user melakukan proses login, memvalidasi dan memanggil *class* Login_Model untuk mencari data user yang ada pada *database*.

9. *Class* Logout_control

Class Logout_control merupakan *class* yang digunakan untuk melakukan proses logout admin atau member dan memanggil *class-class* lain yang berkaitan. Atribut dan operasi yang dimiliki *class* Logout_control dapat dilihat pada gambar 3.36.



Gambar 3.36 Deskripsi *Class* Logout_control

a. Deskripsi Atribut

Class Logout_control tidak memiliki atribut.

b. Deskripsi Operasi

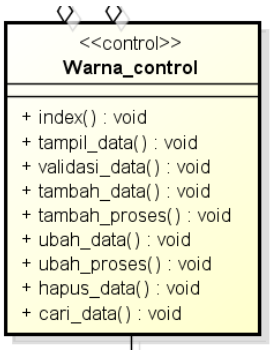
Deskripsi operasi *class* logout_control dapat dilihat pada tabel 3.53.

Tabel 3.53 Deskripsi Operasi *Class* logout_control

Nama	Keterangan
index() : void	Prosedur ketika memanggil logout_control, digunakan untuk keluar dari sistem dan kembali menampilkan form login.

10. *Class* Warna_control

Class Warna_control merupakan *class* yang digunakan untuk mengatur proses pengolahan pada data warna dan bertugas untuk memanggil *class-class* lain yang berkaitan. Atribut dan operasi yang dimiliki *class* Warna_control dapat dilihat pada gambar 3.37.



Gambar 3.37 Deskripsi *Class* Warna_control

a. Deskripsi Atribut

Class Warna_control tidak memiliki atribut.

b. Deskripsi Operasi

Deskripsi operasi *class* warna_control dapat dilihat pada tabel 3.54.

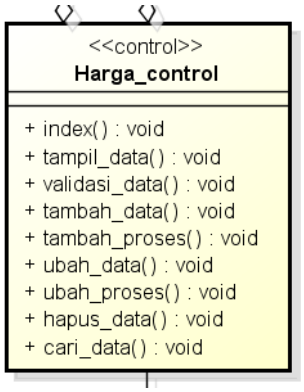
Tabel 3.54 Deskripsi Operasi *Class* warna_control

Nama	Keterangan
index() :	Prosedur yang dijalankan ketika *class* pertama kali dipanggil, prosedur ini akan memanggil prosedur tampilData untuk menampilkan data warna
tampil_data() : void	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan warnai perintah kepada *class* WarnaModel untuk mengambil data pada *database*.
validasi_data() : void	Prosedur yang dipanggil untuk memvalidasi data warna yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
tambah_data() : void	Prosedur untuk melakukan tambah data. Memanggil WarnaView untuk menampilkan form tambah.
tambah_proses() : void	Prosedur untuk melakukan proses penambahan data warna berdasarkan data masukan dari form tambah dan akan memanggil *class* WarnaModel untuk melakukan proses tambah data warna dan menyimpannya pada *database*.
ubah_data() : void	Prosedur untuk melakukan perubahan data warna, akan memanggil *class* WarnaModel untuk mengambil data kemudian memanggil *class* WarnaView untuk menampilkan form ubah.
ubah_proses() : void	Prosedur untuk memproses data masukan dari form ubah, memanggil WarnaModel untuk menyimpan data yang sudah diubah.

hapus_data() : void	Prosedur untuk menghapus data warna, akan menampilkan pesan konfirmasi apakah data warna akan dihapus, jika ya maka akan memanggil WarnaModel untuk melakukan penghapusan data pada *database*.
cari_data() : void	Prosedur untuk mencari data warna pada *database* dan menampilkannya.

11. *Class* Harga_control

Class Harga_control merupakan *class* yang digunakan untuk mengatur proses pengolahan pada data harga dan bertugas untuk memanggil *class-class* lain yang berkaitan. Atribut dan operasi yang dimiliki *class* Harga_control dapat dilihat pada gambar 3.38.



Gambar 3.38 Deskripsi *Class* Harga_control

a. Deskripsi Atribut

Class Harga_control tidak memiliki atribut.

b. Deskripsi Operasi

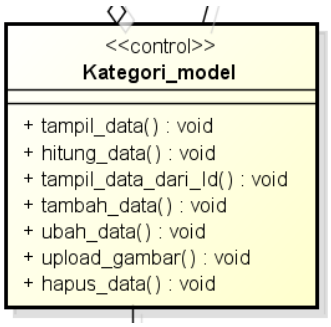
Deskripsi operasi *class* harga_control dapat dilihat pada tabel 3.55.

Tabel 3.55 Deskripsi Operasi *Class* harga_control

Nama	Keterangan
index() :	Prosedur yang dijalankan ketika *class* pertama kali dipanggil, prosedur ini akan memanggil prosedur tampilData untuk menampilkan data harga
tampil_data() : void	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan hargai perintah kepada *class* Harga_Model untuk mengambil data pada *database*.
validasi_data() : void	Prosedur yang dipanggil untuk memvalidasi data harga yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
tambah_data() : void	Prosedur untuk melakukan tambah data. Memanggil Harga_View untuk menampilkan form tambah.
tambah_proses() : void	Prosedur untuk melakukan proses penambahan data harga berdasarkan data masukan dari form tambah dan akan memanggil *class* Harga_Model untuk melakukan proses tambah data harga dan menyimpannya pada *database*.
ubah_data() : void	Prosedur untuk melakukan perubahan data harga, akan memanggil *class* Harga_Model untuk mengambil data kemudian memanggil *class* Harga_View untuk menampilkan form ubah.
ubah_proses() : void	Prosedur untuk memproses data masukan dari form ubah, memanggil Harga_Model untuk menyimpan data yang sudah diubah.
hapus_data() : void	Prosedur untuk menghapus data harga, akan menampilkan pesan konfirmasi apakah data harga akan dihapus, jika ya maka akan memanggil HargaModel untuk melakukan penghapusan data pada *database*.
cari_data() : void	Prosedur untuk mencari data harga pada *database* dan menampilkannya.

12. *Class* Kategori_model

Class Kategori_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah 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 3.39.



Gambar 3.39 Deskripsi *Class* Kategori_model

a. Deskripsi Atribut

Class Kategori_model tidak memiliki atribut.

b. Deskripsi Operasi

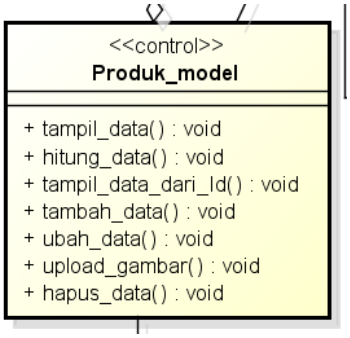
Deskripsi operasi *class* kategori_model dapat dilihat pada tabel 3.56.

Tabel 3.56 Deskripsi Operasi *Class* kategori_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data kategori.
hitung_data() : void	Prosedur menghitung jumlah data pada *database*
Tampil_Data_dari_Id() : void	Prosedur untuk mengambil data dari *database* berdasarkan id kategori.
tambah_data() : void	Prosedur yang dipanggil ketika proses tambah data kategori, melakukan perintah simpan data pada *database*.
ubah_data() : void	Prosedur yang dipanggil ketika proses ubah data kategori, melakukan perintah update data pada *database*.
upload_gambar() : void	Prosedur untuk menyimpan gambar yang diupload dan menyimpan alamatnya pada *database*.
hapus_data() : void	Prosedur yang dipanggil ketika hapus data, menghapus data kategori pada *database*.

13. *Class* Produk_model

Class Produk_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah data produk. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* Produk_model dapat dilihat pada gambar 3.40.



Gambar 3.40 Deskripsi *Class* Produk_model

a. Deskripsi Atribut

Class Produk_model tidak memiliki atribut.

b. Deskripsi Operasi

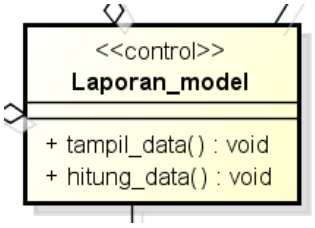
Deskripsi operasi *class* produk_model dapat dilihat pada tabel 3.57.

Tabel 3.57 Deskripsi Operasi *Class* produk_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data produk.
Hitung_data() : void	Prosedur menghitung jumlah data pada *database*
Tampil_Data_dari_Id() : void	Prosedur untuk mengambil data dari *database* berdasarkan id produk.
tambah_data() : void	Prosedur yang dipanggil ketika proses tambah data produk, melakukan perintah simpan data pada *database*.
ubah_data() : void	Prosedur yang dipanggil ketika proses ubah data produk, melakukan perintah update data pada *database*.
upload_gambar() : void	Prosedur untuk menyimpan gambar yang diupload dan menyimpan alamatnya pada *database*.
hapus_data() : void	Prosedur yang dipanggil ketika hapus data, menghapus data produk pada *database*.

14. *Class* Laporan_model

Class Laporan_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah data laporan. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* Laporan_model dapat dilihat pada gambar 3.41.



Gambar 3.41 Deskripsi *Class* Laporan_model

a. Deskripsi Atribut

Class Laporan_model tidak memiliki atribut.

b. Deskripsi Operasi

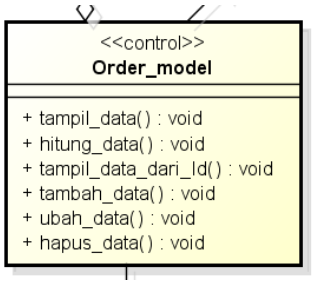
Deskripsi operasi *class* laporan_model dapat dilihat pada tabel 3.58.

Tabel 3.58 Deskripsi Operasi *Class* laporan_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data berdasarkan jangka waktu yang dimasukan.
Hitung_data() : void	Prosedur menghitung jumlah data pada *database*

15. *Class* Order_model

Class Order_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah data order. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* Order_model dapat dilihat pada gambar 3.42.



Gambar 3.42 Deskripsi *Class* Order_model

a. Deskripsi Atribut

Class Order_model tidak memiliki atribut.

b. Deskripsi Operasi

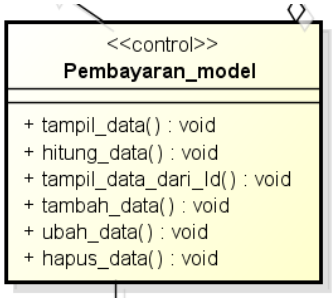
Deskripsi operasi *class* order_model dapat dilihat pada tabel 3.59.

Tabel 3.59 Deskripsi Operasi *Class* order_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data order.
Hitung_data() : void	Prosedur menghitung jumlah data pada *database*
Tampil_data_dari_id() : void	Prosedur untuk mengambil data dari *database* berdasarkan id order.
tambah_data() : void	Prosedur yang dipanggil ketika proses tambah data order, melakukan perintah simpan data pada *database*.
ubah_data() : void	Prosedur yang dipanggil ketika proses ubah data order, melakukan perintah update data pada *database*.
hapus_data() : void	Prosedur yang dipanggil ketika hapus data, menghapus data order pada *database*.

16. *Class* Pembayaran_model

Class Pembayaran_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah data pembayaran. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* Pembayaran_model dapat dilihat pada gambar 3.43.



Gambar 3.43 Deskripsi *Class* Pembayaran_model

a. Deskripsi Atribut

Class Pembayaran_model tidak memiliki atribut.

b. Deskripsi Operasi

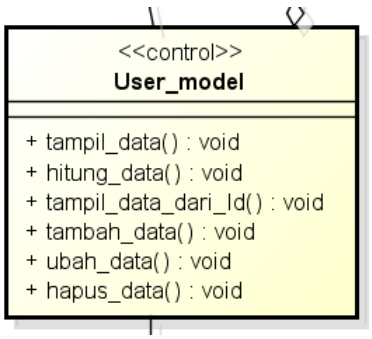
Deskripsi operasi *class* pembayaran_model dapat dilihat pada tabel 3.60.

Tabel 3.60 Deskripsi Operasi *Class* pembayaran_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data pembayaran.
hitung_data() : void	Prosedur menghitung jumlah data pada *database*
tampil_data_dari_Id() : void	Prosedur untuk mengambil data dari *database* berdasarkan id pembayaran.
tambah_data() : void	Prosedur yang dipanggil ketika proses tambah data pembayaran, melakukan perintah simpan data pada *database*.
ubah_data() : void	Prosedur yang dipanggil ketika proses ubah data pembayaran, melakukan perintah update data pada *database*.
hapus_data() : void	Prosedur yang dipanggil ketika hapus data, menghapus data pembayaran pada *database*.

17. *Class* User_model

Class User_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah data user. *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 3.44.



Gambar 3.44 Deskripsi *Class* User_model

a. Deskripsi Atribut

Class User_model tidak memiliki atribut.

b. Deskripsi Operasi

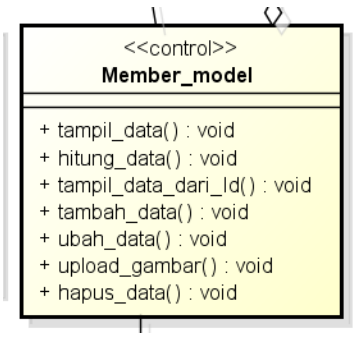
Deskripsi operasi *class* user_model dapat dilihat pada tabel 3.61.

Tabel 3.61 Deskripsi Operasi *Class* user_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data user.
hitung_data() : void	Prosedur menghitung jumlah data pada *database*
tampil_Data_dari_Id() : void	Prosedur untuk mengambil data dari *database* berdasarkan id user.
tambah_data() : void	Prosedur yang dipanggil ketika proses tambah data user, melakukan perintah simpan data pada *database*.
ubah_data() : void	Prosedur yang dipanggil ketika proses ubah data user, melakukan perintah update data pada *database*.
hapus_data() : void	Prosedur yang dipanggil ketika hapus data, menghapus data user pada *database*.

18. *Class* Member_model

Class Member_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah 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 3.45.



Gambar 3.45 Deskripsi *Class* Member_model

a. Deskripsi Atribut

Class Member_model tidak memiliki atribut.

b. Deskripsi Operasi

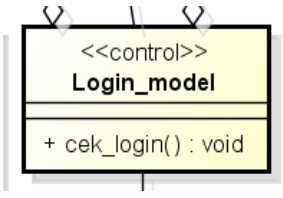
Deskripsi operasi *class* member_model dapat dilihat pada tabel 3.62.

Tabel 3.62 Deskripsi Operasi *Class* member_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data member.
hitung_data() : void	Prosedur menghitung jumlah data pada *database*
tampil_data_dari_Id() : void	Prosedur untuk mengambil data dari *database* berdasarkan id member.
tambah_data() : void	Prosedur yang dipanggil ketika proses tambah data member, melakukan perintah simpan data pada *database*.
ubah_data() : void	Prosedur yang dipanggil ketika proses ubah data member, melakukan perintah update data pada *database*.
Upload_gambar() : void	Prosedur untuk menyimpan gambar yang diupload dan menyimpan alamatnya pada *database*.
hapus_data() : void	Prosedur yang dipanggil ketika hapus data, menghapus data member pada *database*.

19. *Class* Login_model

Class Login_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah data login. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* Login_model dapat dilihat pada gambar 3.46.



Gambar 3.46 Deskripsi *Class* Login_model

a. Deskripsi Atribut

Class Login_model tidak memiliki atribut.

b. Deskripsi Operasi

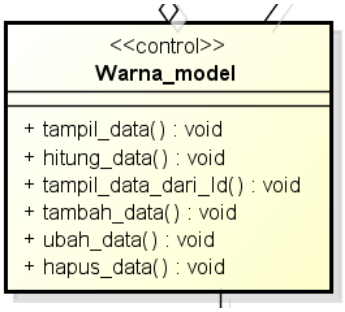
Deskripsi operasi *class* login_model dapat dilihat pada tabel 3.63.

Tabel 3.63 Deskripsi Operasi *Class* login_model

Nama	Keterangan
Cek_login() : void	Prosedur ketika user akan login, mengecek apakah data login yang dimasukan terdaftar pada *database*.

20. *Class* Warna_model

Class Warna_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah data warna. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* Warna_model dapat dilihat pada gambar 3.47.



Gambar 3.47 Deskripsi *Class* Warna_model

a. Deskripsi Atribut

Class Warna_model tidak memiliki atribut.

b. Deskripsi Operasi

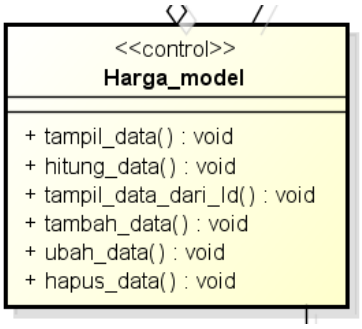
Deskripsi operasi *class* warna_model dapat dilihat pada tabel 3.64.

Tabel 3.64 Deskripsi Operasi *Class* warna_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data warna.
hitung_data() : void	Prosedur menghitung jumlah data pada *database*
tampil_data_dari_Id() : void	Prosedur untuk mengambil data dari *database* berdasarkan id warna.
tambah_data() : void	Prosedur yang dipanggil ketika proses tambah data warna, melakukan perintah simpan data pada *database*.
ubah_data() : void	Prosedur yang dipanggil ketika proses ubah data warna, melakukan perintah update data pada *database*.
hapus_data() : void	Prosedur yang dipanggil ketika hapus data, menghapus data warna pada *database*.

21. *Class* Harga_model

Class Harga_model merupakan *class* yang berisi prosedur-prosedur untuk mengolah data harga. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang dimiliki *class* Harga_model dapat dilihat pada gambar 3.48.



Gambar 3.48 Deskripsi *Class* Harga_model

a. Deskripsi Atribut

Class Harga_model tidak memiliki atribut.

b. Deskripsi Operasi

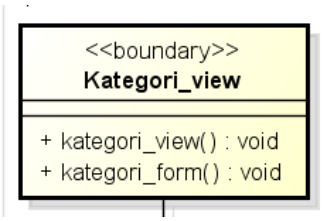
Deskripsi operasi *class* harga_model dapat dilihat pada tabel 3.65.

Tabel 3.65 Deskripsi Operasi *Class* harga_model

Nama	Keterangan
tampil_data() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data harga.
hitung_data() : void	Prosedur menghitung jumlah data pada *database*
Tampil_data_dari_Id() : void	Prosedur untuk mengambil data dari *database* berdasarkan id harga.
tambah_data() : void	Prosedur yang dipanggil ketika proses tambah data harga, melakukan perintah simpan data pada *database*.
ubah_data() : void	Prosedur yang dipanggil ketika proses ubah data harga, melakukan perintah update data pada *database*.
hapus_data() : void	Prosedur yang dipanggil ketika hapus data, menghapus data harga pada *database*.

22. *Class* Kategori_view

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



Gambar 3.49 Deskripsi *Class* Kategori_view

a. Deskripsi Atribut

Class Kategori_view tidak memiliki atribut.

b. Deskripsi Operasi

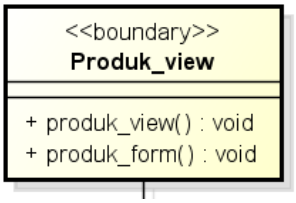
Deskripsi operasi *class* kategori_view dapat dilihat pada tabel 3.66.

Tabel 3.66 Deskripsi Operasi *Class* kategori_view

Nama	Keterangan
Kategori_view() : void	Prosedur untuk menampilkan data kategori yang hasil prosedur dari *class* Kategori_model dan Kategori_control.
Kategori_form() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* Kategori_control.

23. *Class* Produk_view

Class Produk_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* Produk_view dapat dilihat pada gambar 3.50.



Gambar 3.50 Deskripsi *Class* Produk_view

a. Deskripsi Atribut

Class Produk_view tidak memiliki atribut.

b. Deskripsi Operasi

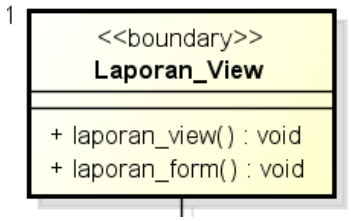
Deskripsi operasi *class* produk_view dapat dilihat pada tabel 3.67.

Tabel 3.67 Deskripsi Operasi *Class* produk_view

Nama	Keterangan
Produk_view() : void	Prosedur untuk menampilkan data produk yang hasil prosedur dari *class* Produk_model dan Produk_control.
Produk_form() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* Produk_control.

24. *Class* Laporan_view

Class Laporan_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* Laporan_view dapat dilihat pada gambar 3.51.



Gambar 3.51 Deskripsi *Class* Laporan_view

a. Deskripsi Atribut

Class Laporan_view tidak memiliki atribut.

b. Deskripsi Operasi

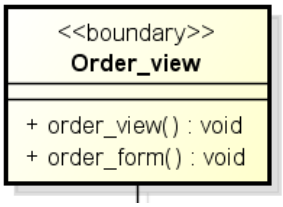
Deskripsi operasi *class* laporan_view dapat dilihat pada tabel 3.68.

Tabel 3.68 Deskripsi Operasi *Class* laporan_view

Nama	Keterangan
Laporan_view() : void	Prosedur untuk menampilkan data laporan yang hasil prosedur dari *class* Laporan_model dan Laporan_control.
Laporan_form() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* Laporan_control.

25. *Class* Order_view

Class Order_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* Order_view dapat dilihat pada gambar 3.52.



Gambar 3.52 Deskripsi *Class* Order_view

a. Deskripsi Atribut

Class Order_view tidak memiliki atribut.

b. Deskripsi Operasi

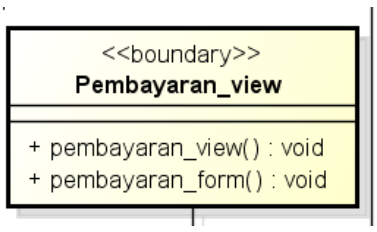
Deskripsi operasi *class* order_view dapat dilihat pada tabel 3.69.

Tabel 3.69 Deskripsi Operasi *Class* order_view

Nama	Keterangan
Order_view() : void	Prosedur untuk menampilkan data order yang hasil prosedur dari *class* Order_model dan Order_control.
Order_form() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* Order_control.

26. *Class* Pembayaran_view

Class Pembayaran_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* Pembayaran_view dapat dilihat pada gambar 3.53.



Gambar 3.53 Deskripsi *Class* Pembayaran_view

a. Deskripsi Atribut

Class Pembayaran_view tidak memiliki atribut.

b. Deskripsi Operasi

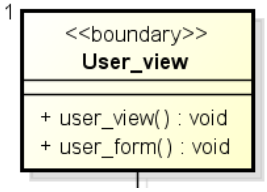
Deskripsi operasi *class* pembayaran_view dapat dilihat pada tabel 3.70.

Tabel 3.70 Deskripsi Operasi *Class* pembayaran_view

Nama	Keterangan
Pembayaran_view() : void	Prosedur untuk menampilkan data pembayaran berdasarkan hasil prosedur *class* Pembayaran_model dan Pembayaran_control.
Pembayaran_form() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* Pembayaran_Control.

27. *Class* User_view

Class User_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* User_view dapat dilihat pada gambar 3.54.



Gambar 3.54 Deskripsi *Class* User_view

a. Deskripsi Atribut

Class User_view tidak memiliki atribut.

b. Deskripsi Operasi

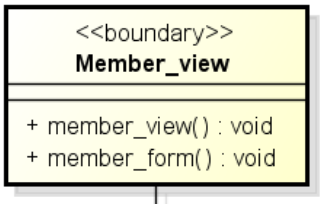
Deskripsi operasi *class* user_view dapat dilihat pada tabel 3.71.

Tabel 3.71 Deskripsi Operasi *Class* user_view

Nama	Keterangan
User_view() : void	Prosedur untuk menampilkan data user berdasarkan hasil prosedur *class* User_model dan User_control.
User_form() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* User_Control.

28. *Class* Member_view

Class Member_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* Member_view dapat dilihat pada gambar 3.55.



Gambar 3.55 Deskripsi *Class* Member_view

a. Deskripsi Atribut

Class Member_view tidak memiliki atribut.

b. Deskripsi Operasi

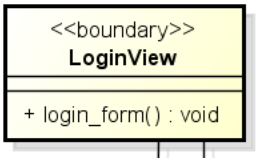
Deskripsi operasi *class* member_view dapat dilihat pada tabel 3.72.

Tabel 3.72 Deskripsi Operasi *Class* member_view

Nama	Keterangan
Member_view() : void	Prosedur untuk menampilkan data member berdasarkan hasil prosedur *class* Member_model dan Member_control.
Member_form() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* Member_Control.

29. *Class* Login_view

Class Login_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* Login_view dapat dilihat pada gambar 3.56.



Gambar 3.56 Deskripsi *Class* Login_view

a. Deskripsi Atribut

Class Login_view tidak memiliki atribut.

b. Deskripsi Operasi

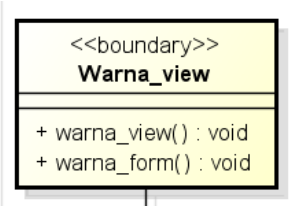
Deskripsi operasi *class* login_view dapat dilihat pada tabel 3.73.

Tabel 3.73 Deskripsi Operasi *Class* login_view

Nama	Keterangan
login_form() : void	Prosedur untuk menampilkan form yang akan digunakan untuk proses login.

30. *Class* Warna_view

Class Warna_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* Warna_view dapat dilihat pada gambar 3.57.



Gambar 3.57 Deskripsi *Class* Warna_view

a. Deskripsi Atribut

Class Warna_view tidak memiliki atribut.

b. Deskripsi Operasi

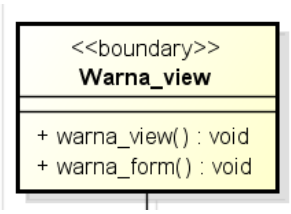
Deskripsi operasi *class* warna_view dapat dilihat pada tabel 3.74.

Tabel 3.74 Deskripsi Operasi *Class* warna_view

Nama	Keterangan
Warna_view() : void	Prosedur untuk menampilkan data warna berdasarkan hasil prosedur *class* Warna_model dan Warna_control.
Warna_form() : void	Prosedur menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* Warna_Control.

31. *Class* Harga_view

Class Harga_view merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang dimiliki *class* Harga_view dapat dilihat pada gambar 3.58.



Gambar 3.58 Deskripsi *Class* Harga_view

a. Deskripsi Atribut

Class Harga_view tidak memiliki atribut.

b. Deskripsi Operasi

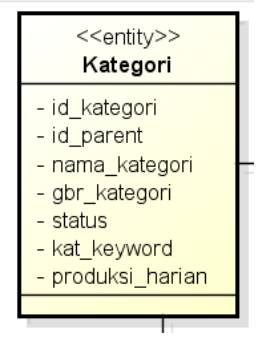
Deskripsi operasi *class* harga_view dapat dilihat pada tabel 3.75.

Tabel 3.75 Deskripsi Operasi *Class* harga_view

Nama	Keterangan
Harga_view() : void	Prosedur untuk menampilkan data harga berdasarkan hasil prosedur *class* Harga_model dan Harga_control.
Harga_form() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data dan ubah data pada *class* Harga_Control.

32. *Class* Kategori

Class Kategori merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Kategori dapat dilihat pada gambar 3.59.



Gambar 3.59 Deskripsi *Class* Kategori

a. Deskripsi Atribut

Deskripsi atribut untuk *class* kategori dapat dilihat pada tabel 3.76.

Tabel 3.76 Deskripsi Atribut *Class* Kategori

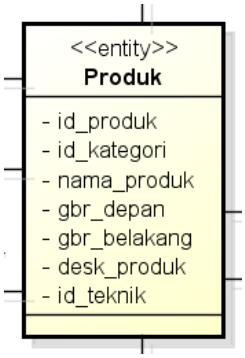
Nama	Tipe	Ukuran	Ket
Id_kategori	INT	5	Primary Key
Id_parent	INT	5	
Nama_kategori	Varchar	50	
Gbr_kategori	Varchar	100	
Status	INT	5	
Kat_Keyword	Varchar	50	
Produksi_harian	INT	5	

b. Deskripsi Operasi

Class Kategori tidak memiliki operasi.

33. *Class* Produk

Class Produk merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Produk dapat dilihat pada gambar 3.60.



Gambar 3.60 Deskripsi *Class* Produk

a. Deskripsi Atribut

Deskripsi atribut untuk *class* produk dapat dilihat pada tabel 3.77.

Tabel 3.77 Deskripsi Atribut *Class* produk

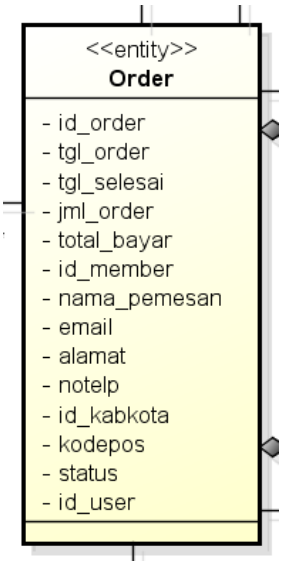
Nama	Tipe	Ukuran	Ket
Id_produk	INT	5	Primary Key
Id_kategori	INT	5	
Nama_produk	Varchar	100	
Gbr_depan	Varchar	100	
Gbr_belakang	Varchar	100	
Desk_produk	TEXT		
Id_teknik	INT	5	

b. Deskripsi Operasi

Class Produk tidak memiliki operasi.

34. *Class* Order

Class Order merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Order dapat dilihat pada gambar 3.61.



Gambar 3.61 Deskripsi *Class* Order

a. Deskripsi Atribut

Deskripsi atribut untuk *class* order dapat dilihat pada tabel 3.78.

Tabel 3.78 Deskripsi Atribut *Class* order

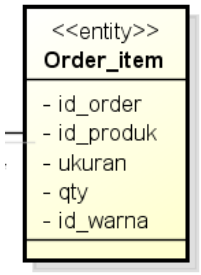
Nama	Tipe	Ukuran	Ket
Id_order	INT	5	Primary Key
Tgl_order	DATE		
Tgl_selesai	DATE		
Jml_order	INT	11	
Total_bayar	INT	11	
Id_member	INT	5	
Nama_pemesan	Varchar	50	
Email	Varchar	50	
Alamat	TEXT		
Notelp	Varchar	15	
Id_kabkota	INT	5	
Kodepos	INT	5	
Status	Varchar	20	
Id_user	INT	5	

b. Deskripsi Operasi

Class Order tidak memiliki operasi.

35. *Class* Order_item

Class Order_item merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Order_item dapat dilihat pada gambar 3.62.



Gambar 3.62 Deskripsi *Class* Order_item

a. Deskripsi Atribut

Deskripsi atribut untuk *class* order_item dapat dilihat pada tabel 3.79.

Tabel 3.79 Deskripsi Atribut *Class* order_item

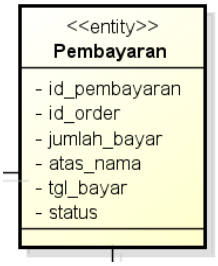
Nama	Tipe	Ukuran	Ket
Id_order	INT	5	Primary Key
Id_produk	INT	5	
ukuran	DATE		
Tgl_selesai	DATE		
qty	INT	5	
Id_warna	TEXT		

b. Deskripsi Operasi

Class Order_item tidak memiliki operasi.

36. *Class* Pembayaran

Class Pembayaran merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Pembayaran dapat dilihat pada gambar 3.63.



Gambar 3.63 Deskripsi *Class* Pembayaran

a. Deskripsi Atribut

Deskripsi atribut untuk *class* pembayaran dapat dilihat pada tabel 3.80.

Tabel 3.80 Deskripsi Atribut *Class* pembayaran

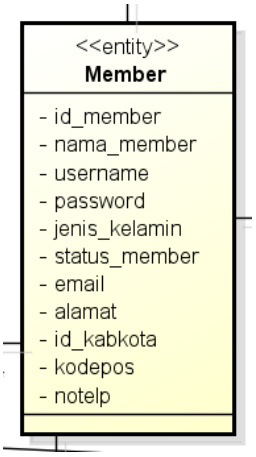
Nama	Tipe	Ukuran	Ket
Id_pembayaran	INT	5	Primary Key
Id_order	INT	5	
Jumlah_bayar	INT	11	
Atas_nama	Varchar	100	
Tgl_bayar	DATE		
status	Varchar	20	

b. Deskripsi Operasi

Class Pembayaran tidak memiliki operasi.

37. *Class* Member

Class Member merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Member dapat dilihat pada gambar 3.64.



Gambar 3.64 Deskripsi *Class* Member

a. Deskripsi Atribut

Deskripsi atribut untuk *class* member dapat dilihat pada tabel 3.82.

Tabel 3.81 Deskripsi Atribut *Class* member

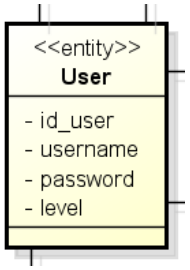
Nama	Tipe	Ukuran	Ket
Id_member	INT		Primary Key
Nama_member	Varchar	100	
username	Varchar	20	
Password	Varchar	51	
Jenis_kelamin	Varchar	20	
Status_member	Varchar	20	
Email	Varchar	30	
alamat	TEXT		
Id_kabkota	INT	3	
Kodepos	Varchar	5	
Notelp	Varchar	15	

b. Deskripsi Operasi

Class Member tidak memiliki operasi.

38. *Class* User

Class User merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* User dapat dilihat pada gambar 3.65.



Gambar 3.65 Deskripsi *Class* User

a. Deskripsi Atribut

Deskripsi atribut untuk *class* user dapat dilihat pada tabel 3.82.

Tabel 3.82 Deskripsi Atribut *Class* user

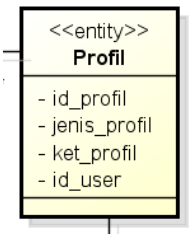
Nama	Tipe	Ukuran	Ket
Id_user	INT		Primary Key
Username	Varchar	30	
Password	Varchar	50	
Level	Varchar	20	

b. Deskripsi Operasi

Class User tidak memiliki operasi.

39. *Class* Profil

Class Profil merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Profil dapat dilihat pada gambar 3.66.



Gambar 3.66 Deskripsi *Class* Profil

a. Deskripsi Atribut

Deskripsi atribut untuk *class* profil dapat dilihat pada tabel 3.83.

Tabel 3.83 Deskripsi Atribut *Class* profil

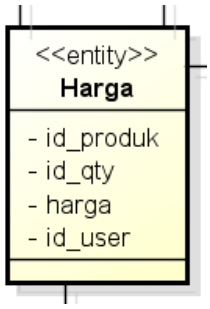
Nama	Tipe	Ukuran	Ket
Id_profil	INT		Primary Key
Jenis_profil	Varchar	20	
Ket_profil	Varchar	100	
Id_user	INT		

b. Deskripsi Operasi

Class Profil tidak memiliki operasi.

40. *Class* Harga

Class Harga merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Harga dapat dilihat pada gambar 3.67.



Gambar 3.67 Deskripsi *Class* Harga

a. Deskripsi Atribut

Deskripsi atribut untuk *class* harga dapat dilihat pada tabel 3.84.

Tabel 3.84 Deskripsi Atribut *Class* harga

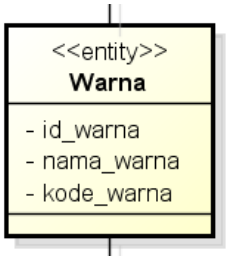
Nama	Tipe	Ukuran	Ket
Id_produk	INT	5	Primary Key
Id_qty	INT	5	
harga	INT	11	
Id_user	INT	5	

b. Deskripsi Operasi

Class Harga tidak memiliki operasi.

41. *Class* Warna

Class Warna merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Warna dapat dilihat pada gambar 3.68.



Gambar 3.68 Deskripsi *Class* Warna

a. Deskripsi Atribut

Deskripsi atribut untuk *class* warna dapat dilihat pada tabel 3.85.

Tabel 3.85 Deskripsi Atribut *Class* warna

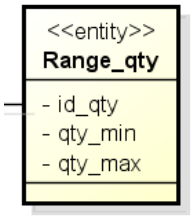
Nama	Tipe	Ukuran	Ket
Id_warna	INT		Primary Key
Nama_warna	Varchar	20	
Kode_warna	Varchar	7	

b. Deskripsi Operasi

Class Warna tidak memiliki operasi.

42. *Class* Range_qty

Class Range_qty merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Range_qty dapat dilihat pada gambar 3.69.



Gambar 3.69 Deskripsi *Class* Range_qty

a. Deskripsi Atribut

Deskripsi atribut untuk *class* range_qty dapat dilihat pada tabel 3.86.

Tabel 3.86 Deskripsi Atribut *Class* range_qty

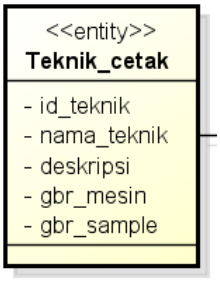
Nama	Tipe	Ukuran	Ket
Id_qty	INT		Primary Key
Qty_min	INT	5	
Qty_max	INT	5	

b. Deskripsi Operasi

Class Range_qty tidak memiliki operasi.

43. *Class* Teknik_cetak

Class Teknik_cetak merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Teknik_cetak dapat dilihat pada gambar 3.70.



Gambar 3.70 Deskripsi *Class* Teknik_cetak

a. Deskripsi Atribut

Deskripsi atribut untuk *class* teknik_cetak dapat dilihat pada tabel 3.87.

Tabel 3.87 Deskripsi Atribut *Class* teknik_cetak

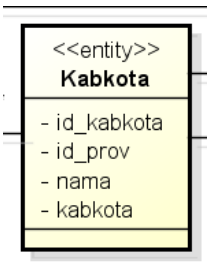
Nama	Tipe	Ukuran	Ket
id_teknik	INT	5	Primary Key
nama_teknik	Varchar	20	
Deskripsi	TEXT		
gbr_mesin	Vachar	50	
gbr_sample	Varchar	50	

b. Deskripsi Operasi

Class Teknik_cetak tidak memiliki operasi.

44. *Class* Kabkota

Class Kabkota merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Kabkota dapat dilihat pada gambar 3.71.



Gambar 3.71 Deskripsi *Class* Kabkota

a. Deskripsi Atribut

Deskripsi atribut untuk *class* kabkota dapat dilihat pada tabel 3.88.

Tabel 3.88 Deskripsi Atribut *Class* kabkota

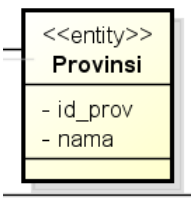
Nama	Tipe	Ukuran	Ket
Id_kabkota	INT	5	Primary Key
Id_prov	INT	5	
Nama	Varchar	50	
kabkota	Varchar	20	

b. Deskripsi Operasi

Class Kabkota tidak memiliki operasi.

45. *Class* Provinsi

Class Provinsi merupakan entitas yang berisi atribut-atribut yang digunakan dalam proses pengolahan data yang akan dilakukan. Atribut dan operasi yang dimiliki *class* Provinsi dapat dilihat pada gambar 3.72.



Gambar 3.72 Deskripsi *Class* Provinsi

a. Deskripsi Atribut

Deskripsi atribut untuk *class* provinsi dapat dilihat pada tabel 3.89.

Tabel 3.89 Deskripsi Atribut *Class* provinsi

Nama	Tipe	Ukuran	Ket
Id_prov	INT	5	Primary Key
nama	Varchar	50	

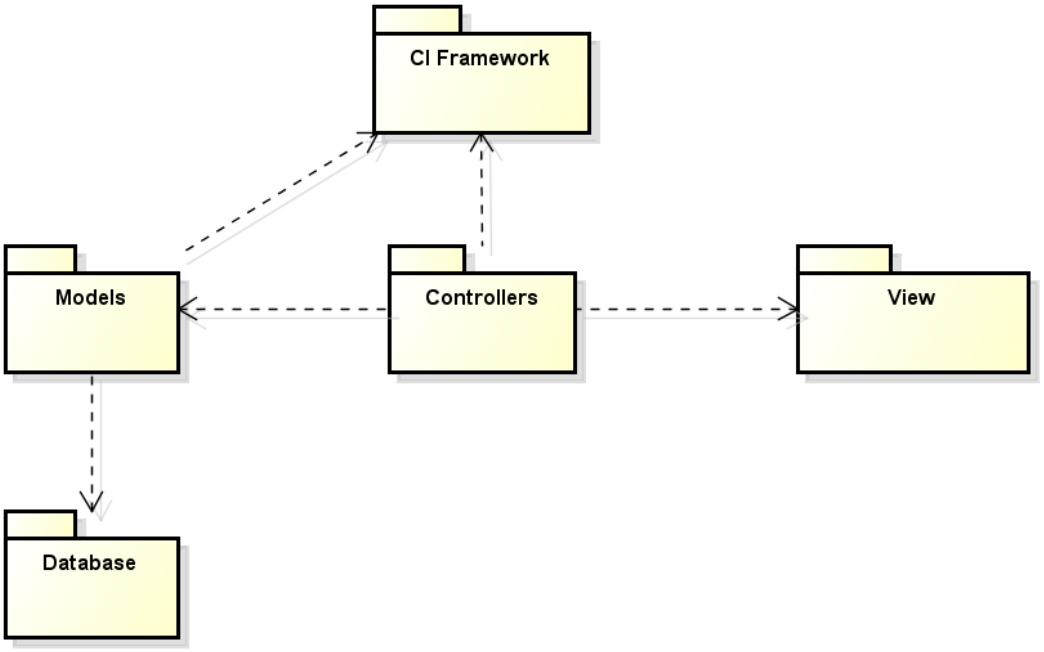
b. Deskripsi Operasi

Class Provinsi tidak memiliki operasi.

3.1.4.6 *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[2].

Pecakage diagram untuk pengelompokkan *class* pada *E-commerce* Pabrik Baju 12Titan dapat dilihat pada gambar 3.73.

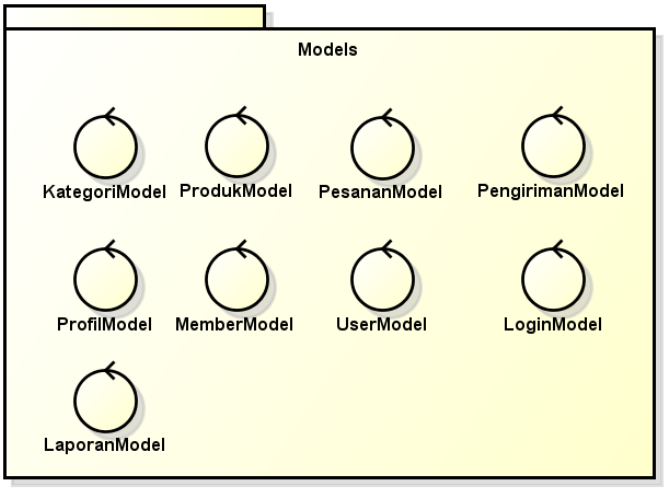


Gambar 3.73 *Package* Diagram *E-commerce* Pabrik Baju 12Titan

Deskripsi untuk *package* diagram adalah sebagai berikut :

1. ***Package* Models**

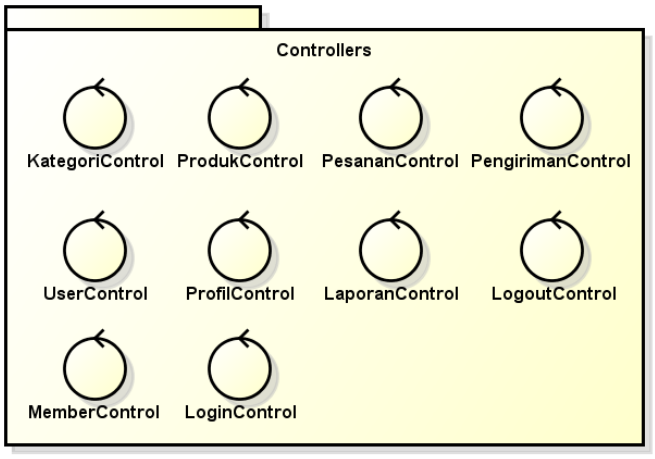
   Package Models berisi *class-class* model yang mengakses langsung pada *database* untuk dapat mengolah data. Isi dari *package* Models dapat dilihat pada gambar 3.74.



Gambar 3.74 Isi *Package* Models

2. *Package* Controllers

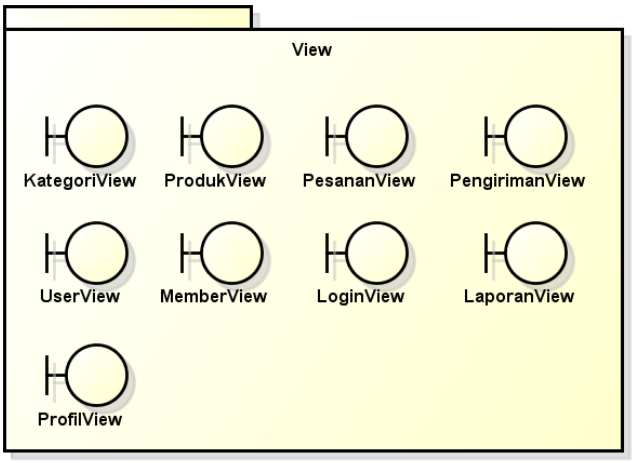
Package Controllers berisi *class-class* control yang merupakan pengatur dalam setiap proses yang ada. Isi dari *package* Controllers dapat dilihat pada gambar 3.75.



Gambar 3.75 Isi *Package* Controllers

3. *Package* View

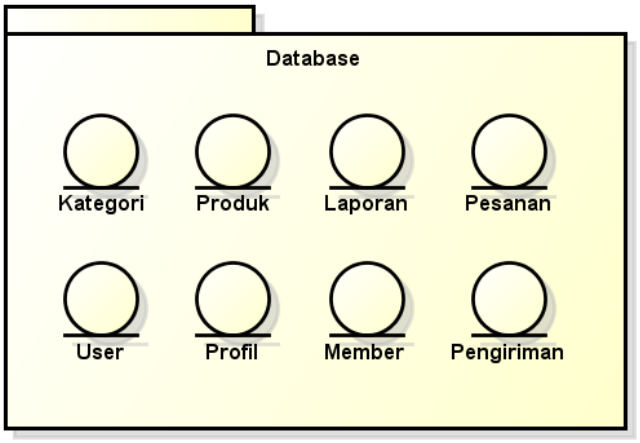
Package View berisi *class-class* view yang digunakan untuk menampilkan interface yang dapat berupa tampilan tabel atau form. Isi dari *Package* View dapat dilihat pada gambar 3.76.



Gambar 3.76 Isi *Package* Views

4. *Package Database*

Package Database berisi *class-class* sntity yang berisi atribut-atribut yang akan digunakan pada proses pengolahan data dan penyimpanan pada *database*. Isi dari *Package Database* dapat dilihat pada gambar 3.77.

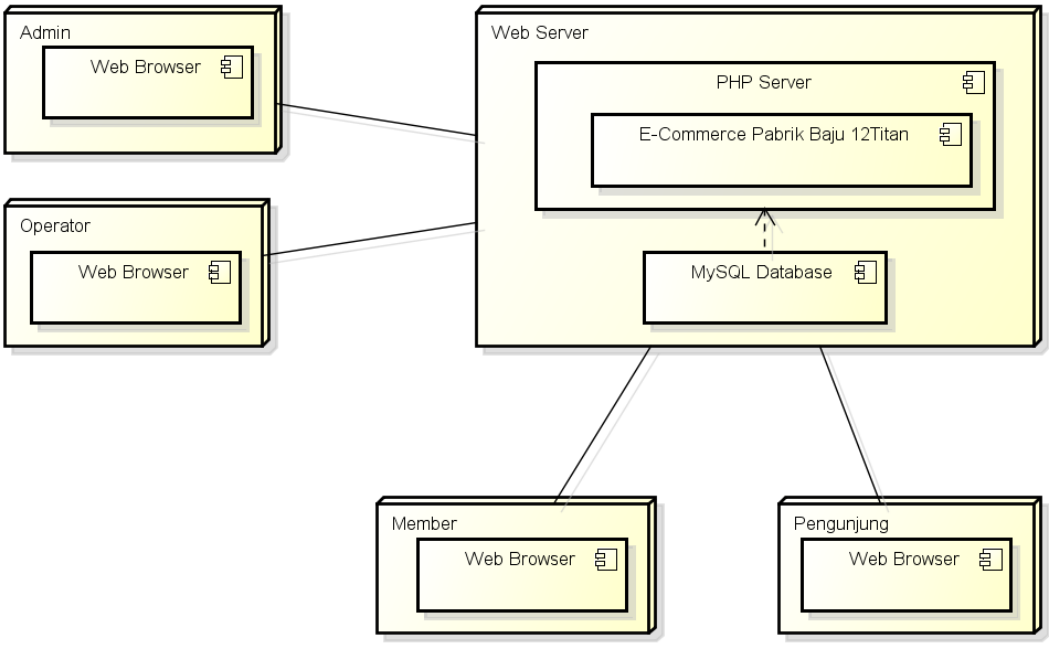


Gambar 3.77 Isi *Package Database*

3.1.4.7 *Deployment* Diagram

Deployment diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras mana[2].

Deployment diagram *E-commerce* Pabrik Baju 12Titan dapat dilihat pada gambar 3.76.



Gambar 3.78 *Deployment* Diagram *E-commerce* Pabrik Baju 12Titan

3.2 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah melalui tahapan analisis. Pada bagian ini akan dijelaskan perancangan sistem yang dimaksudkan untuk menggambarkan perbedaan antara sistem yang sedang berjalan dengan sistem yang diusulkan.

3.2.1 Perancangan Data

Perancangan data atau lebih dikenal dengan perancangan basis data yaitu menciptakan atau merancang data yang terhubung dan disimpan secara bersamasama. Untuk menggambarkannya digunakanlah skema relasi dan struktur *file*. Dari dua hasil tersebut, implementasi basis data akan bisa dikerjakan.

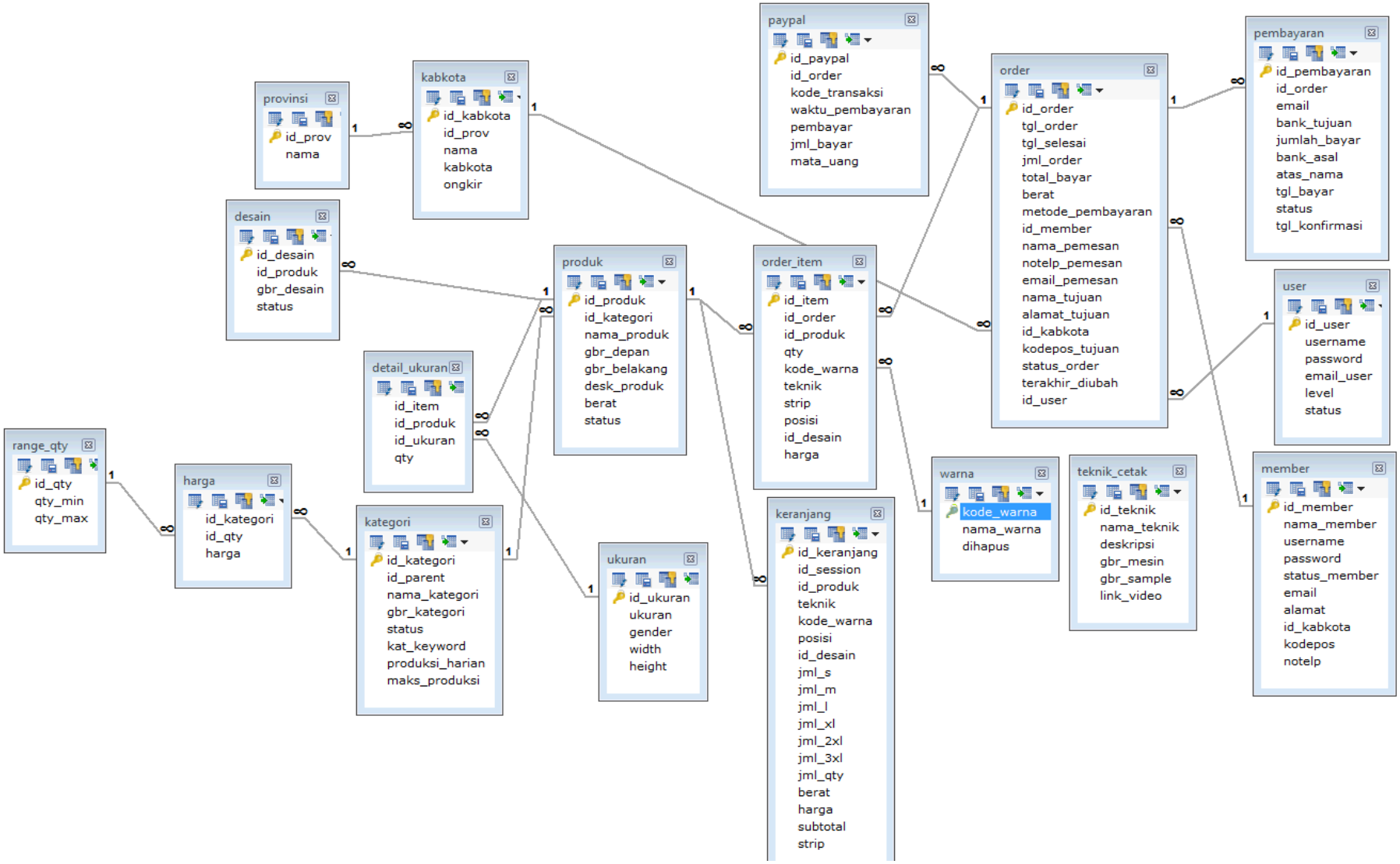
3.2.1.1 Skema Relasi

Skema relasi merupakan penggambaran hubungan diantara dua tabel atau lebih pada sistem *database*. Tabel yang ada pada aplikasi yang akan dibangun adalah sebagai berikut :

1. Desain {id_desain, id_produk, gbr_desain, status}
2. Detail_ukuran {id_item, id_produk, id_ukuran, qty}
3. Harga {id_produk, id_qty, harga}
4. Kabkota {id_kabkota, id_prov, nama, kabkota, ongkir}
5. Kategori {id_kategori, id_parent, nama_kategori, gbr_kategori, status, kat_keyword, produksi_harian, maks_produksi}
6. Keranjang {id_keranjang, id_session, id_produk, teknik, kode_warna, posisi, id_desain, jml_s, jml_m, jml_l, jml_xl, jml_2xl, jml_3xl, jml_qty, berat, harga, subtotal, strip}
7. Produk {id_produk, id_kategori, nama_produk, gbr_depan, gbr_belakang, desk_produk, berat, status}
8. Warna {kode_warna, nama_warna, dihapus}
9. Range_qty {id_qty, qty_min, qty_max}
10. Ukuran {id_ukuran, ukuran, gender, width, height}
11. Teknik_cetak {id_teknik, nama_teknik, deskripsi, gbr_mesin, gbr_sample, link_video}
12. Provinsi {id_prov, nama_prov}

13. Order {id_order, tgl_order, tgl_selesai, jml_order, total_bayar, berat, metode_pembayaran, id_member, nama_pemesan, notelp_pemesan, email_pemesan, nama_tujuan, alamat_tujuan, id_kabkota, kodepos_tujuan, status_order, id_user}
14. Paypal {id_paypal, id_order, kode_transaksi, waktu_pembayaran, pemabayar, jml_bayar, mata_uang}
15. Order_item {id_item, id_order, id_produk, qty, kode_warna, teknik, strip, posisi, id_desain, harga}
16. Pembayaran {id_pembayaran, id_order, email, jumlah_bayar, bank_asal, atas_nama, tgl_bayar, status, tgl_konfirmasi}
17. Member {id_member, nama_member, username, password, status_member, email, alamat, id_kabkota, kodepos, notelp}
18. User {id_user, username, password, email_user, level, status}

Skema relasi *E-commerce* pada Pabrik Baju 12Titan dapat dilihat pada gambar 3.79.



Gambar 3.79 Skema Relasi *E-commerce* Pabrik Baju 12Titan

3.2.1.2 Struktur Tabel

Struktur tabel adalah spesifikasi *file* dari tiap tabel yang digunakan dalam sistem yang akan dibangun. Struktur tabel dari sistem yang dibangun adalah sebagai berikut:

1. **Desain**

   Nama tabel : Desain

   Primary Key : id_desain

Tabel 3.90 Struktur Tabel Desain

Nama Field	Tipe	Ukuran	Keterangan
Id_desain	INT	2	Primary Key
Id_produk	INT	2	
Gbr_produk	Varchar	50	
Status	INT	1	

2. **Detail Ukuran**

   Nama tabel : detail_ukuran

   Primary Key : -

Tabel 3.91 Struktur Tabel Detail ukuran

Nama Field	Tipe	Ukuran	Keterangan
Id_item	INT	2	Primary Key
Id_produk	INT	2	Foreign Key Ref produk(id_produk)
Id_ukuran	INT	2	Foreign Key Ref ukuran(id_ukuran)
Qty	INT	3	

3. **Kategori**

   Nama tabel : Kategori

   Primary Key : id_kategori

Tabel 3.92 Struktur Tabel Kategori

Nama Field	Tipe	Ukuran	Keterangan
id_kategori	INT	2	Primary Key
id_parent	INT	2	
nama_kategori	Varchar	50	
gbr_kategori	Varchar	100	
Status	INT	2	
kat_keyword	Varchar	50	
Produksi_harian	INT	4	
Maks_produksi	INT	4	

4. Produk

Nama tabel : Produk

Primary Key : id_produk

Tabel 3.93 Struktur Tabel Produk

Nama Field	Tipe	Ukuran	Keterangan
id_produk	INT	3	Primary Key
id_kategori	INT	2	Foreign Key Ref Kategori(id_kategori)
nama_produk	Varchar	100	
gbr_depan	Varchar	100	
gbr_belakang	Varchar	100	
berat	float		
desk_produk	TEXT		
status	INT	3	

5. Warna

Nama tabel : Warna

Primary Key : kode_warna

Tabel 3.94 Struktur Tabel Warna

Nama Field	Tipe	Ukuran	Keterangan
kode_warna	varchar	7	Primary Key
Nama_warna	Varchar	30	
dihapus	char	1	

6. Ukuran

Nama tabel : Ukuran

Primary Key : id_ukuran

Tabel 3.95 Struktur Tabel Kategori

Nama Field	Tipe	Ukuran	Keterangan
id_ukuran	INT	3	Primary Key
Ukuran	Varchar	5	
Gender	Varchar	100	
Width	INT	4	
Height	INT	4	

7. Teknik_cetak

Nama tabel : Teknik_cetak

Primary Key : id_teknik

Tabel 3.96 Struktur Tabel Teknik_cetak

Nama Field	Tipe	Ukuran	Keterangan
id_teknik	INT	3	Primary Key
Nama_teknik	Varchar	50	
Deskripsi	TEXT		
Gbr_mesin	Varchar	100	
Gbr_sample	Varchar	100	
Link_video	Varchar	100	

8. Range_qty

Nama tabel : Range_qty

Primary Key : id_qty

Tabel 3.97 Struktur Tabel Range_qty

Nama Field	Tipe	Ukuran	Keterangan
Id_qty	INT	5	Primary Key
Qty_min	INT	5	
Qty_max	INT	5	

9. Harga

Nama tabel : Harga

Primary Key : -

Tabel 3.98 Struktur Tabel Harga

Nama Field	Tipe	Ukuran	Keterangan
Id_kategori	INT	5	Foreign Key Ref kategori(id_kategori)
Id_qty	INT	5	Foreign Key Ref range_qty(id_qty)
Harga	INT	11	

10. Paypal

Nama tabel : paypal

Primary Key : id_paypal

Tabel 3.99 Struktur Tabel Pembayaran

Nama Field	Tipe	Ukuran	Keterangan
Id_paypal	INT	3	Primary Key
Id_order	INT	5	Foreign Key Ref order(id_order)
Kode_transaksi	INT	10	
Waktu_pembayaran	Varchar	100	
Pembayar	DATE		
Jml_bayar	Varchar	20	
Mata_uang	varchar	10	

11. Pembayaran

Nama tabel : Pembayaran

Primary Key : id_pembayaran

Tabel 3.100 Struktur Tabel Pembayaran

Nama Field	Tipe	Ukuran	Keterangan
Id_pembayaran	INT	3	Primary Key
Id_order	INT	5	Foreign Key Ref order(id_order)
Email	Varchar	50	
Bank_tujuan	Varchar	10	

Jumlah_bayar	INT	10	
Atas_nama	Varchar	100	
Tgl_bayar	DATE		
Status	Varchar	20	
Tgl_konfirmasi	DATE		

12. Kabkota

Nama tabel : Kabkota

Primary Key : id_kabkota

Tabel 3.101 Struktur Tabel Kabkota

Nama Field	Tipe	Ukuran	Keterangan
Id_kabkota	INT	3	Primary Key
Id_prov	INT	3	Foreign Key Ref Provinsi(id_prov)
Nama	Varchar	50	
Kabkota	Varchar	20	
ongkir	INT	11	

13. Provinsi

Nama tabel : Provinsi

Primary Key : id_prov

Tabel 3.102 Struktur Tabel Provinsi

Nama Field	Tipe	Ukuran	Keterangan
Id_prov	INT	3	Primary Key
Nama	Varchar	100	

14. Keranjang

Nama tabel : Keranjang

Primary Key : id_keranjang

Tabel 3.103 Struktur Tabel Keranjang

Nama Field	Tipe	Ukuran	Keterangan
Id_keranjang	INT	3	Primary Key

Id_session	Varchar	35	
Id_produk	INT	5	Foreign Key Ref produk(id_produk)
Teknik	Varchar	10	
Kode_warna	Varchar	7	
Posisi	Varchar	20	
Id_desain	INT	2	Foreign Key Ref desain(id_desain)
Jml_s	INT	2	
Jml_m	INT	2	
Jml_l	INT	2	
Jml_xl	INT	2	
Jml_2xl	INT	2	
Jml_3xl	INT	2	
Jml_qty	INT	2	
Berat	Float		
Harga	INT	11	
Subtotal	INT	11	
strip	INT	11	

15. Order

Nama tabel : Order

Primary Key : id_order

Tabel 3.104 Struktur Tabel Order

Nama Field	Tipe	Ukuran	Keterangan
Id_order	INT	3	Primary Key
Tgl_order	DATE		
Tgl_selesai	DATE		
Jml_order	INT	5	
Total_bayar	INT	5	
Metode_pembayaran	Varchar	10	
Id_member	INT	5	Foreign Key Ref member(id_member)
Nama_pemesan	INT	2	
Notelp_pemesan	Varchar	15	
Email_pemesan	Varchar	100	
Nama_tujuan	Varchar	100	
Alamat_tujuan	TEXT		
Id_kabkota	INT		Foreign Key Ref kabkota(id_kabkota)
Kodepos_tujuan	INT	5	
Status_order	Varchar	30	

16. Order_item

Nama tabel : Order_item

Primary Key : Id_item

Tabel 3.105 Struktur Tabel Order_item

Nama Field	Tipe	Ukuran	Keterangan
Id_item	INT	3	Primary Key
Id_order	INT	3	Foreign Key Ref Order(id_order)
Id_produk	INT	5	Foreign Key Ref produk(id_produk)
Qty	INT	5	
kode_warna	INT	5	Foreign Key Ref warna(kode_warna)
teknik	Varchar	10	
Strip	INT	5	
Posisi	Varchar	20	
Id_desain	INT	2	Foreign Key Ref desain(id_desain)
Harga	INT	11	

17. Member

Nama tabel : Member

Primary Key : id_member

Tabel 3.106 Struktur Tabel Member

Nama Field	Tipe	Ukuran	Keterangan
Id_member	INT	5	Primary Key
Nama_member	Varchar	5	
Username	INT	4	
Password	INT	4	
Status_member	Varchar	20	
Email	Varchar	50	
Alamat	TEXT		
Id_kabkota	INT	5	Foreign Key kabkota(id_kabkota)
kodepos	Varchar	5	
Notelp	Varchar	5	

18. User

Nama tabel : User

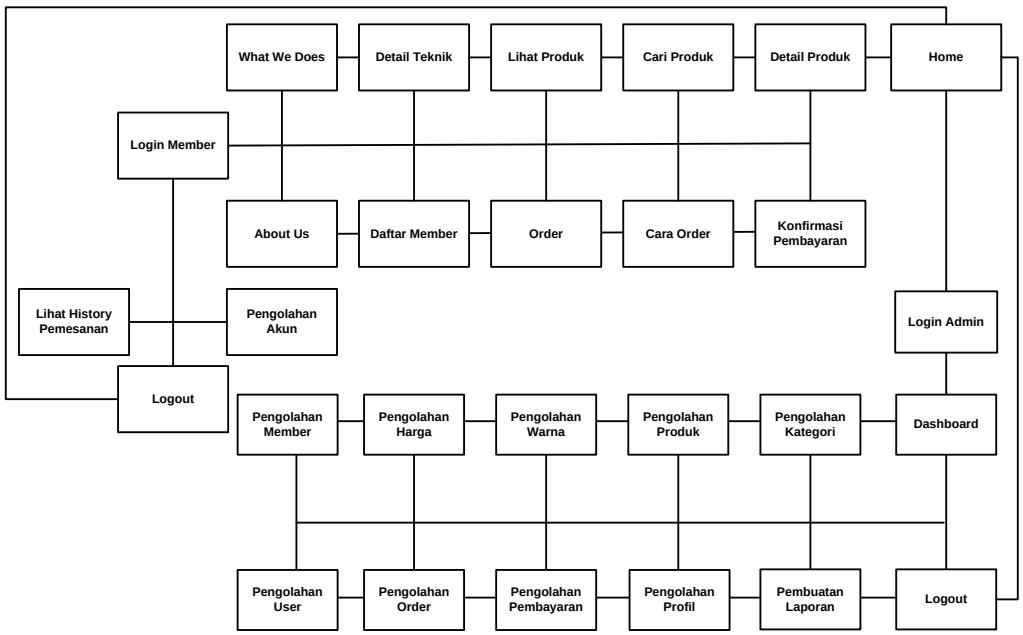
Primary Key : id_user

Tabel 3.107 Struktur Tabel User

Nama Field	Tipe	Ukuran	Keterangan
Id_user	INT	3	Primary Key
Username	Varchar	30	
Password	Varchar	51	
Email_user	Varchar	50	
Level	Varchar	20	

3.2.2 Perancangan Arsitektur Menu

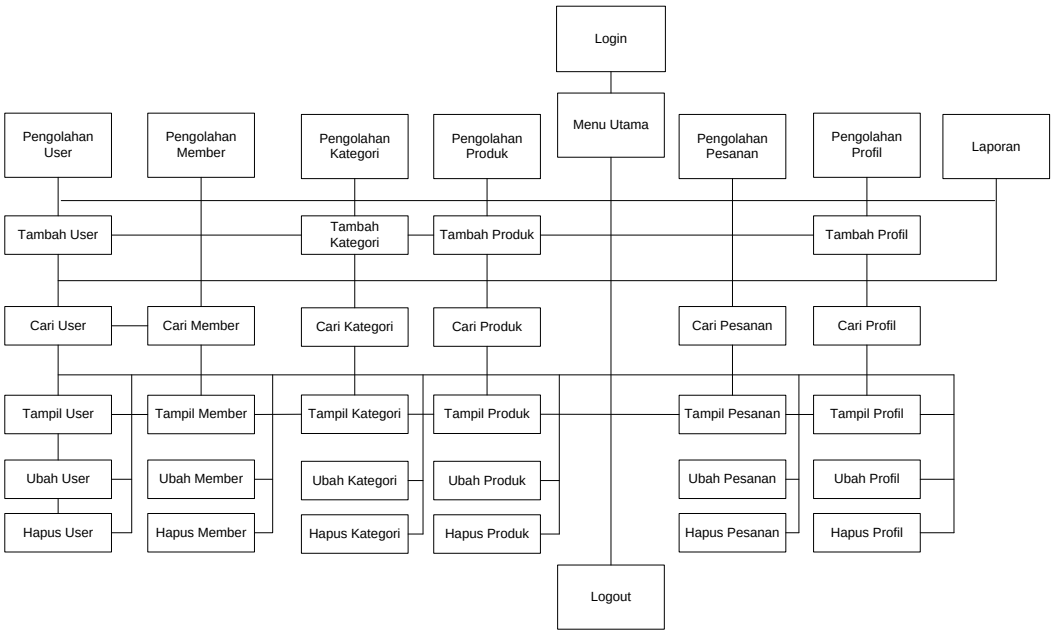
Perancangan struktur menu berisikan menu dan sub menu yang berfungsi untuk memudahkan *user* di dalam menggunakan sistem. Struktur menu *E-commerce* pada Pabrik Baju 12Titan dapat dilihat pada gambar 3.80.



Gambar 3.80 Struktur Menu *E-commerce* Pabrik Baju 12Titan

1. Struktur menu Admin

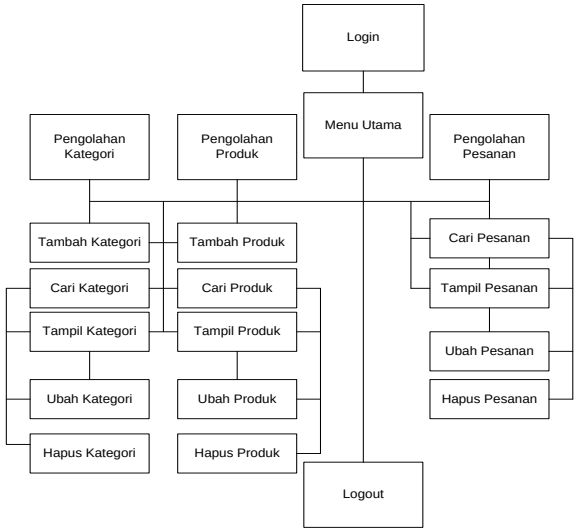
Perancangan struktur menu untuk admin dapat dilihat pada gambar 3.81.



Gambar 3.81 Struktur Menu Admin

2. Struktur Menu Operator

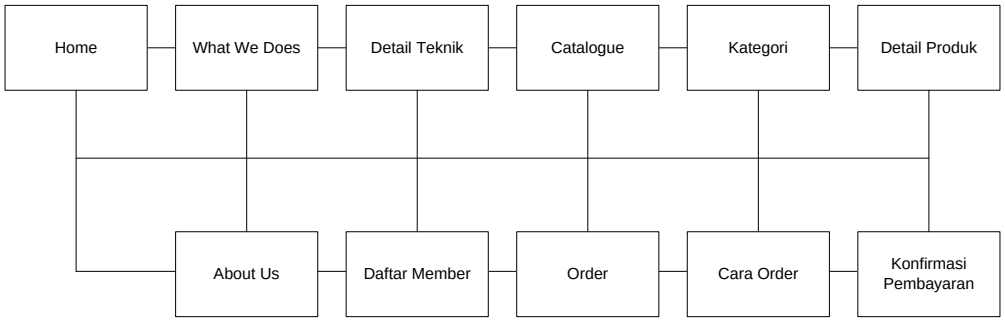
Perancangan struktur menu untuk admin dapat dilihat pada gambar 3.82.



Gambar 3.82 Struktur Menu Operator

3. Struktur Menu Pengunjung

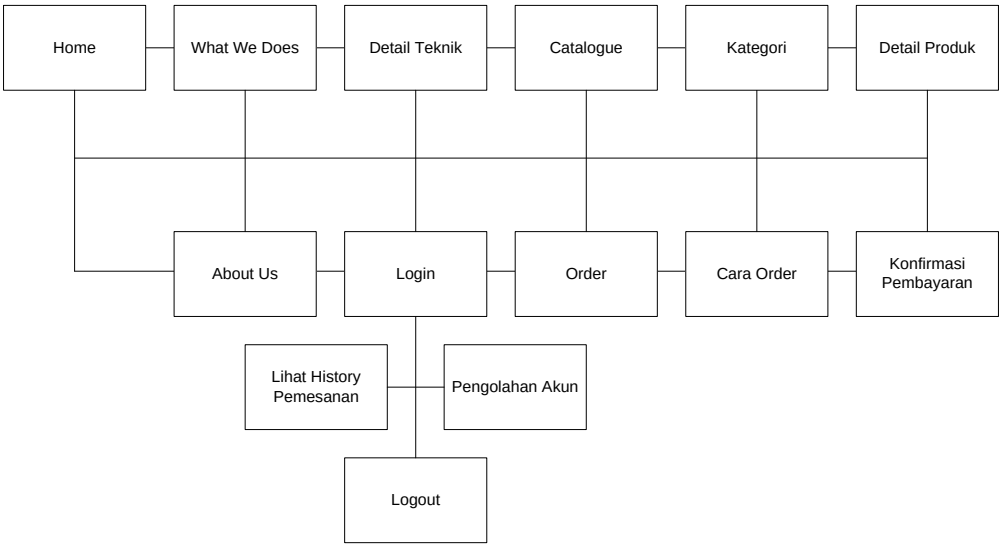
Perancangan struktur menu untuk pengunjung dapat dilihat pada gambar 3.83.



Gambar 3.83 Struktur Menu Pengunjung

4. Struktur Menu Member

Perancangan struktur menu untuk member dapat dilihat pada gambar 3.84.



Gambar 3.84 Struktur Menu Member

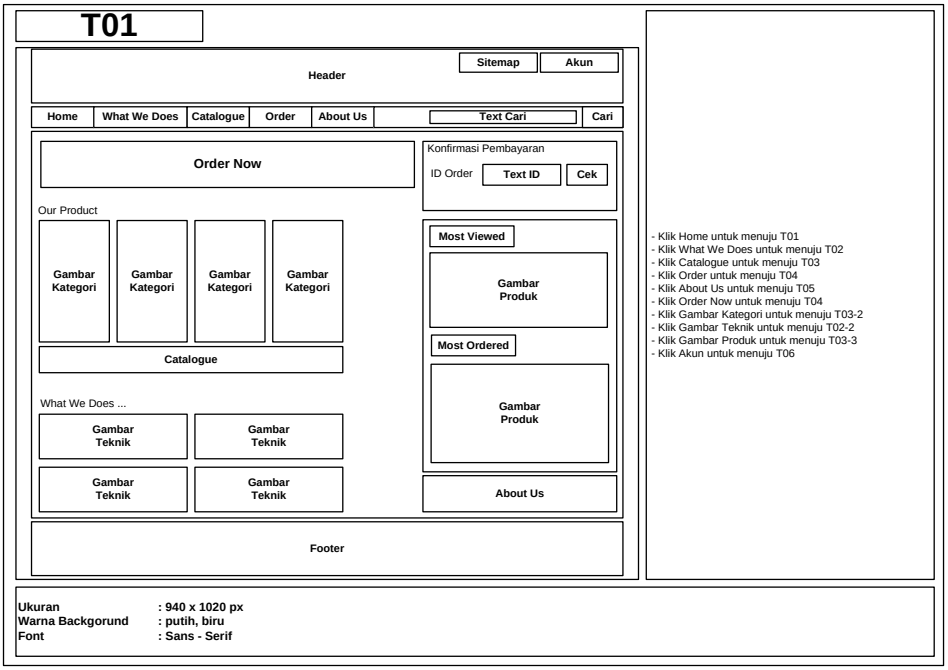
3.2.3 Perancangan Antarmuka

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.*

3.2.3.1 Perancangan Tampilan

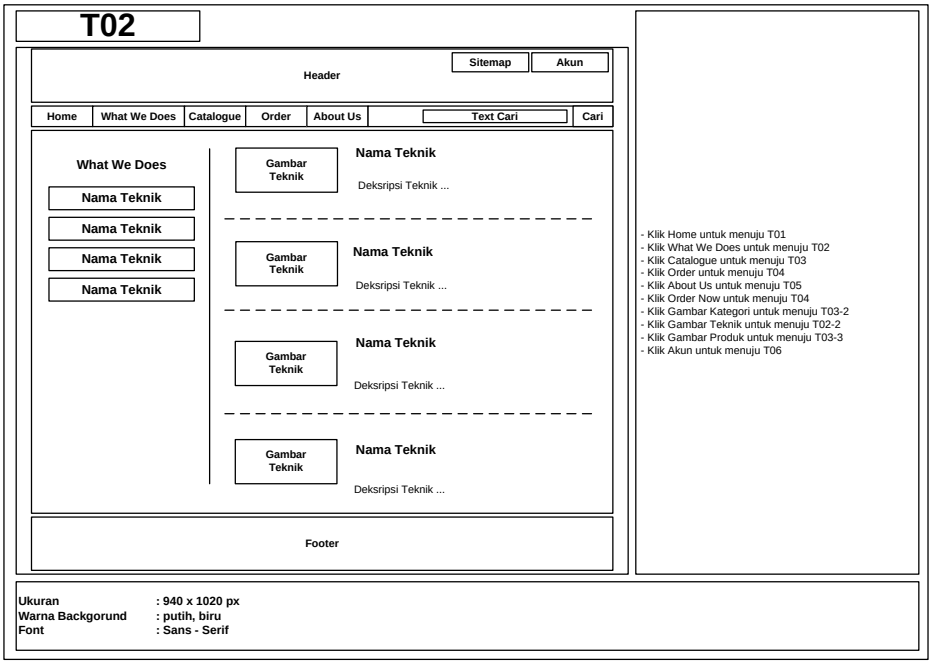
Perancangan tampilan bagi pengunjung adalah sebagai berikut :

1. Perancangan Halaman Home (T01)



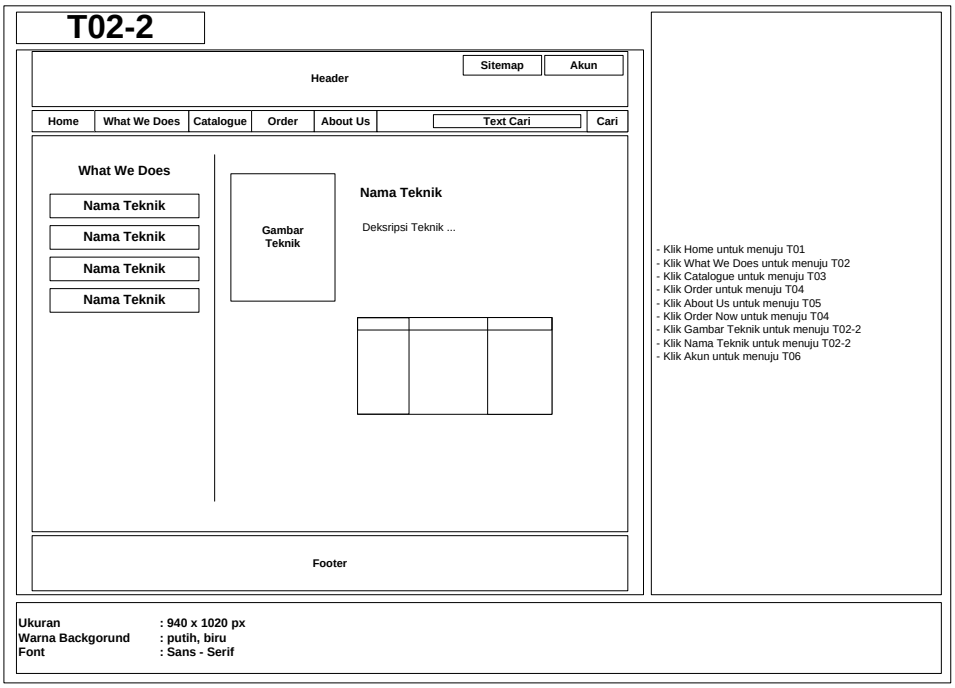
Gambar 3.85 Perancangan Tampilan Halaman Home

2. Perancangan Halaman What We Does



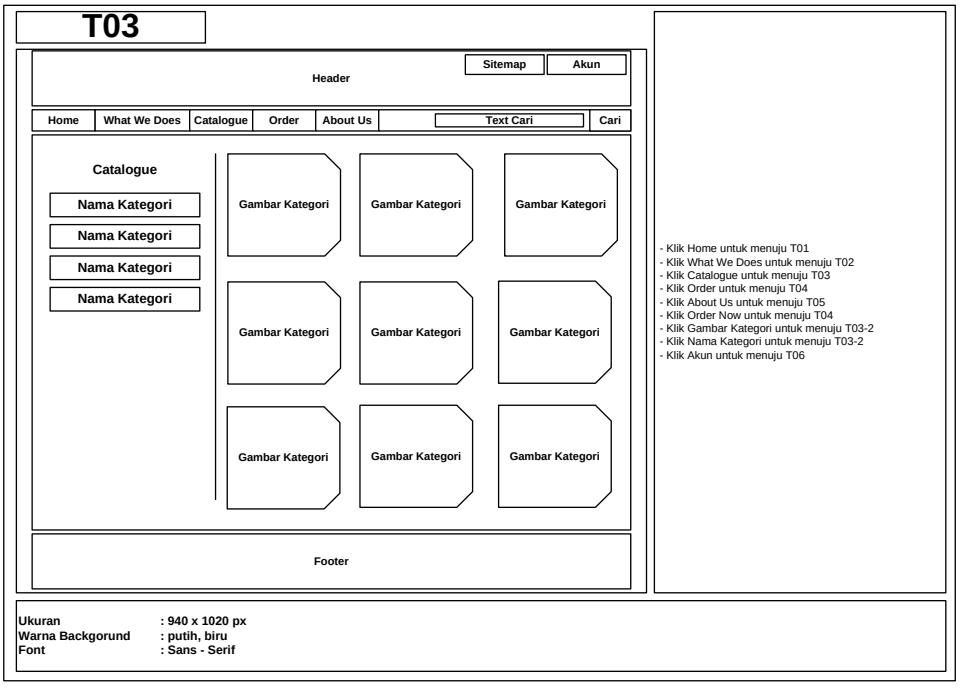
Gambar 3.86 Perancangan Tampilan What We Does

3. Perancangan Halaman Detail Teknik



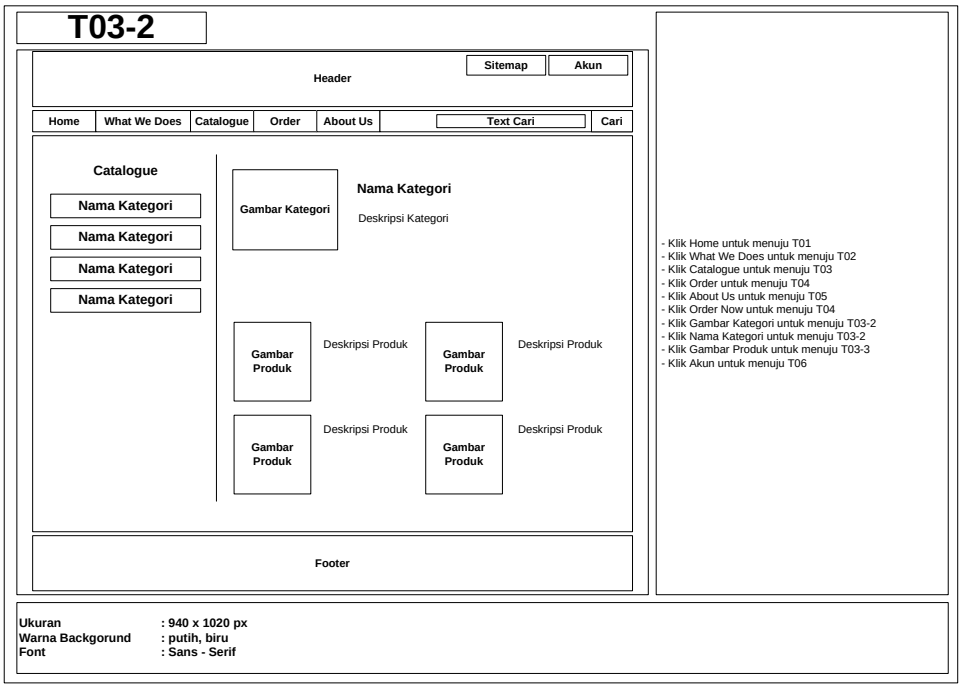
Gambar 3.87 Perancangan Halaman Detail Teknik

4. Perancangan Halaman Catalogue



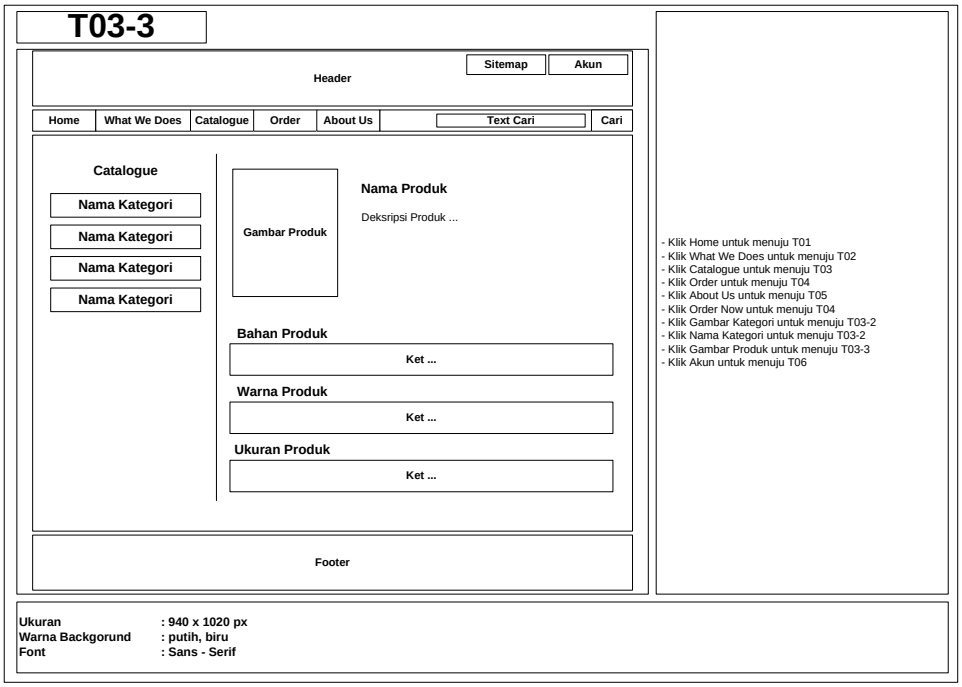
Gambar 3.88 Perancangan Halaman Catalogue

5. Perancangan Detail Kategori



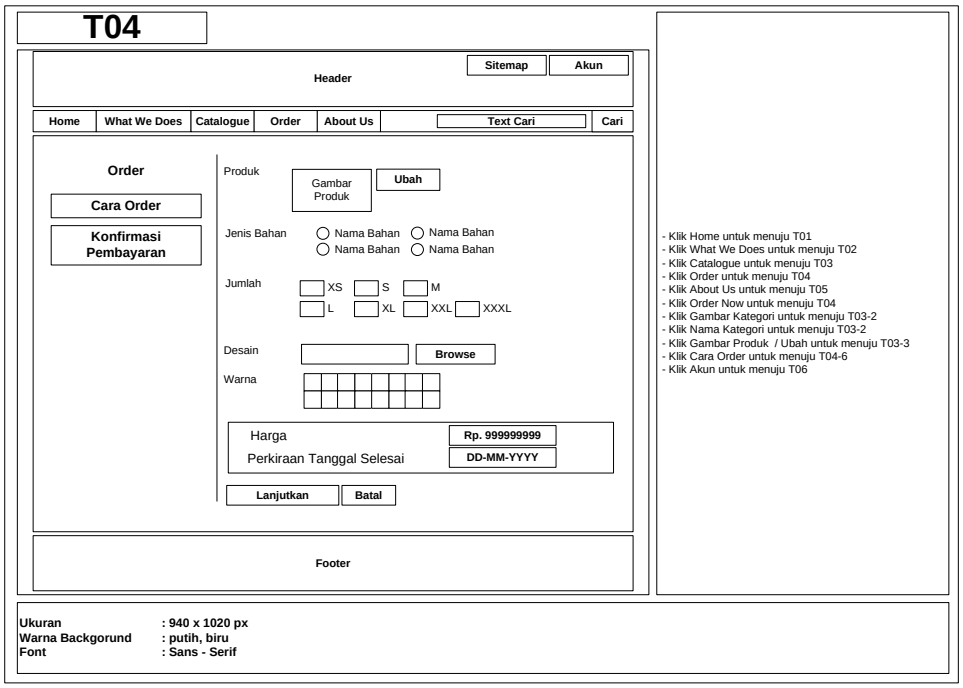
Gambar 3.89 Perancangan Detail Kategori

6. Perancangan Halaman Detail Produk



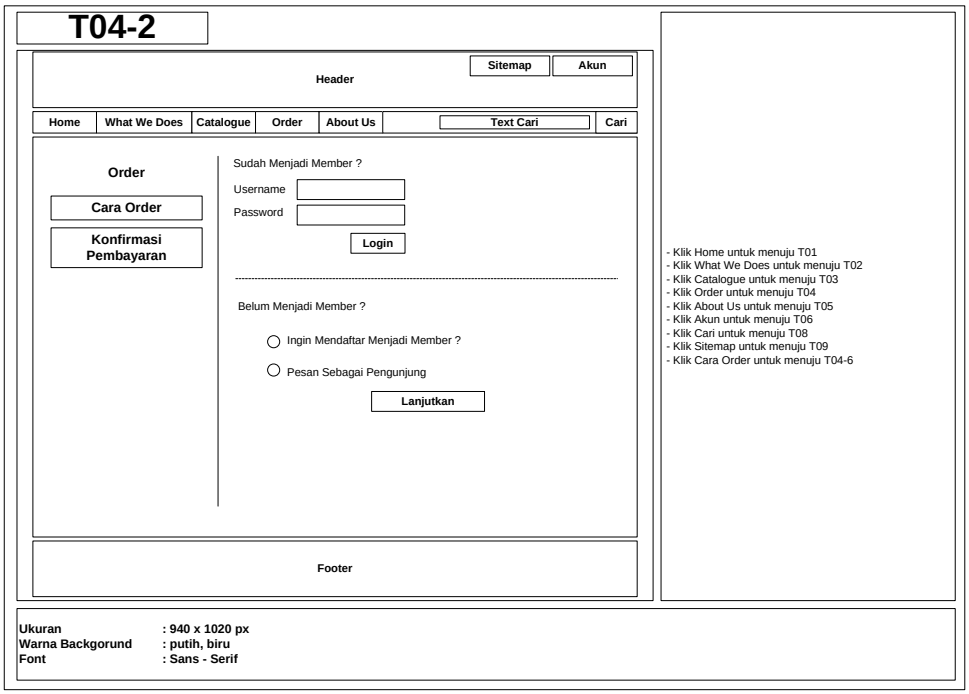
Gambar 3.90 Perancangan Halaman Detail Produk

7. Perancangan Halaman Order



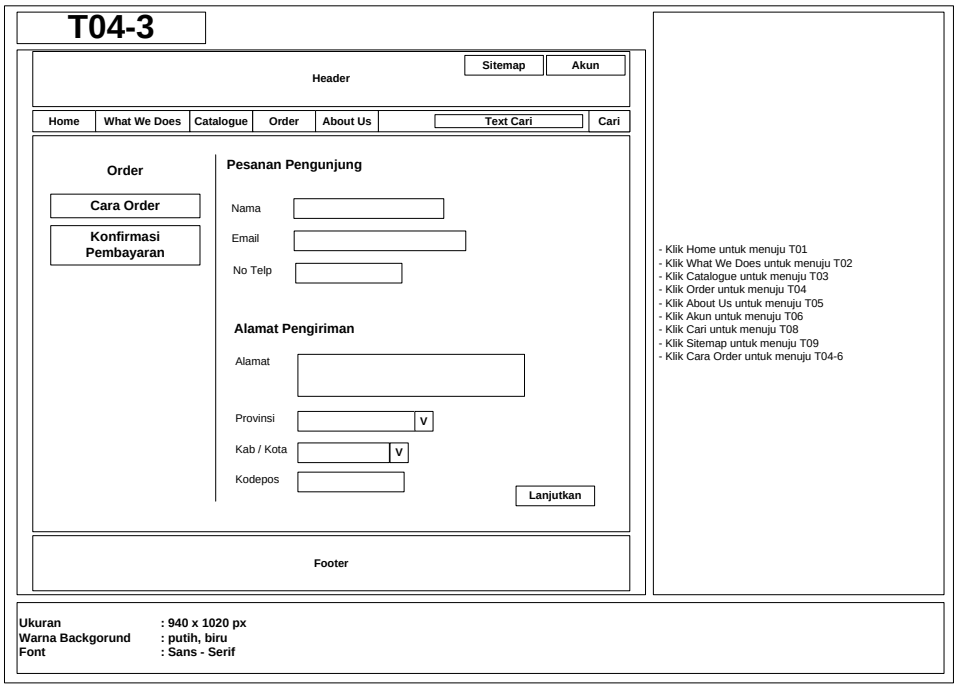
Gambar 3.91 Perancangan Tampilan Order

8. Perancangan Halaman Order Langkah 2



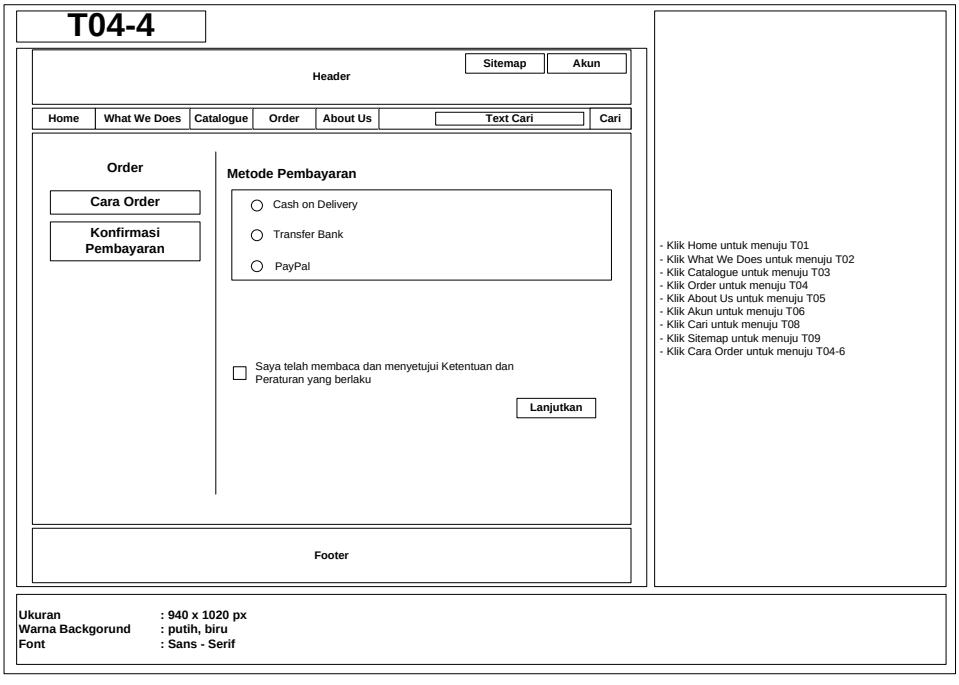
Gambar 3.92 Perancangan Halaman Order Langkah 2

9. Perancangan Halaman Order Langkah 3



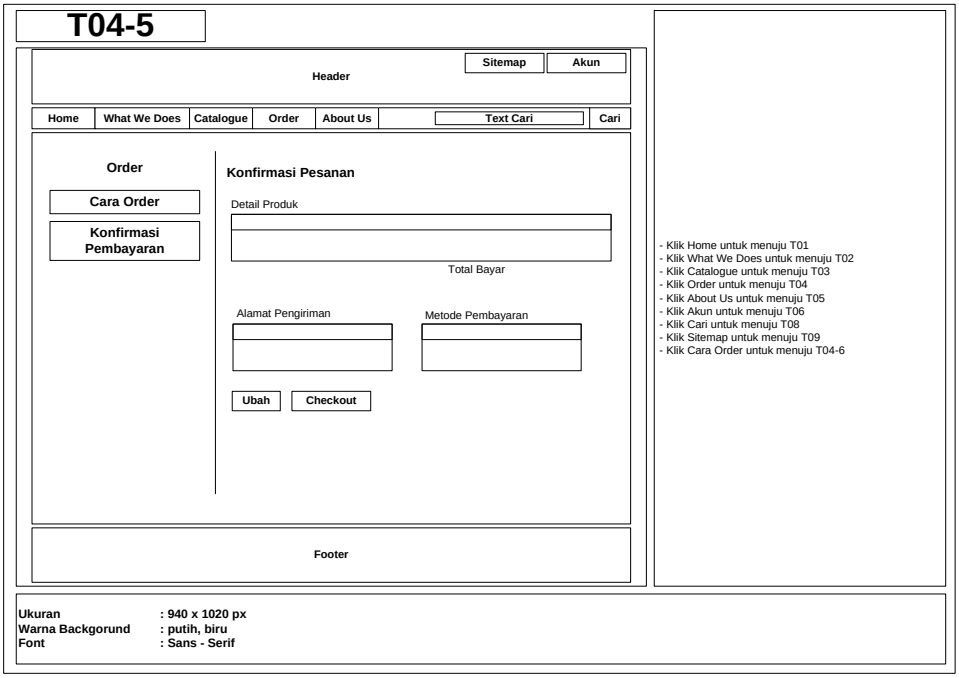
Gambar 3.93 Perancangan Halaman Order Langkah 3

10. Perancangan Halaman Order Langkah 4



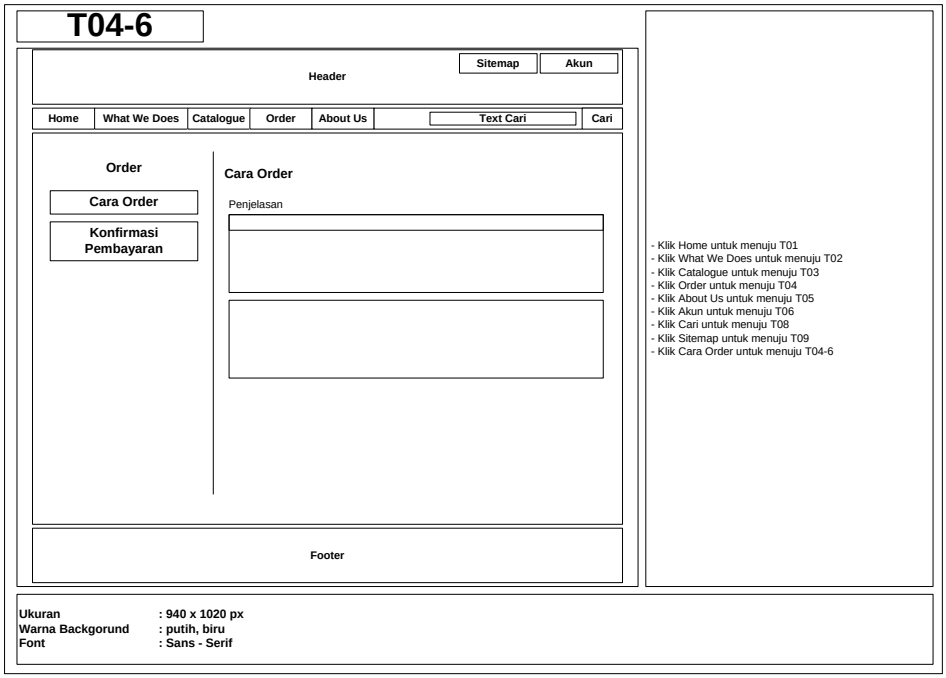
Gambar 3.94 Perancangan Halaman Order Langkah 4

11. Perancangan Halaman Order Langkah 5



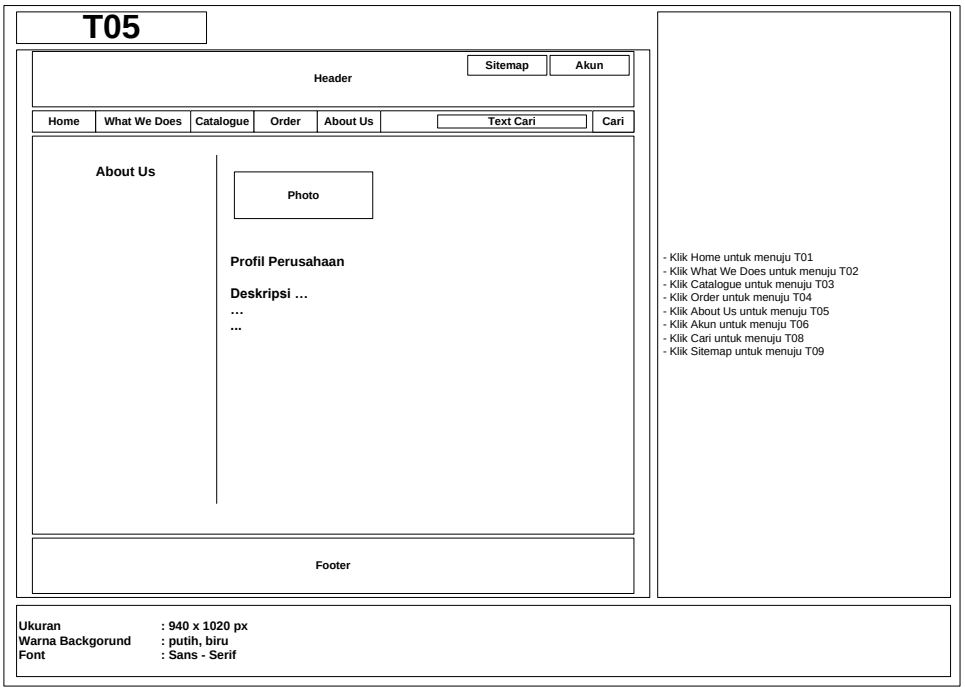
Gambar 3.95 Perancangan Halaman Order Langkah 5

12. Perancangan Halaman Cara Order



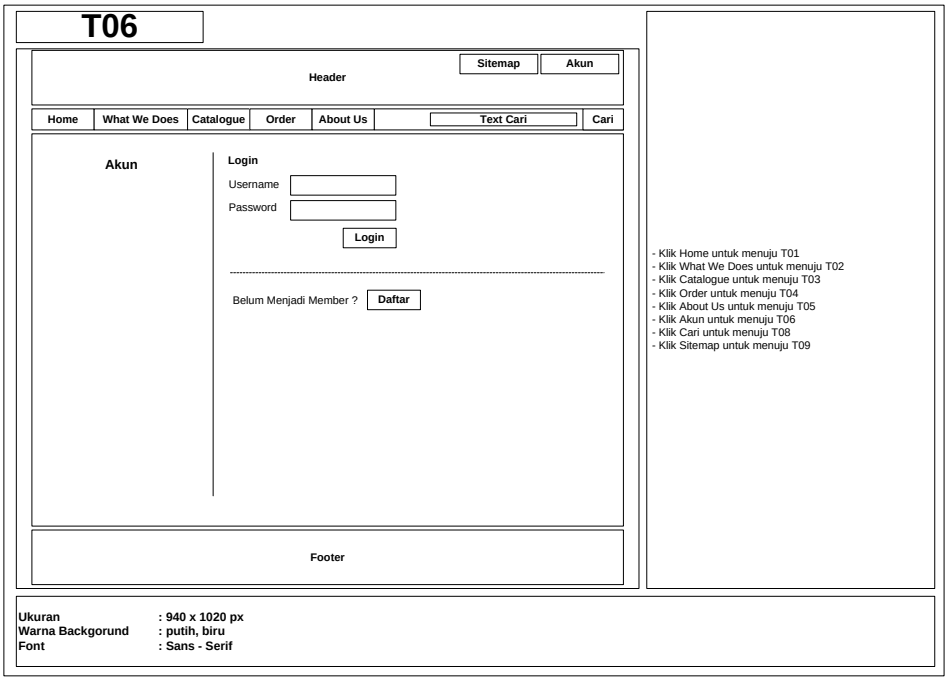
Gambar 3.96 Perancangan Halaman Cara Order

13. Perancangan Halaman About Us



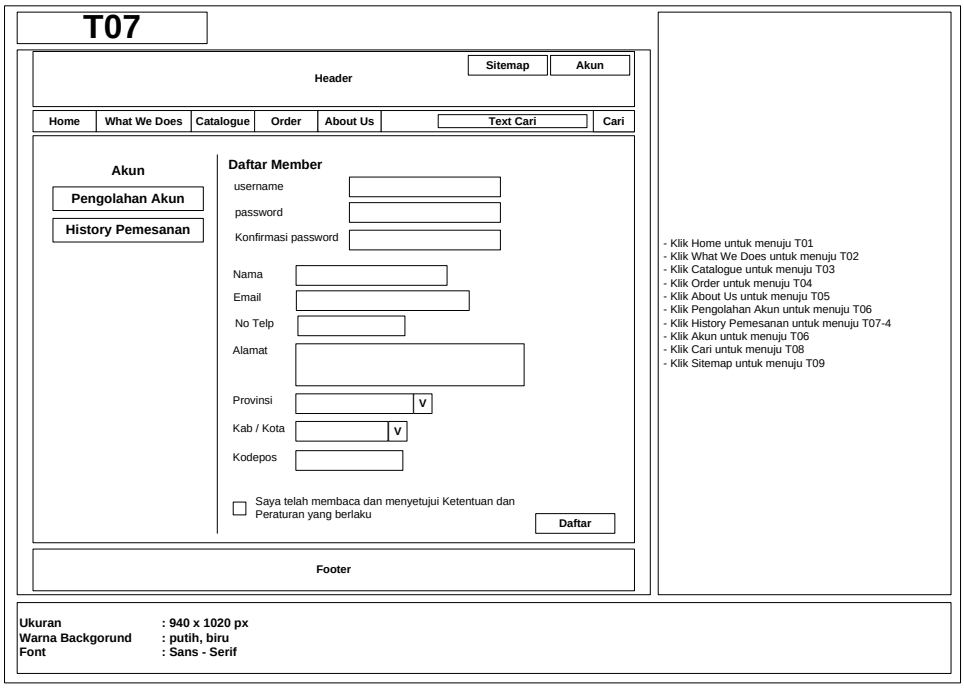
Gambar 3.97 Perancangan Tampilan About Us

14. Perancangan Halaman Login Akun Member



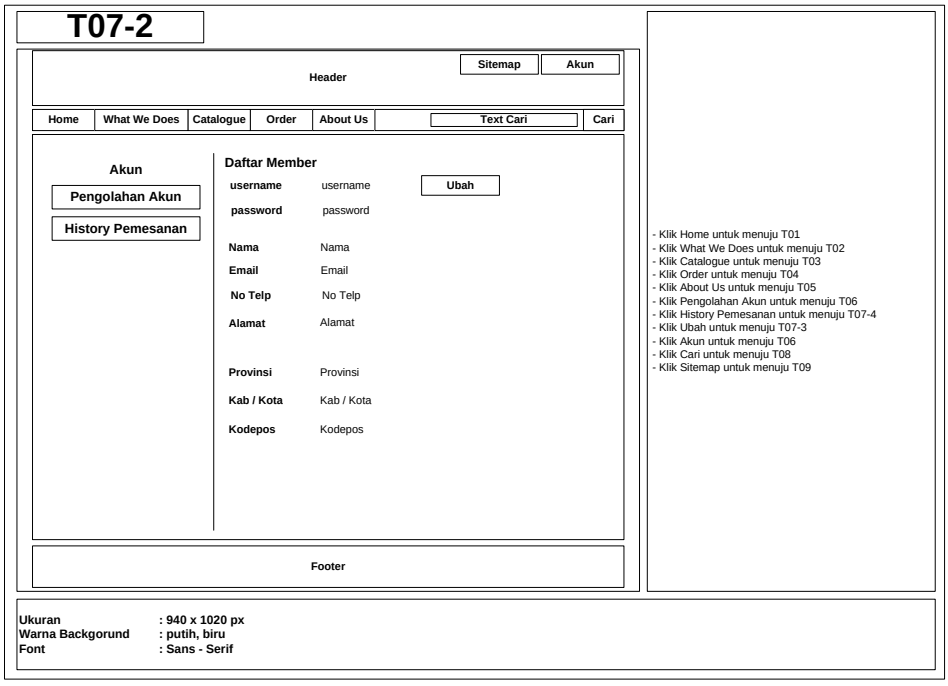
Gambar 3.98 Perancangan Halaman Login Akun Member

15. Perancangan Halaman Daftar Member



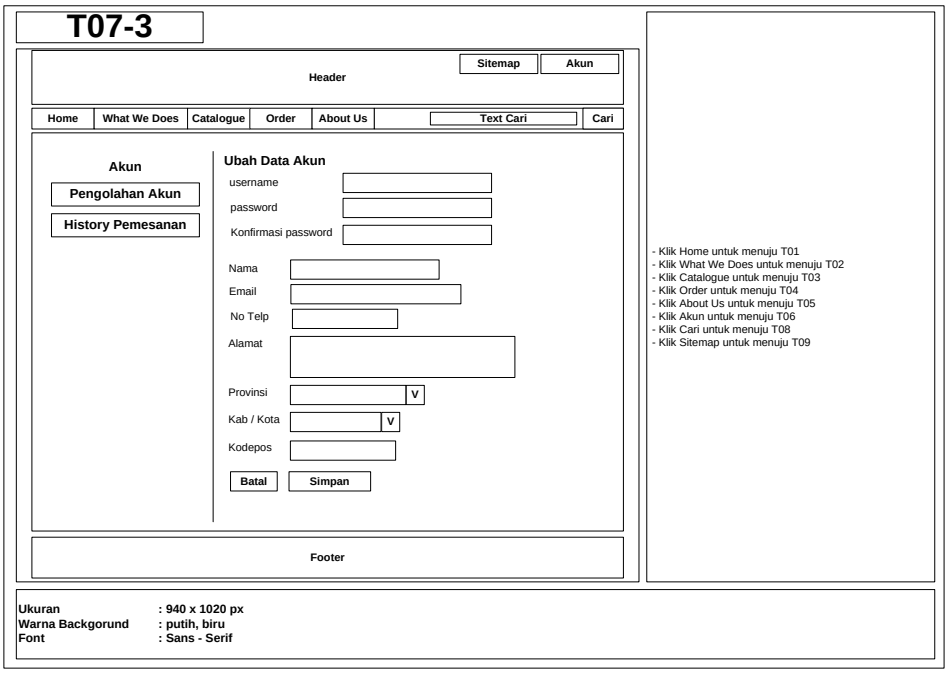
Gambar 3.99 Perancangan Halaman Daftar Member

16. Perancangan Halaman Pengolahan Data Akun



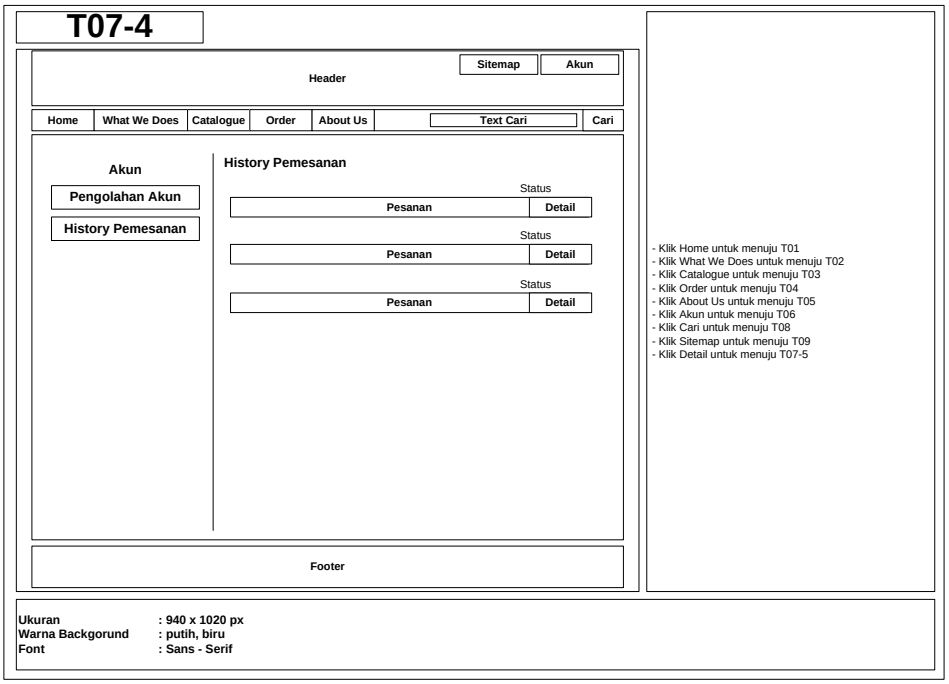
Gambar 3.100 Perancangan Halaman Pengolahan Data Akun

17. Perancangan Halaman Ubah Data Akun



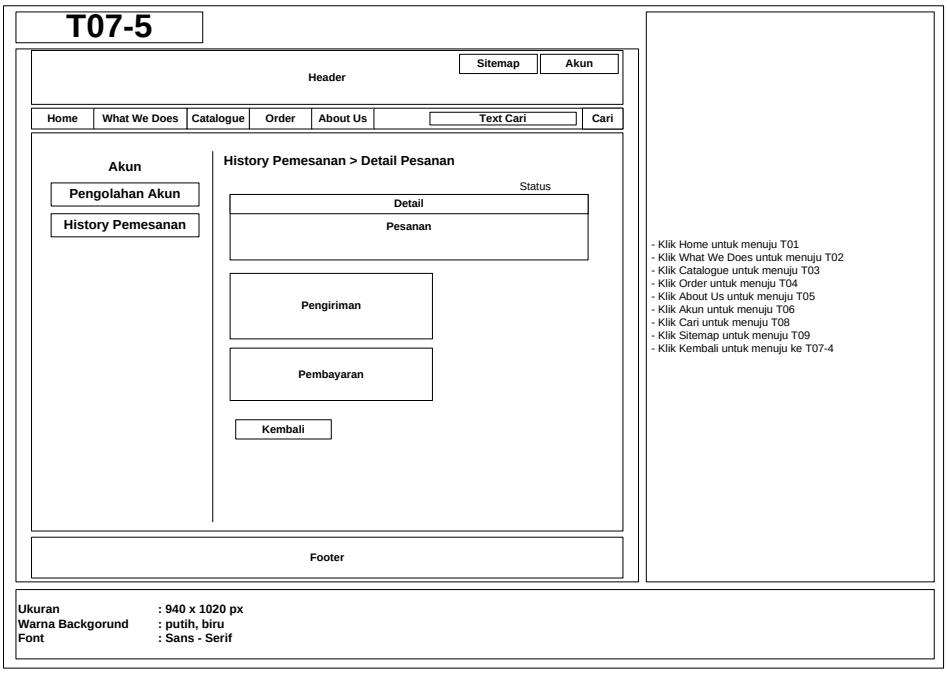
Gambar 3.101 Perancangan Halaman Ubah Data Akun

18. Perancangan Halaman History Pemesanan



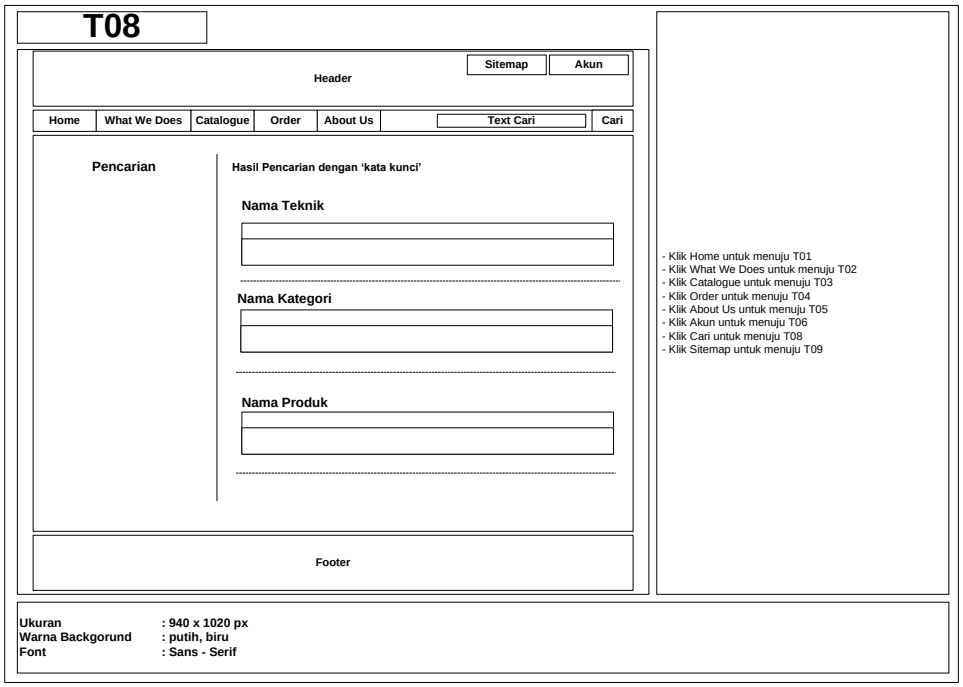
Gambar 3.102 Perancangan Halaman History Pemesanan

19. Perancangan Halaman Detail History Pemesanan



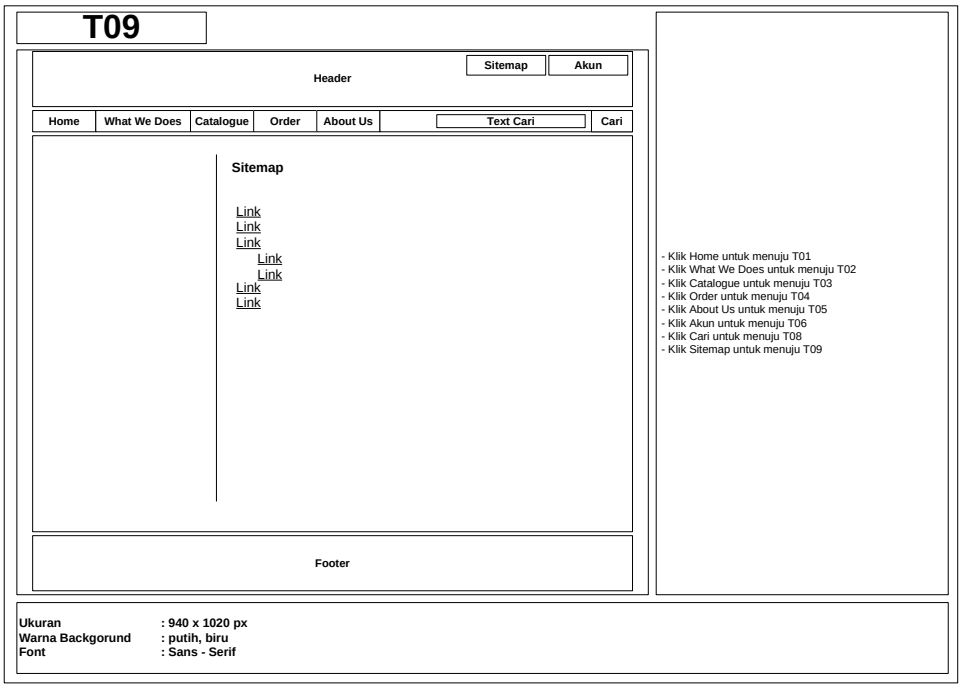
Gambar 3.103 Perancangan Halaman Detail History Pemesanan

20. Perancangan Halaman Pencarian



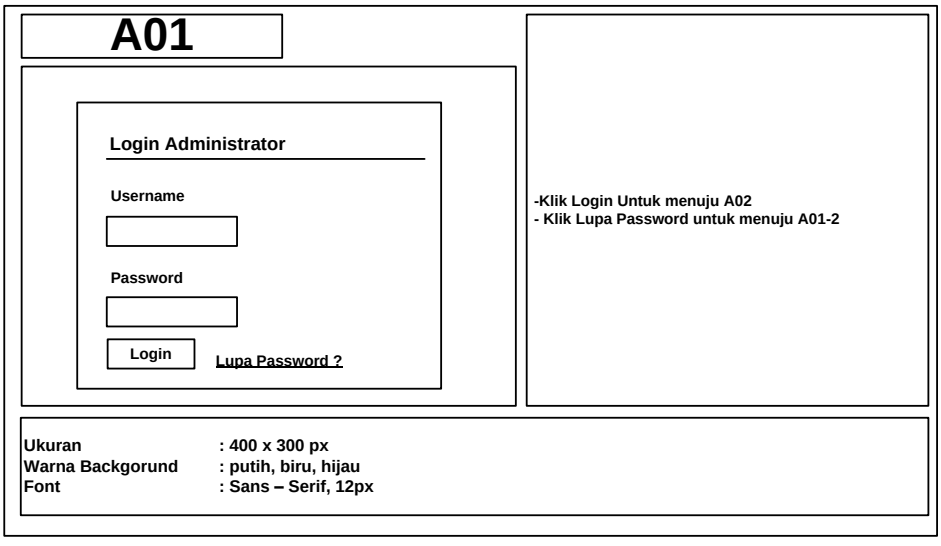
Gambar 3.104 Perancangan Halaman Pencarian

21. Perancangan Halaman Sitemap



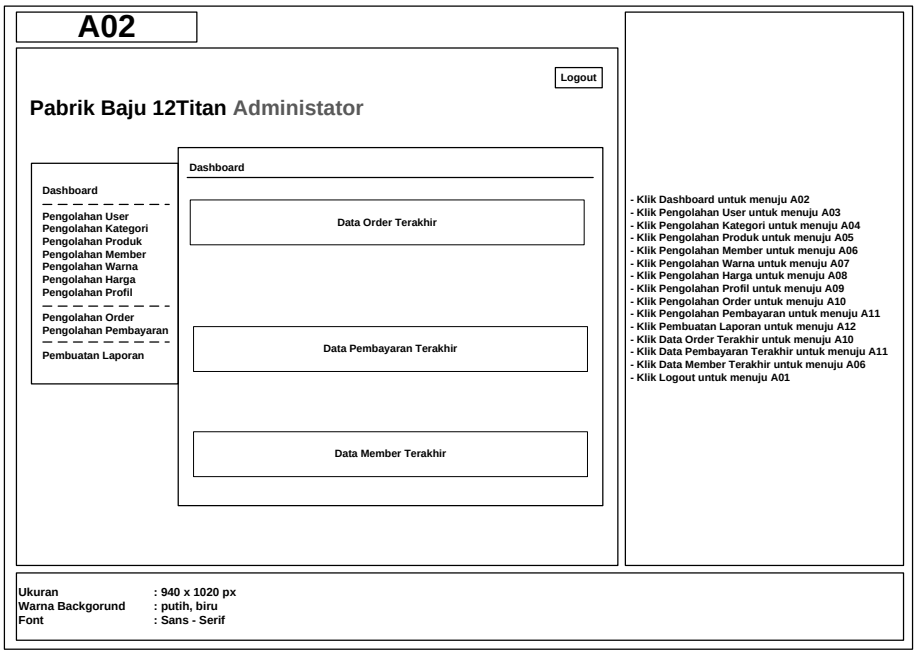
Gambar 3.105 Perancangan Halaman Sitemap

22. Perancangan Halaman Login Admin dan Operator



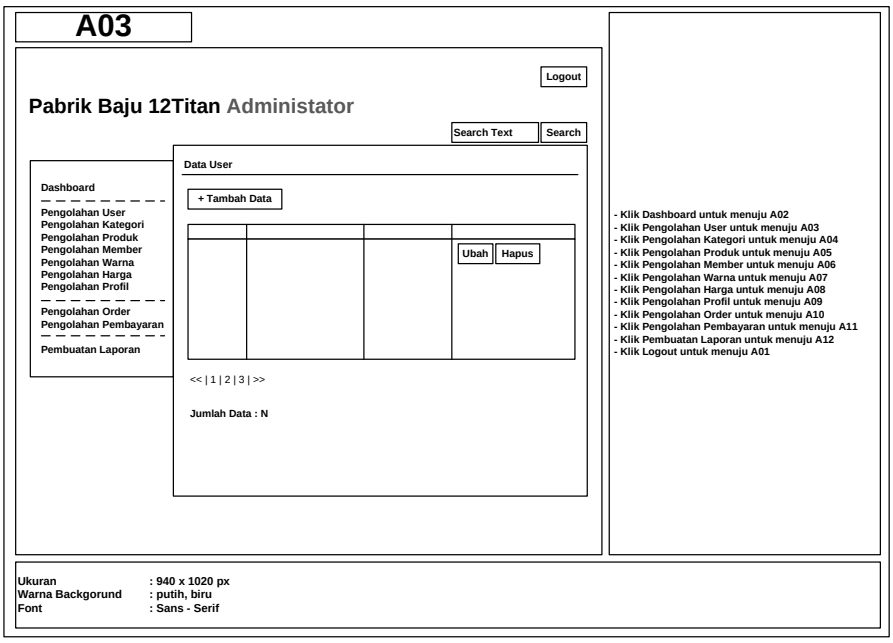
Gambar 3.106 Perancangan Halaman Login Admin dan Operator

23. Perancangan Halaman Dashboard



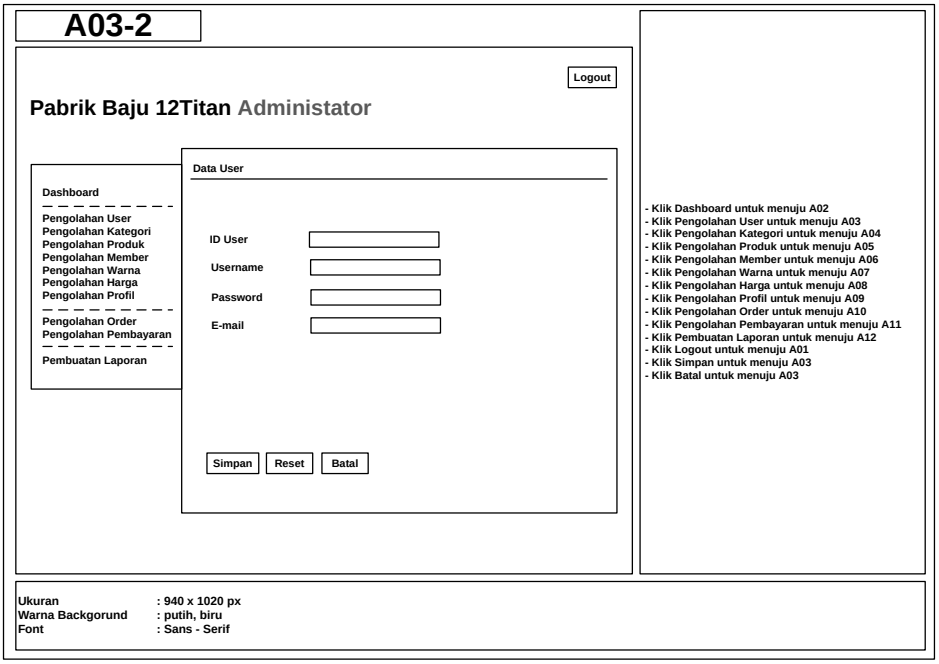
Gambar 3.107 Perancangan Halaman Dashboard

24. Perancangan Halaman Pengolahan User



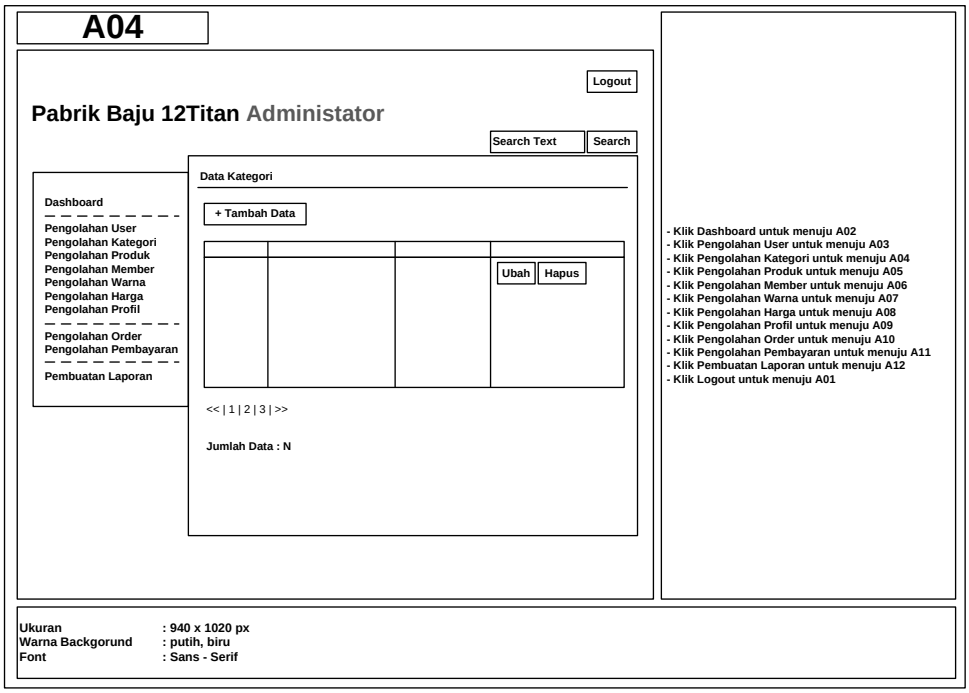
Gambar 3.108 Perancangan Halaman Pengolahan User

25. Perancangan Halaman Form User



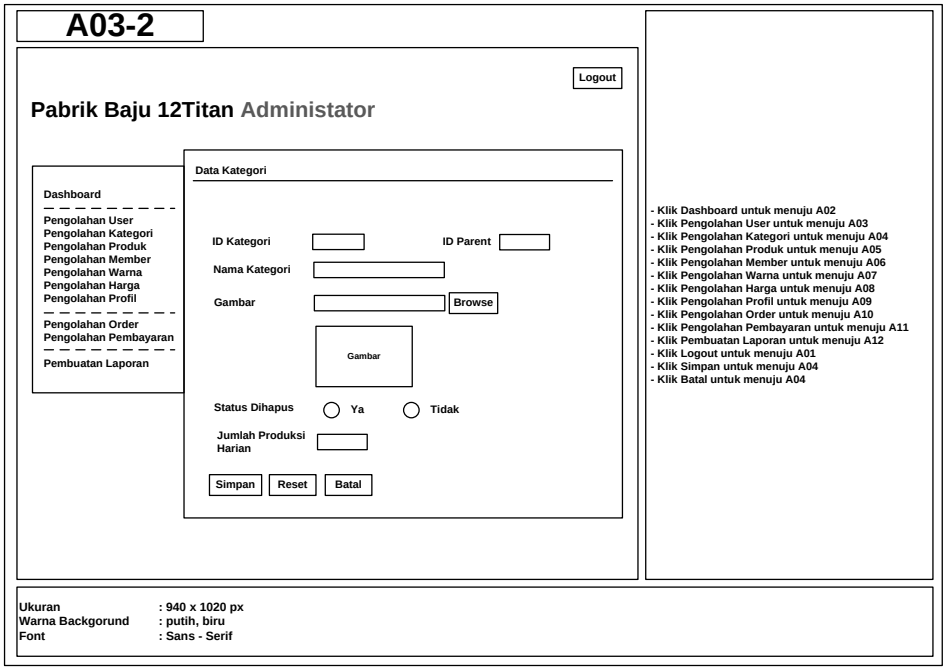
Gambar 3.109 Perancangan Halaman Form User

26. Perancangan Halaman Pengolahan Kategori



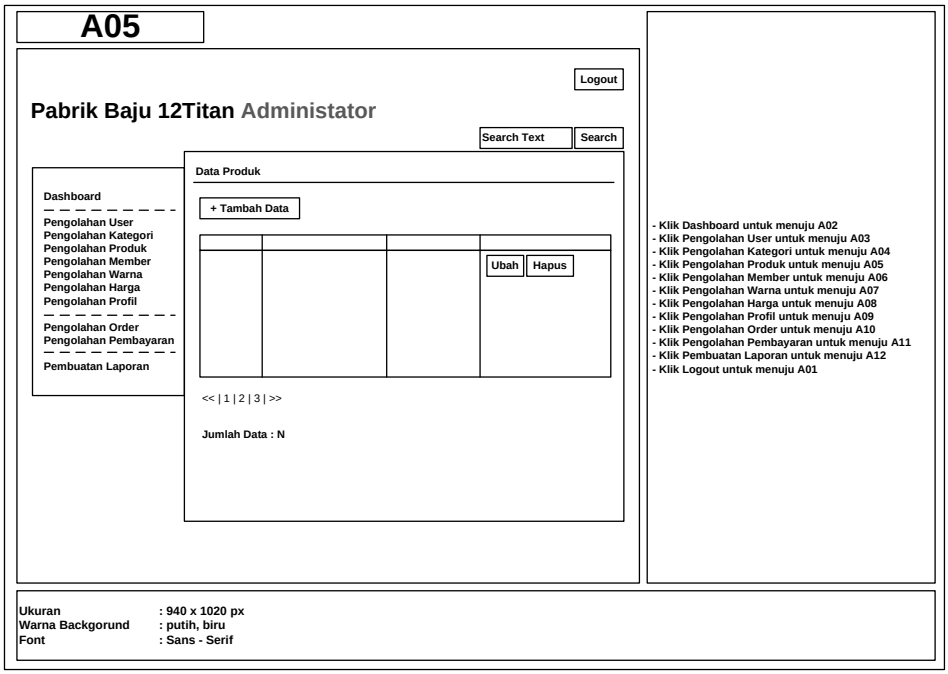
Gambar 3.110 Perancangan Halaman Pengolahan Kategori

27. Perancangan Halaman Form Kategori



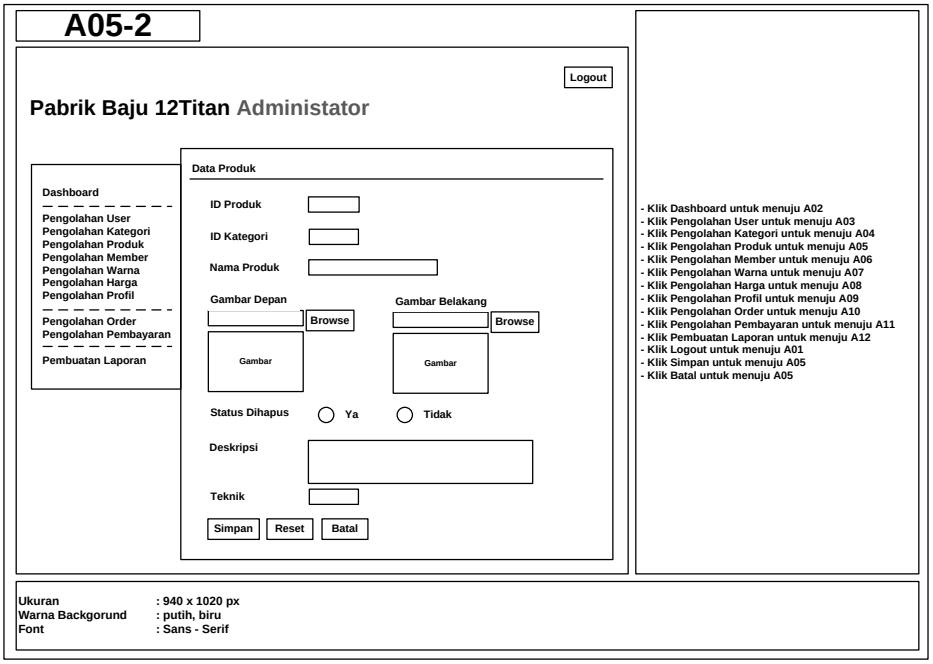
Gambar 3.111 Perancangan Halaman Form Kategori

28. Perancangan Halaman Pengolahan Produk



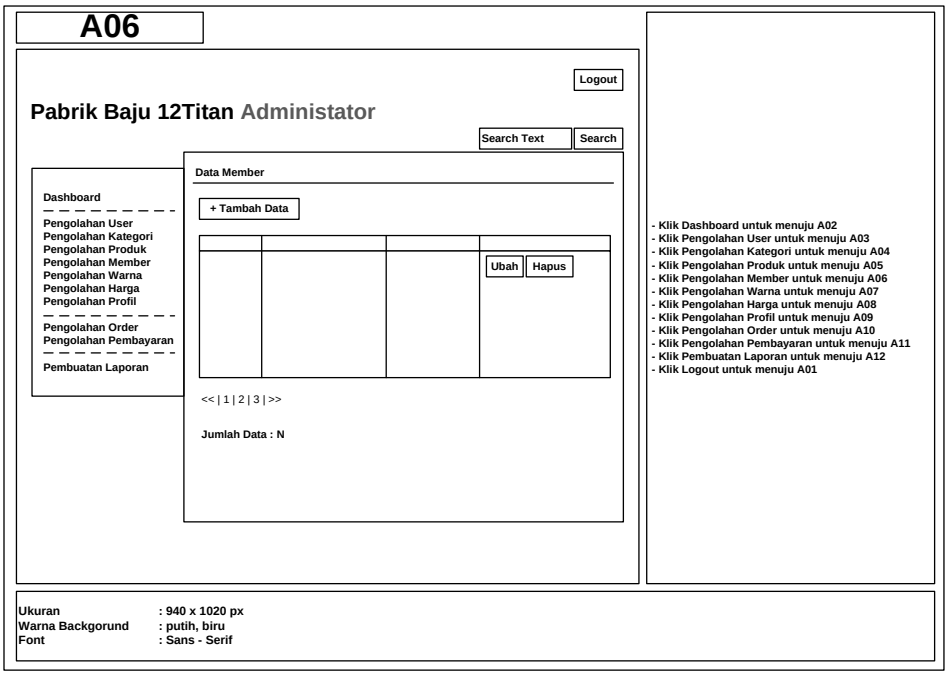
Gambar 3.112 Perancangan Halaman Pengolahan Produk

29. Perancangan Halaman Form Produk



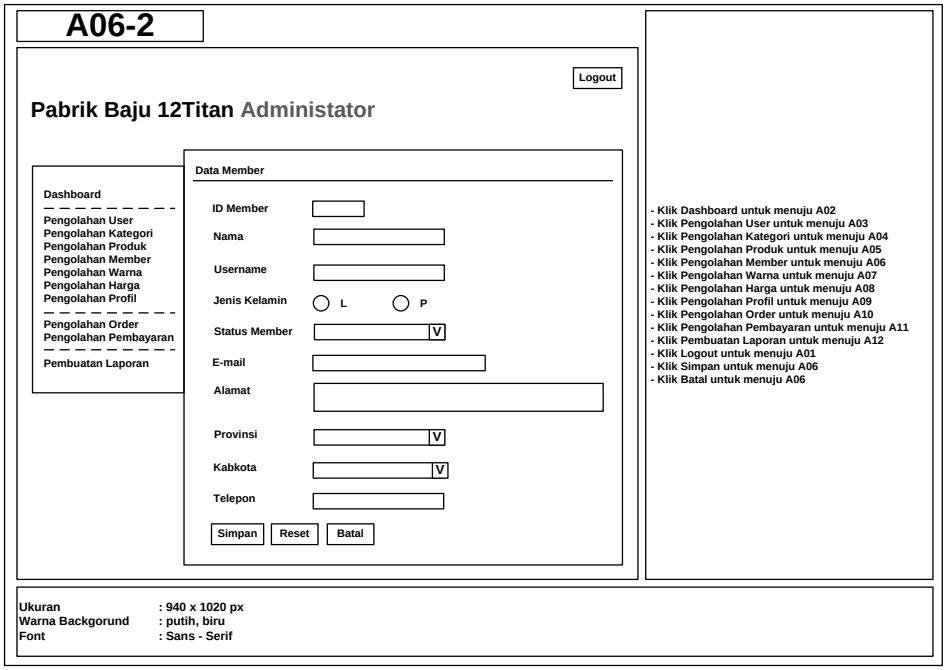
Gambar 3.113 Perancangan Halaman Form Produk

30. Perancangan Halaman Pengolahan Member



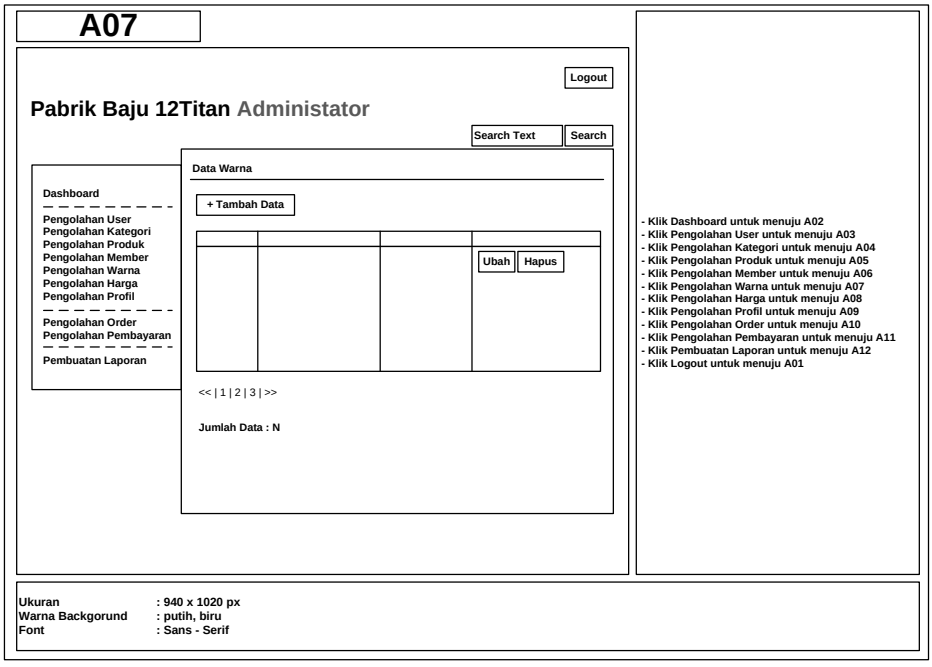
Gambar 3.114 Perancangan Halaman Pengolahan Member

31. Perancangan Halaman Form Member



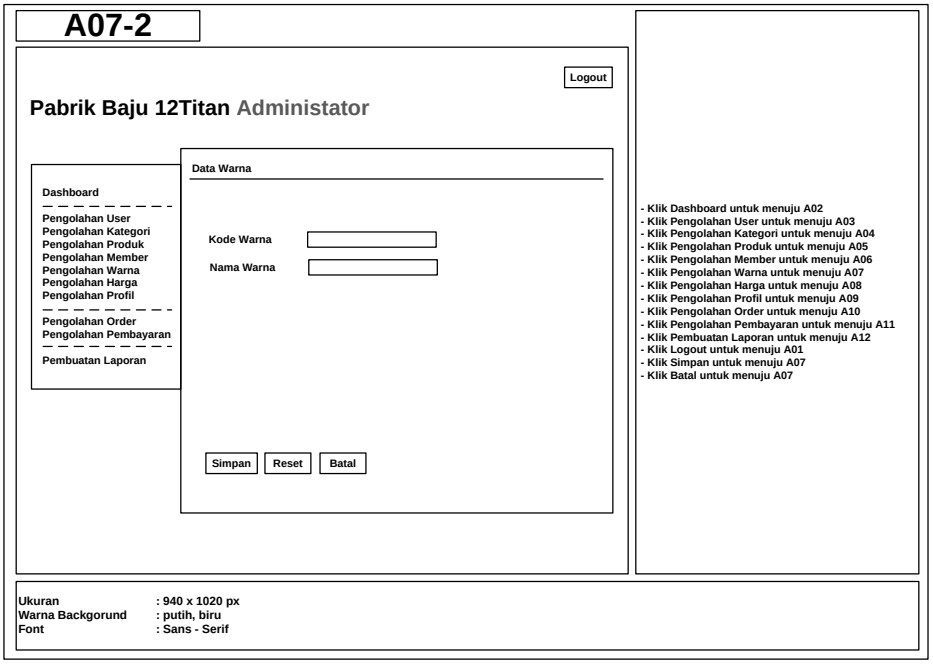
Gambar 3.115 Perancangan Halaman Form Member

32. Perancangan Halaman Pengolahan Warna



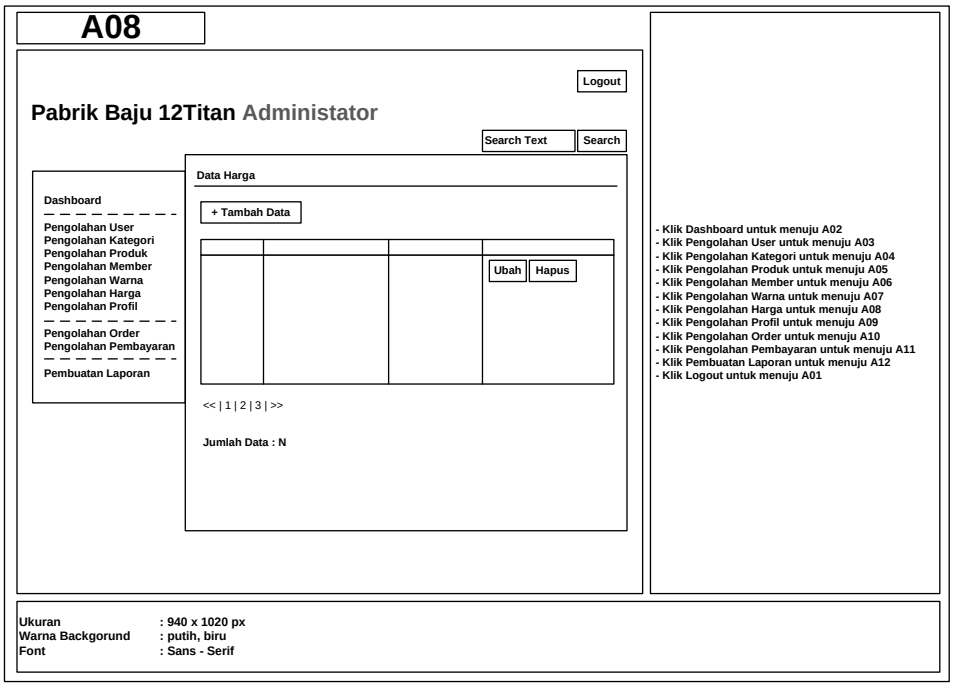
Gambar 3.116 Perancangan Halaman Pengolahan Warna

33. Perancangan Halaman Form Warna



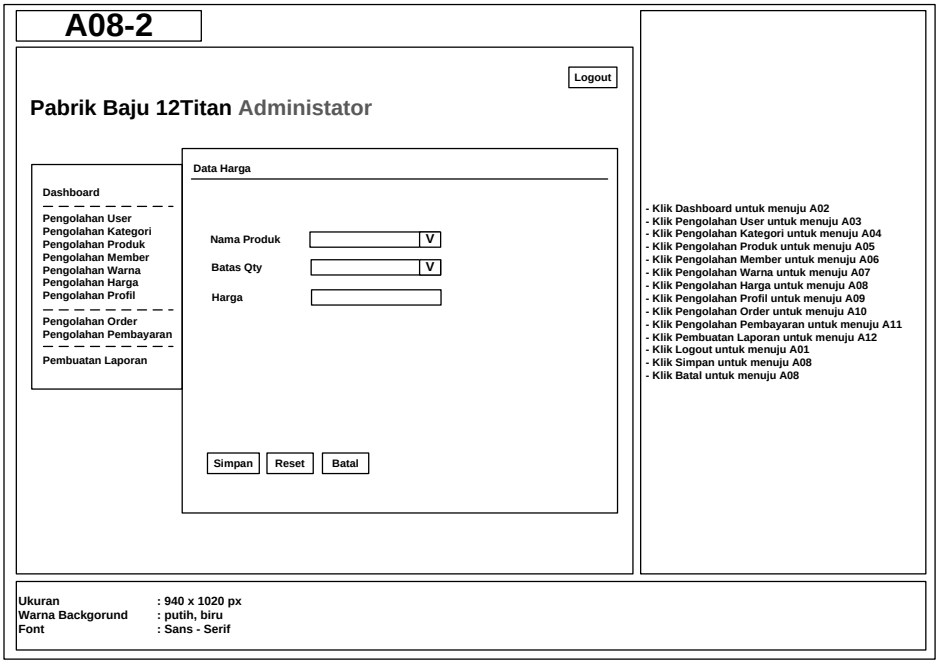
Gambar 3.117 Perancangan Halaman Form Warna

34. Perancangan Halaman Pengolahan Harga



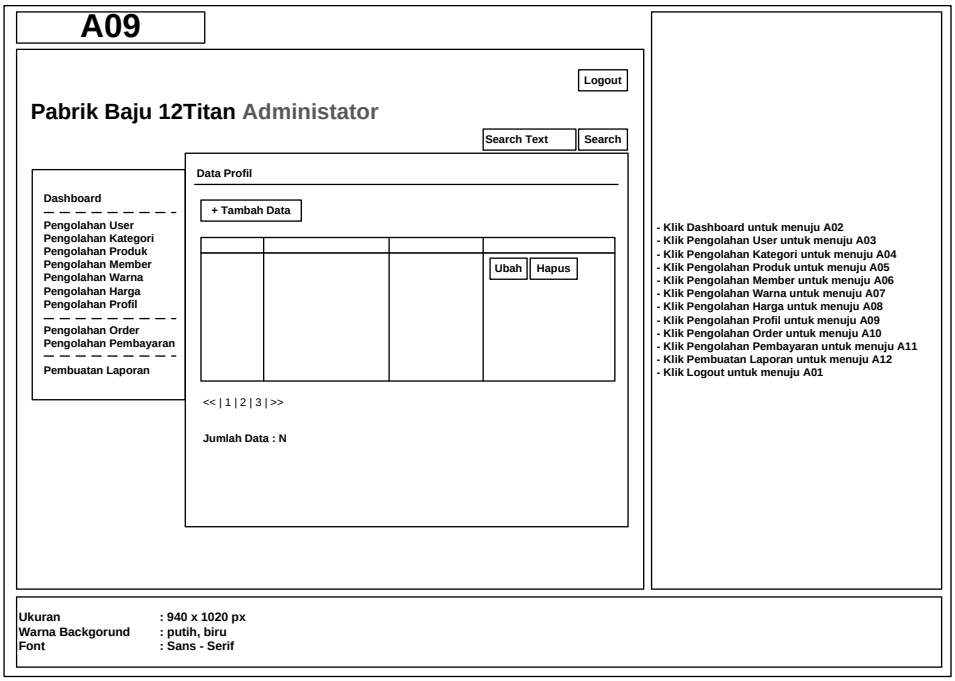
Gambar 3.118 Perancangan Halaman Pengolahan Harga

35. Perancangan Halaman Form Harga



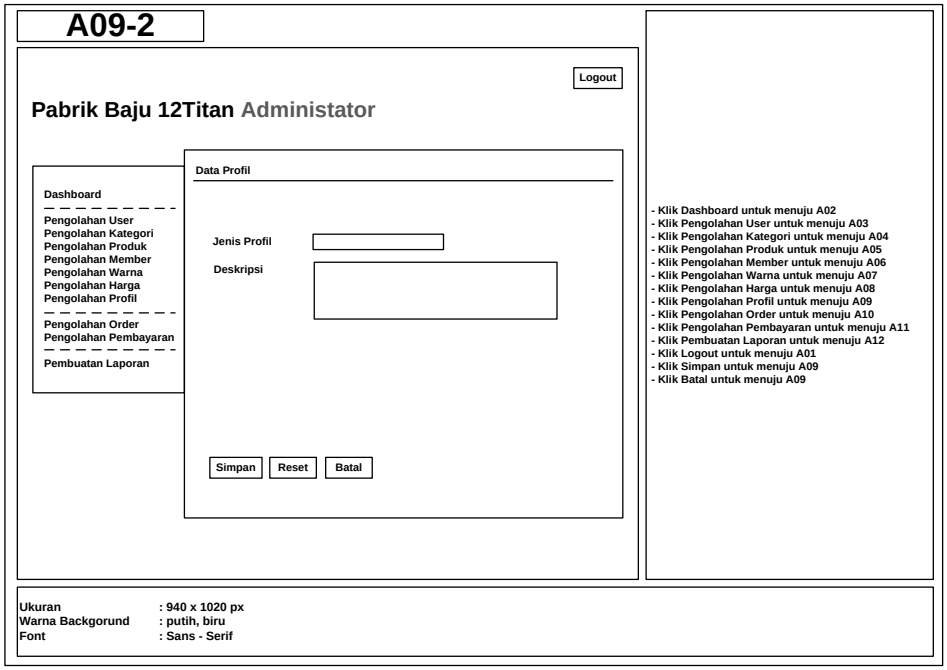
Gambar 3.119 Perancangan Halaman Form Harga

36. Perancangan Halaman Pengolahan Profil



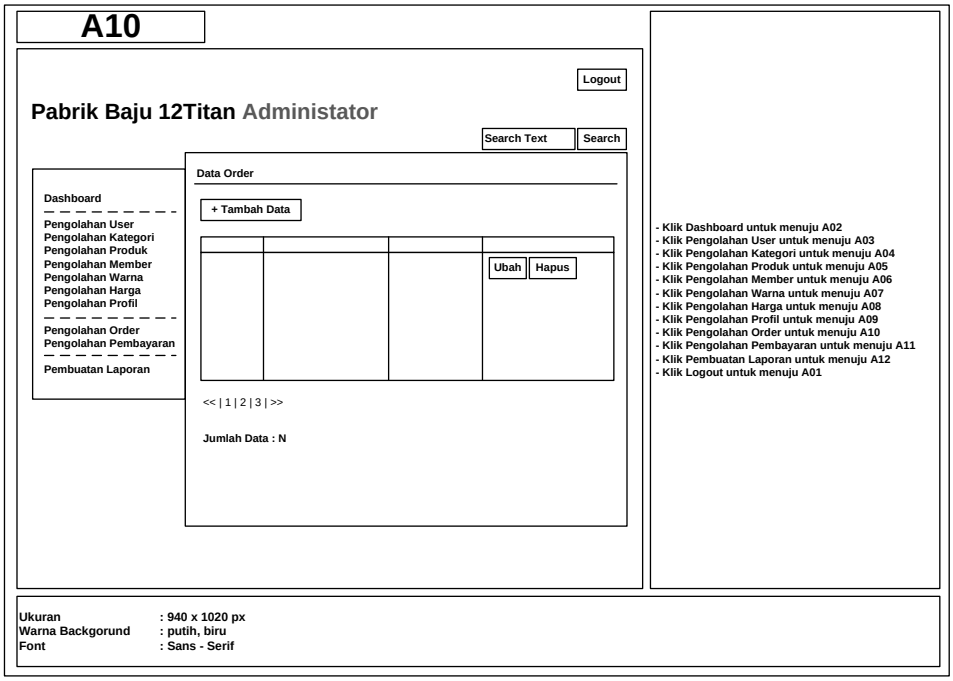
Gambar 3.120 Perancangan Halaman Pengolahan Profil

37. Perancangan Halaman Form Profil



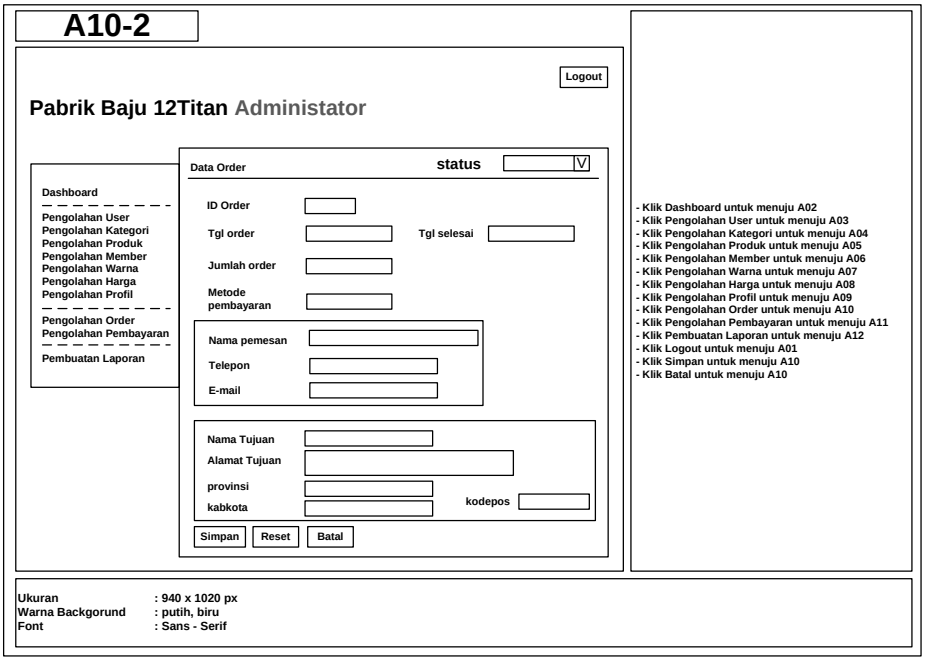
Gambar 3.121 Perancangan Halaman Form Profil

38. Perancangan Halaman Pengolahan Order



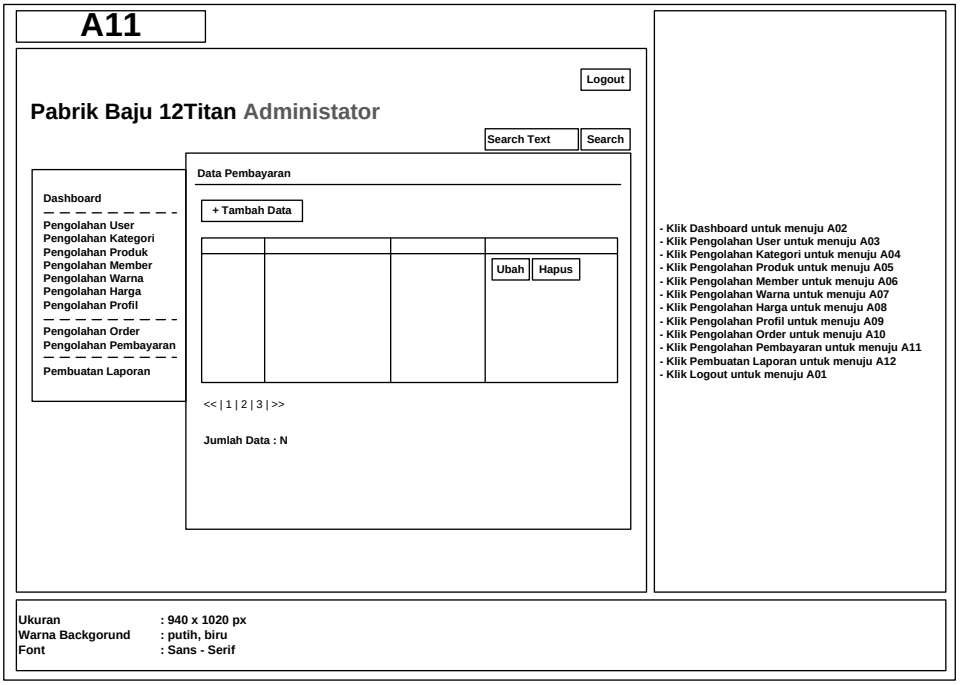
Gambar 3.122 Perancangan Halaman Pengolahan Order

39. Perancangan Halaman Form Order



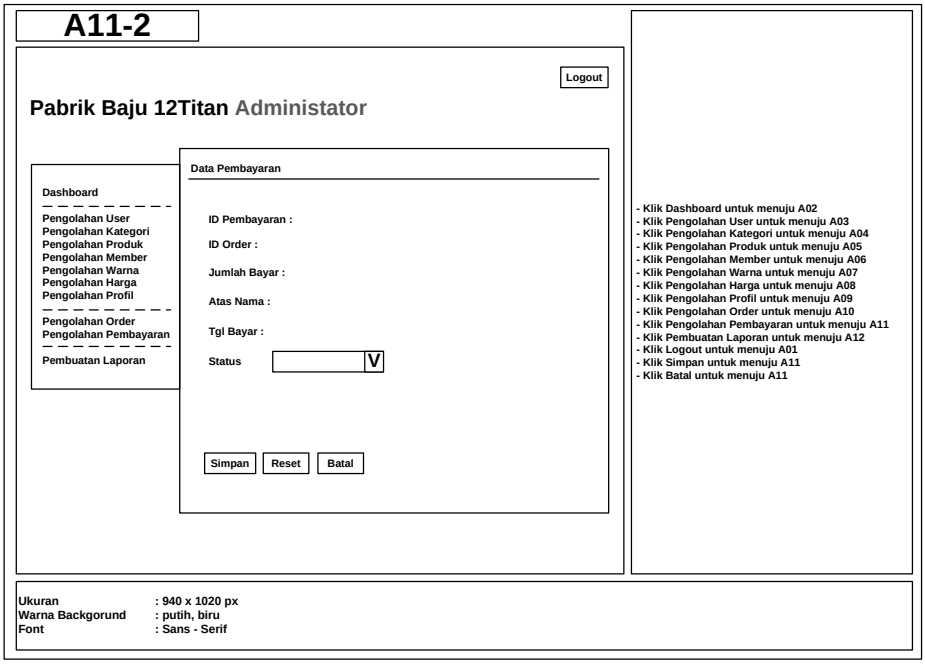
Gambar 3.123 Perancangan Halaman Form Order

40. Perancangan Halaman Pengolahan Pembayaran



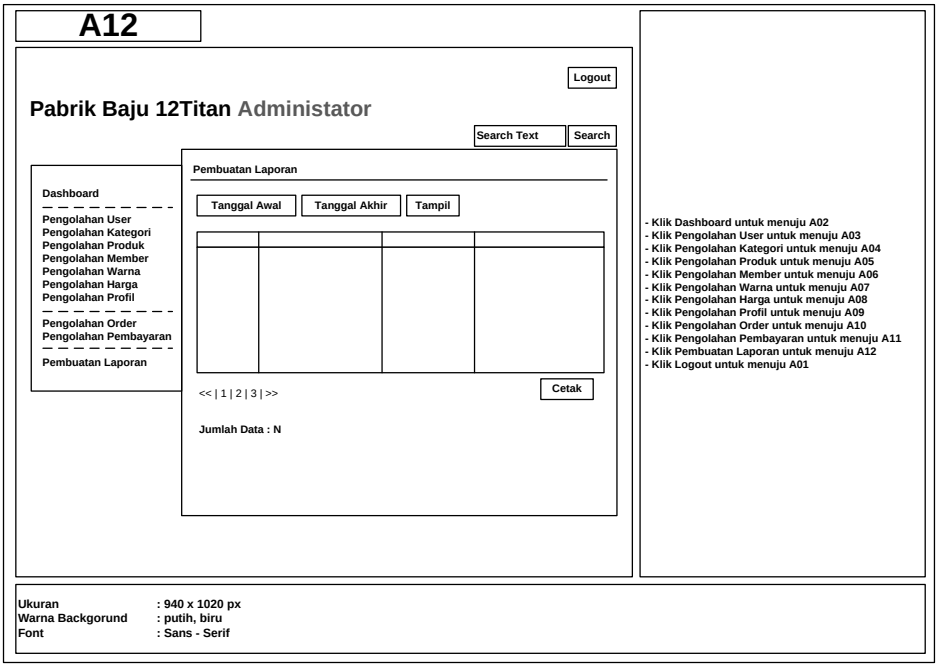
Gambar 3.124 Perancangan Halaman Pengolahan Pembayaran

41. Perancangan Halaman Form Pembayaran



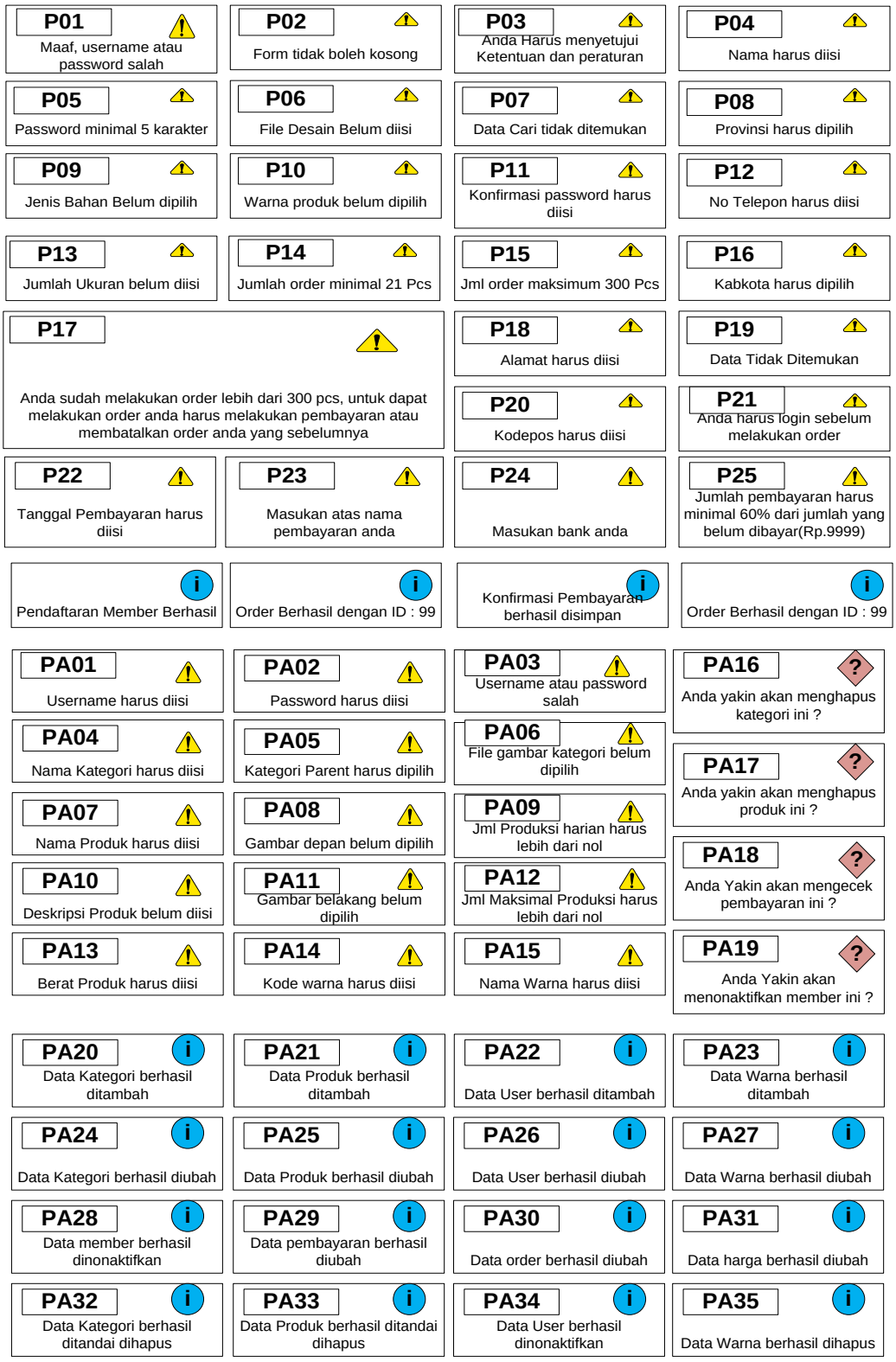
Gambar 3.125 Perancangan Halaman Form Pembayaran

42. Perancangan Halaman Pembuatan Laporan



Gambar 3.126 Perancangan Halaman Pembuatan Laporan

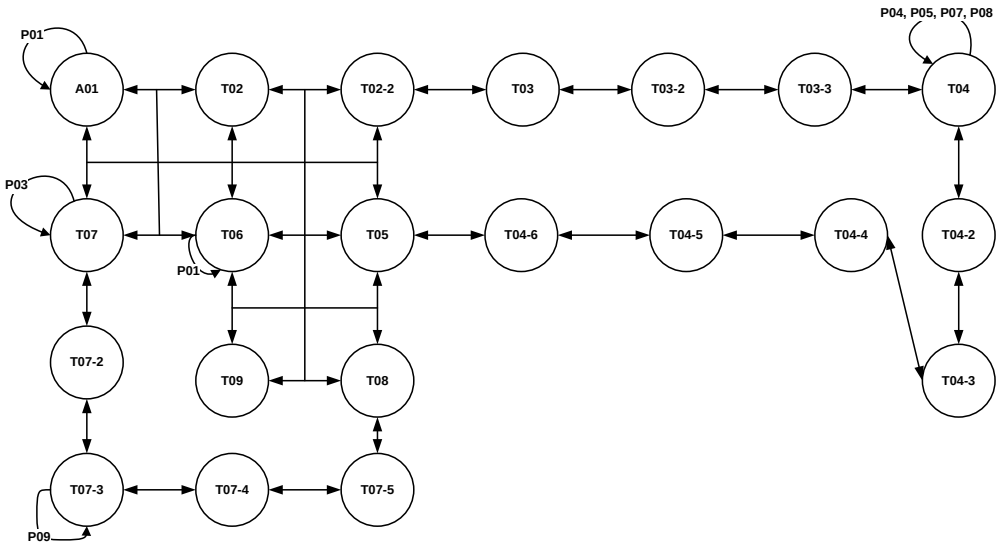
3.2.3.2 Perancangan Pesan



Gambar 3.127 Perancangan Pesan

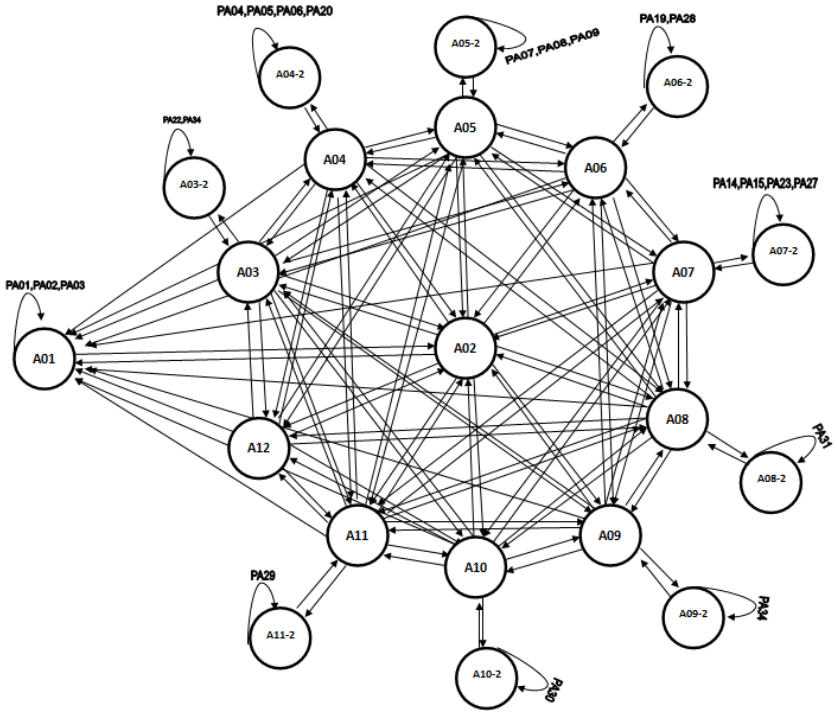
3.2.4 Jaringan Semantik

1. Jaringan Semantik untuk pengunjung



Gambar 3.128 Jaringan Semantik Pengunjung

2. Jaringan Semantik untuk Admin



Gambar 3.129 Jaringan Semantik Admin

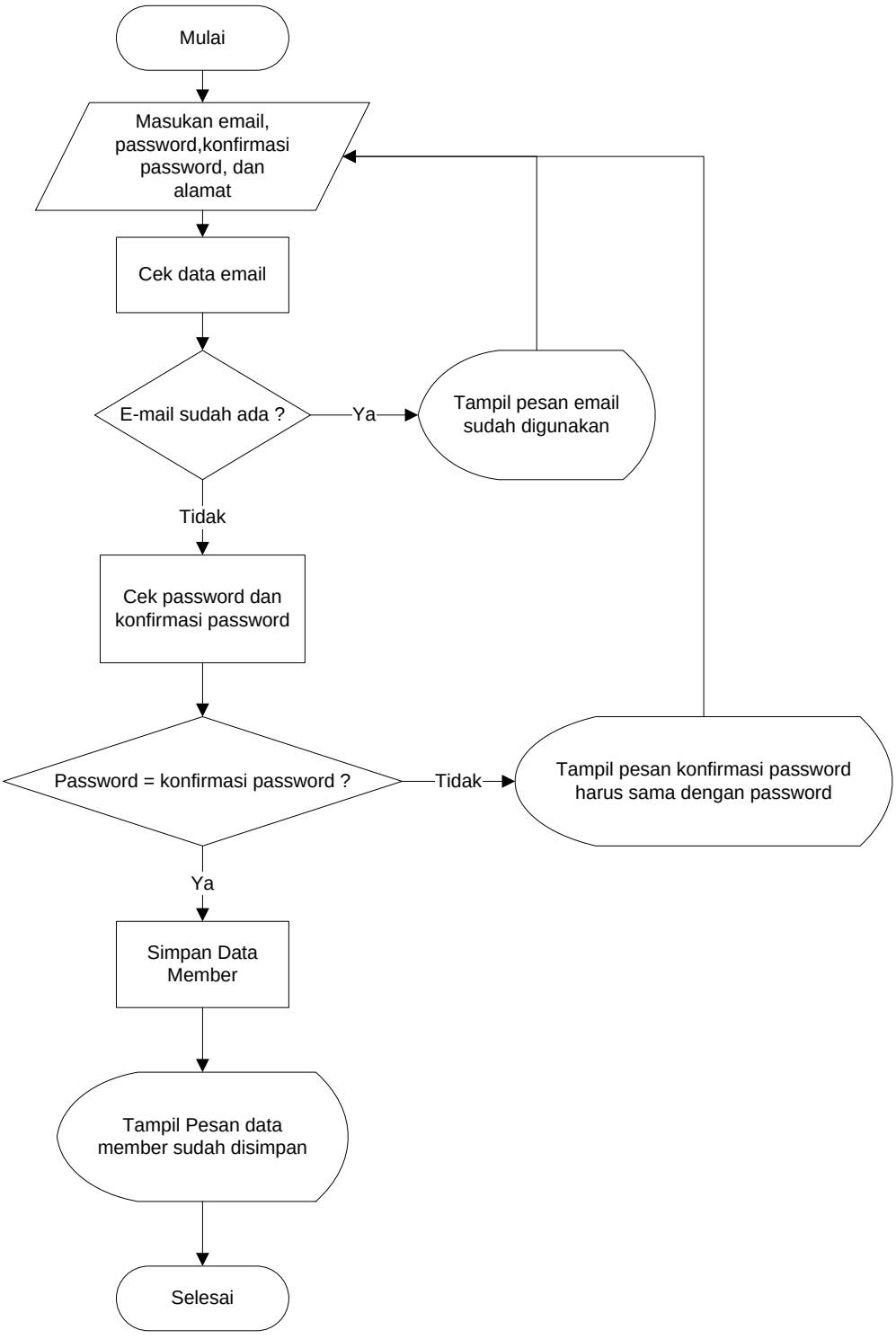
3.2.5 Perancangan Prosedural

Perancangan prosedural adalah tata cara atau urutan langkah-langkah 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 aplikasi pengolahan data produksi adalah sebagai berikut:

1. **Flowchart Pendaftaran Member**

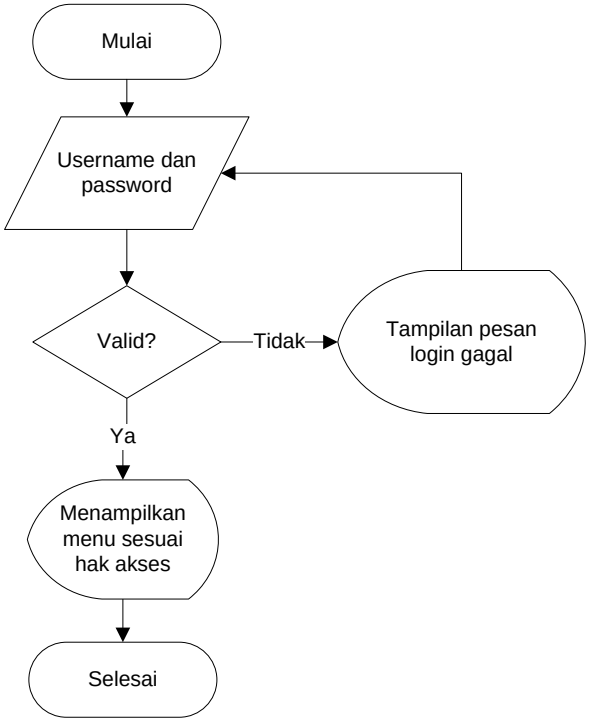
   Prosedur pendaftaran member merupakan proses pengisian data member beserta pengecekan data yang dimasukan untuk mendaftar menjadi seorang member. Flowchart pendaftaran member dapat dilihat pada gambar 3.130.



Gambar 3.130 Flowchart Pendaftaran Member

2. Flowchart *Login*

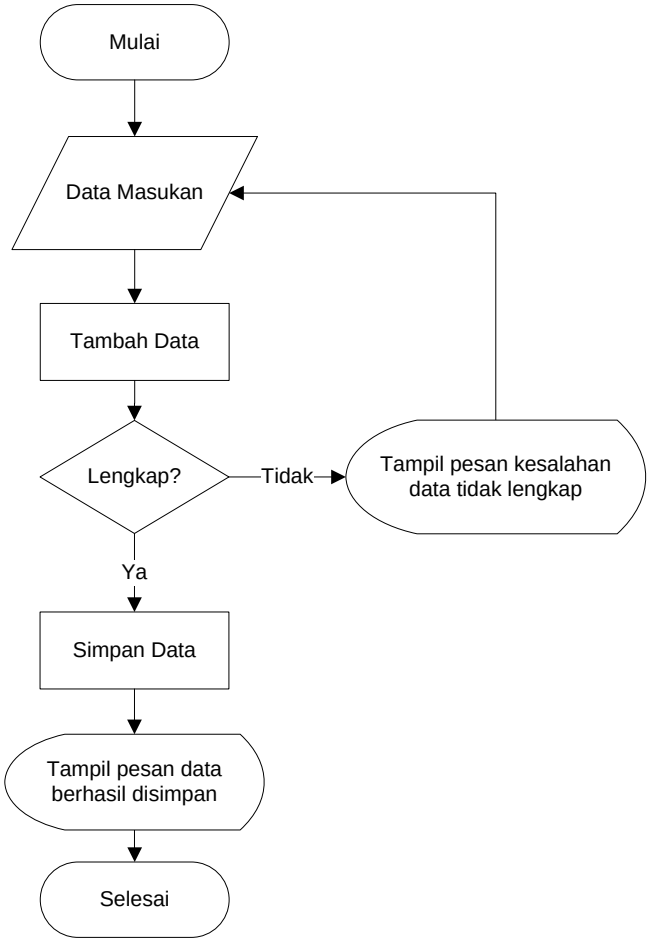
Diagram alir pada prosedur *login* adalah sebagai berikut:



Gambar 3.131 Flowchart Login

3. Flowchart Tambah Data

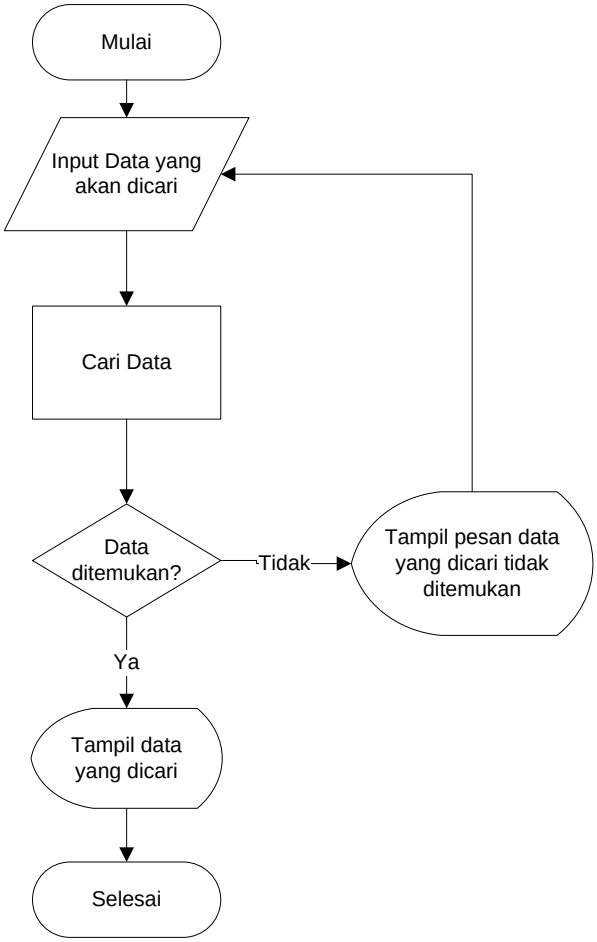
Diagram alir pada prosedur tambah data adalah sebagai berikut:



Gambar 3.132 Flowchart Tambah Data

4. Flowchart Cari Data

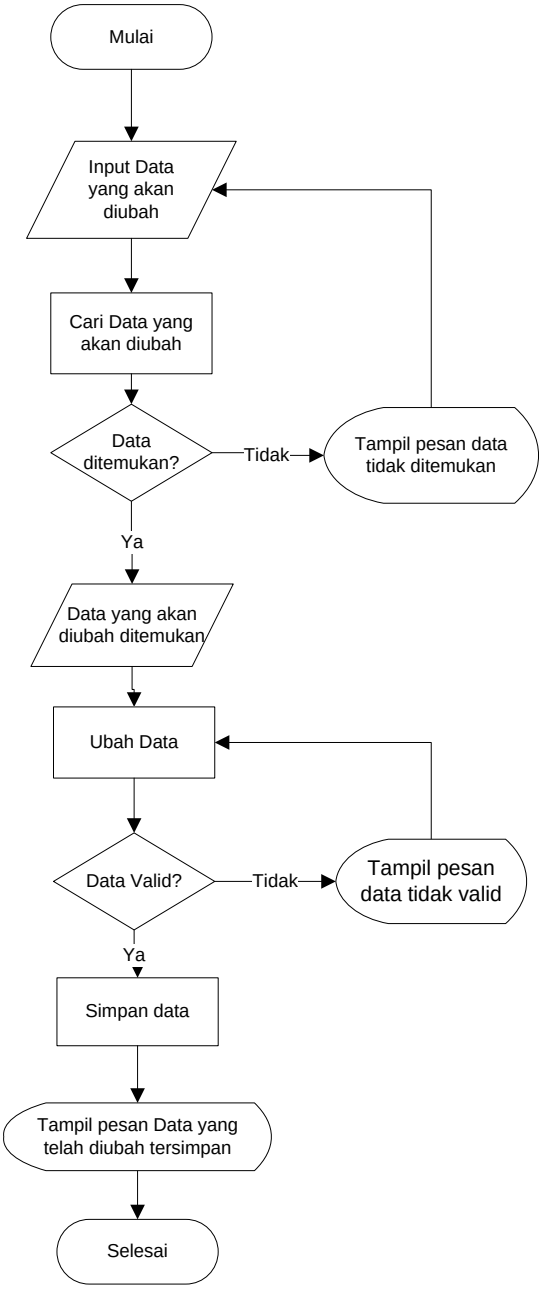
Diagram alir pada prosedur cari data adalah sebagai berikut:



Gambar 3.133 Flowchart Cari Data

5. Flowchart ubah data

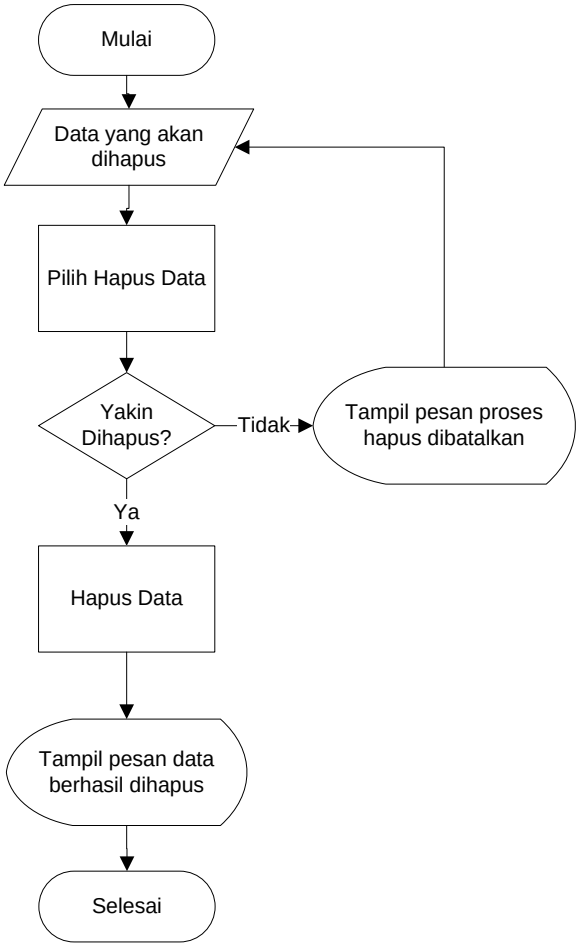
Diagram alir pada prosedur ubah data adalah sebagai berikut:



Gambar 3.134 Flowchart Ubah Data

6. Flowchart hapus data

Diagram alir pada prosedur hapus data adalah sebagai berikut:



Gambar 3.135 Flowchart Hapus Data

7. Flowchart Pemesanan Produk

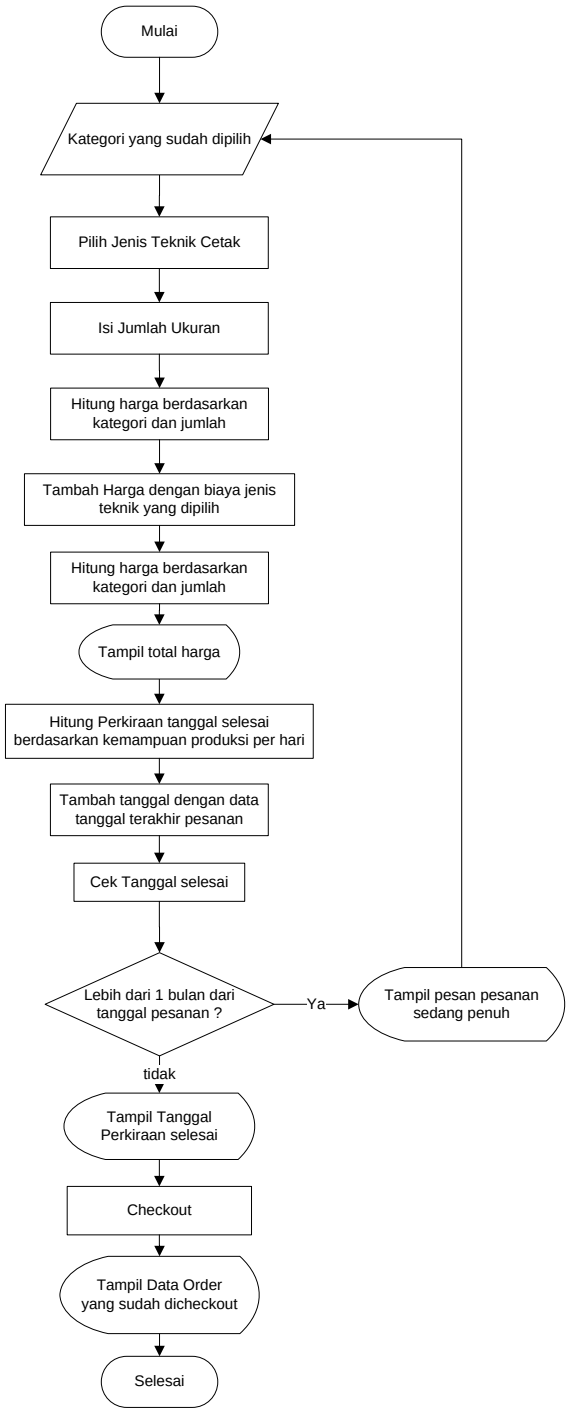
Diagram alir pada prosedur Pemesanan produk adalah sebagai berikut:



Gambar 3.136 Flowchart Pemesanan Produk

8. Flowchart Estimasi Harga dan Tanggal Perkiraan Selesai

Diagram alir pada prosedur penentuan harga dan perkiraan tanggal selesai adalah sebagai berikut:



Gambar 3.137 Flowchart Estimasi Harga dan Tanggal Perkiraan Selesai