BAB 3
ANALISIS PERANCANGAN

3.1 Analisis Sistem

Analisis sistem merupakan tahapan kritis untuk mengkaji, memahami kebutuhan yang akan dikembangkan, dengan berlandaskan pada sistem yang telah berjalan. Dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan, dan hambatanyang ada, sehingga dapat diusulkan perbaikan.

Dalam proses analisis akan dibahas mengenai analisis yang sedang berjalan, analisis yang akan dikembangkan, analisis kebutuhan non fungsional, analisis *web analytics*, dan analisis kebutuhan fungsional.

3.1.1 Analisis Sistem Yang Berjalan

Prosedur mengindikasikan rangkaian aktivitas, tugas-tugas, langkah-langkah, keputusan-keputusan, perhitungan-perhitungan dan proses-proses yang dijalankan melalui serangkaian pekerjaan yang menghasilkan suatu tujuan yang diinginkan, dapat berupa suatu produk atau sebuah akibat.

Berdasarkan hasil wawancara dan observasi yang dilakukan pada pihak Campus Fashion Centre prosedur bisnis yang berjalan dapat diklasifikasikan yaitu: prosedur pengadaan barang, prosedur pendataan barang, prosedur transaksi penjualan, prosedur laporan penjualan harian, prosedur laporan penjualan bulanan.

3.1.1.1.Pengadaan Barang

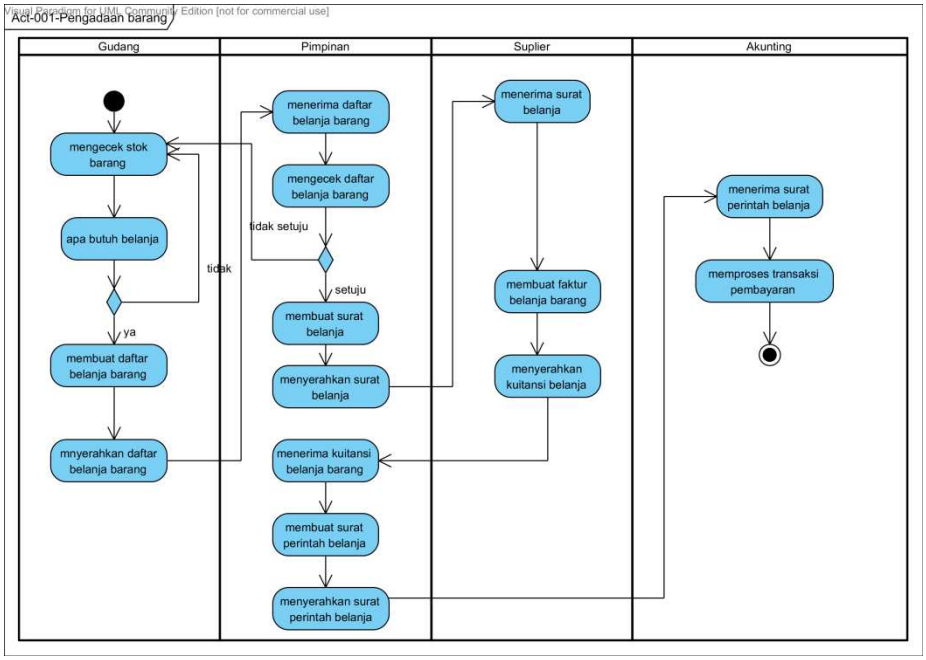
Prosedur yang dilakukan untuk pengadaan barang, yaitu sebagai berikut:

1. Setiap akhir pekan bagian gudang mengecek stok barang yang ada digudang.
2. Bagian gudang membuat rincian stok barang, apabila dirasa perlu menambah stok barang, bagian gudang akan memberikan laporan rencana belanja kepada Pimpinan.
3. Pimpinan menerima laporan mengenai rencana belanja dari Bagian Gudang, Pimpinan mengecek laporan rencana belanja, jika sesuai dan

dianggap perlu menambah stok barang, pimpinan membuat surat belanja yang disampaikan kepada suplier.

4. Suplier menerima daftar belanja dan membuat nota belanja sesuai dengan daftar belanja yang diterima, setelah nota belanja dibuat akan disampaikan kepada Pimpinan.
5. Pimpinan menerima nota belanja dari Suplier dan membuat surat perintah belanja, surat perintah belanja disampaikan kepada Bagian Akunting.
6. Bagian Akunting menerima surat perintah belanja dari Pimpinan, kemudian melakukan proses belanja.

Prosedur pengadaan barang yang sedang berjalan dapat dilihat pada diagram *activity* pada gambar 3.1.



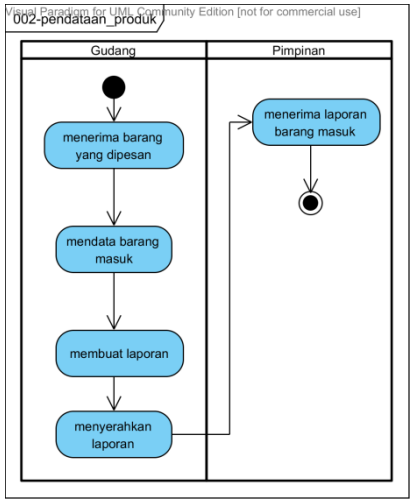
Gambar 3. 1. Diagram *Activity* Pengadaan Barang

3.1.1.2.Pendataan Barang

Prosedur yang dilakukan untuk pendataan barang, yaitu sebagai berikut:

1. Bagian Gudang menerima barang dan mendata barang yang diterima.
2. Setelah pendataan selesai, Bagian Gudang membuat laporan pendataan barang masuk, dan dilaporkan pada Pimpinan.

Proses pendataan barang masuk yang sedang berjalan dapat dilihat pada diagram *activity* pada gambar 3.2.



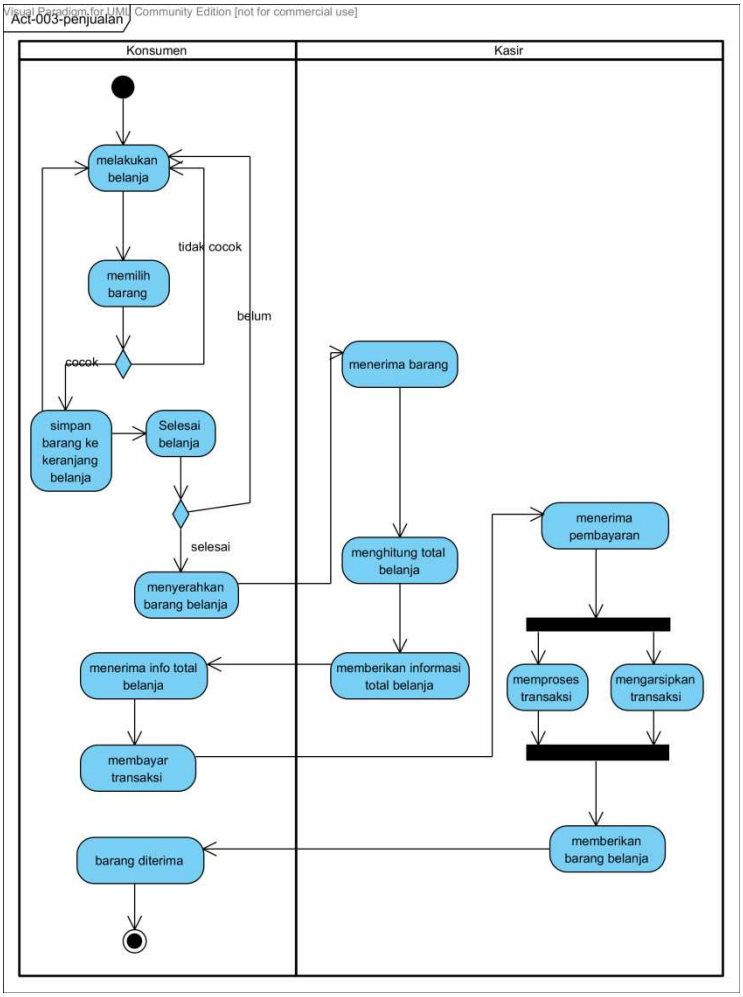
Gambar 3. 2. Diagram *Activity* Pendataan Barang

3.1.1.3.Transaksi Penjualan

Prosedur yang dilakukan untuk transaksi penjualan, yaitu sebagai berikut:

1. Konsumen memilih barang yang dipajang di outlet, bila ada barang yang sesuai dengan keinginan maka barang akan disimpan di keranjang belanja dan kembali berbelanja memilih barang yang lain.
2. Konsumen memberikan keranjang belanja yang berisi barang yang akan dibeli kepada Pramuniaga atau Kasir, jika dirasa telah selasai berbelanja.
3. Kasir menerima keranjang belanja yang berisi barang yang akan dibeli, dan menghitung total belanja yang dilakukan Konsumen. Setelah selesai menghitung total belanja Kasir menginformasikan total belanja kepada Konsumen.
4. Konsumen membayar dengan sejumlah yang diinformasikan kepada Kasir.
5. Kasir menerima pembayaran yang dilakukan oleh Konsumen, dan menghitung jumlah pembayaran.
6. Kasir membungkus barang yang dibeli dan menyerahkan barang kepada Konsumen.
7. Kasir mencatat transaksi yang telah dilakukan.

Proses transaksi penjualan yang sedang berjalan dapat dilihat pada diagram *activity* gambar 3.3.



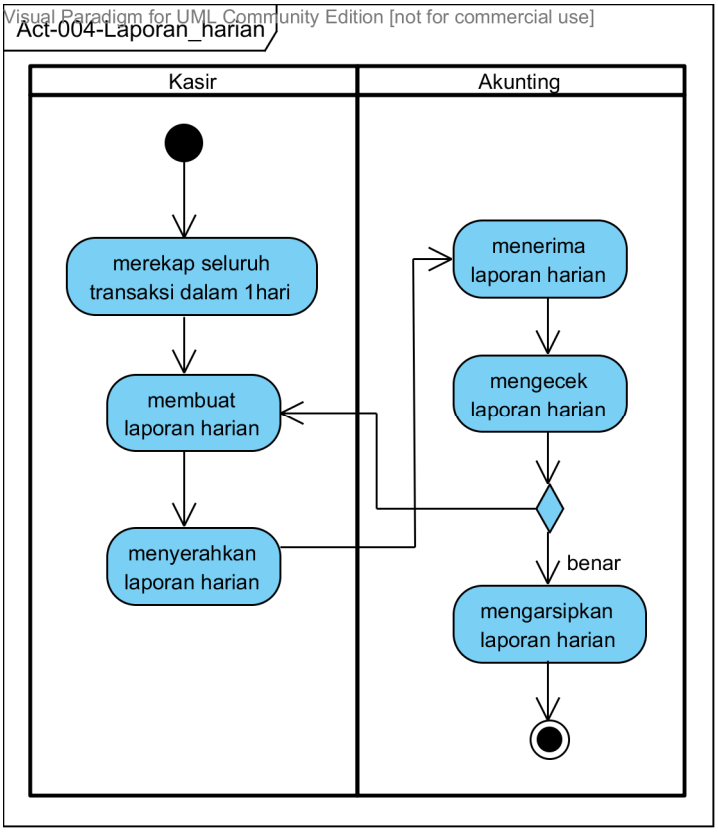
Gambar 3. 3. Diagram *Activity* Transaksi Penjualan

3.1.1.4.Laporan Penjualan Harian

Prosedur yang dilakukan untuk laporan penjualan harian, yaitu sebagai berikut:

1. Kasir mengumpulkan nota transaksi penjualan selama satu hari, kemudian melakukan rekapitulasi.
2. Kasir mencatat hasil transaksi harian pada file Microsoft Excel.
3. Kasir memberikan laporan penjualan harian kepada Bagian Akunting.
4. Bagian Akunting menerima laporan penjualan harian, dan mengecek isi laporan, bila telah dianggap benar maka laporan diarsipkan.

Proses laporan penjualan harian yang sedang berjalan dapat dilihat pada diagram *activity* gambar 3.4.



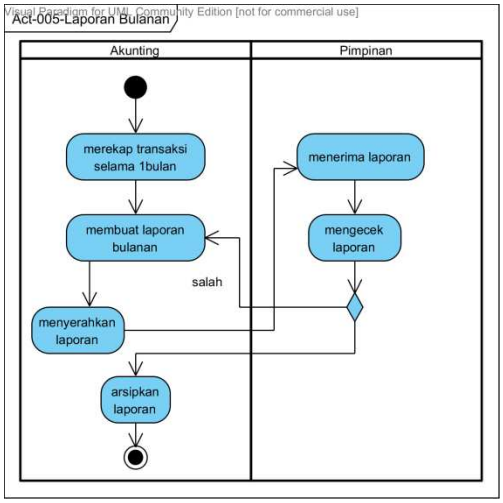
Gambar 3. 4. Diagram *Activity* Laporan Penjualan Harian

3.1.1.5.Laporan Penjualan Bulanan

Prosedur yang dilakukan untuk laporan penjualan bulanan, yaitu sebagai berikut:

1. Bagian Akunting melakukan rekapitulasi terhadap data transaksi penjualan harian yang ada pada bulan tertentu pada file Microsoft Excel.
2. Bagian Akunting menyerahkan laporan penjualan bulanan kepada Pimpinan.
3. Pimpinan mengecek isi laporan, apabila ada yang tidak sesuai Pimpinan akan memerintahkan untuk membuat ulang laporan.
4. Apabila laporan telah sesuai Pimpinan akan menyetujui laporan dan Bagian Akunting mengarsipkan laporan.

Proses laporan penjualan bulanan yang sedang berjalan dapat dilihat pada diagram *activity* gambar 3.5.



Gambar 3. 5. Diagram *Activity* Laporan Bulanan

3.1.2 Analisis Yang Akan Dikembangkan

Berdasarkan analisis yang telah dilakukan pada prosedur yang sedang berjalan, dibutuhkan prosedur tambahan untuk melengkapi prosedur yang telah ada. Diantaranya prosedur registrasi, prosedur *login*, pengolahan data pegawai, pengolahan stok barang, prosedur transaksi pembelian, dan prosedur retur.

3.1.2.1.Registrasi

Registrasi merupakan proses pendaftaran bagi pengunjung untuk menjadi *member*. Registrasi dibutuhkan untuk melakukan pendataan, dan juga memudahkan dalam proses transaksi belanja.

Untuk melakukan proses registrasi harus memenuhi kriteria sebagai berikut:

1. Memiliki *email* aktif.
2. Satu akun *email* hanya dapat didaftarkan sebagai satu akun *member*.
3. Membuat *username* yang digunakan sebagai *login* sebagai *member*, kententuan untuk pembuatan *username* adalah minimal empat karakter dan maksimal 12 karkter terdiri dari *alpha numeric* dan *underscore*, dan bersifat unik.

4. Mengisi *pasword* yang digunakan sebagai proses validasi pada saat *login*.
5. Mengisikan data pribadi dengan benar berupa nama, jenis kelamin, dan tanggal lahir pada *form* registrasi.

Prosedur registrasi untuk menjadi member Campus Fashion Centre adalah sebagai berikut:

1. Buka halaman registrasi.
2. Setelah berada pada halaman registrasi, isi *form* registrasi yang terdiri dari *username*, nama, *email*, jenis kelamin, tanggal lahir, *password* dan konfirmasi *password*.
3. Apabila *form* telah terisi dengan benar tekan tombol daftar, jika validasi gagal, akan muncul pesan kesalahan.
4. Pada saat kondisi validasi benar pengunjung akan mendapat pesan konfirmasi yang digunakan sebagai aktivasi sebagai *member* melalui email yang didaftarkan.
5. Pengunjung harus melakukan aktivasi untuk menjadi *member*, jika aktivasi berhasil pengunjung dapat *login* sebagai *member*.

3.1.2.2.*Login*

Proses *login* merupakan proses validasi pengunjung untuk dapat menggunakan hal akses sebagai *member*. Proses *login* dibutuhkan untuk menjaga kemamanan dan kenyamanan pada saat melakukan transaksi.

Member dapat melakukan proses *login* pada halaman *login* dengan mencantumkan *username* dan *password* yang telah didaftarkan.

3.1.2.3.Pengolahan Data Pegawai

Untuk mengoperasikan aplikasi dibutuhkan operator yang bertugas untuk mengelola data, dari mulai data master, data barang, hingga data transaksi. Pengaturan pengolahan data pegawai diatur oleh Pimpinan.

3.1.2.4.Pengolahan Stok Barang

Stok merupakan sejumlah barang yang tersedia di gudang untuk selanjutnya akan dijual kepada konsumen, dalam hal ini *member* yang melakukan transaksi. Pengolahan stok gudang di atur dengan ketentuan sebagai berikut:

1. Stok barang di *e-commerce* dibedakan dengan di outlet.
2. Pengurangan stok akan terjadi ketika proses *checkout* atau pemesanan, apabila setelah pemesanan dalam jangka waktu 24 jam tidak dilakukan konfirmasi pembayaran maka stok akan dikembalikan.

3.1.2.5.Proses Transaksi Pembelian

Transaksi pembelian merupakan proses yang meliputi dari mulai menambahkan ke keranjang belanja, pemesanan, konfirmasi pembayaran, pengiriman, hingga proses penerimaan barang. Setiap tahapan harus dilakukan secara berurutan dan sesuai aturan yang ditetapkan. Untuk melakukan transaksi pembelian sebelumnya harus terdaftar sebagai *member*. Berikut prosedur untuk melakukan transaksi pembelian:

1. Pengunjung memilih barang yang akan dibeli dengan mengklik *thumbnail* gambar barang yang dipilih. Apabila akan membeli maka akan menambahkan ke keranjang belanja, dengan cara memilih ukuran barang dan mengklik beli, kemudian secara otomatis akan menuju halaman keranjang belanja.
2. Pada halaman keranjang belanja dapat diubah jumlah pesanan atau menghapus pesanan.
3. Apabila proses belanja telah selesai klik lanjutkan transaksi, jika telah login maka akan langusung menuju halaman pemesanan, jika belum maka akan menuju halaman *login* untuk melakukan proses *login* terlebih dahulu.
4. Pada halaman pemesanan akan diminta untuk mengisi *form* alamat, pengiriman, pilihan kurir, dan metode pembayaran. Setelah berhasil mengisi alamat pilihan kurir akan muncul perhitungan total pembayaran beserta tombol *checkout*.

5. Member menklik tombol *checkout*, apabila *form* isian terisi dengan benar maka akan menuju halaman *member*, dan proses pemesanan selesai.
6. Member harus segera melakukan konfirmasi pembayaran sebelum 24 jam terhitung dari proses *checkout* dengan mengisi *form* konfirmasi pembayaran, setelah melakukan konfirmasi pembayaran proses transaksi pembelian selesai, member dipersilahkan untuk menunggu pengiriman barang.

3.1.2.6.Proses Retur

Proses retur merupakan proses pengembalian kembali barang yang telah dibeli dengan alasan tertentu. Untuk melakukan proses retur terdapat beberapa syarat dan ketentuan yang harus dipenuhi, yaitu:

1. Retur dapat diterima, bila barang yang diretur pada saat status order telah diterima dan waktu tenggang retur 2 x 24 jam setelah perubahan status retur diterima.
2. Barang dapat dilakukan retur bila pada saat diterima barang dalam kondisi rusak atau cacat atau pada saat diterima barang tidak sesuai dengan barang pada saat dipesan, hal ini dibuktikan dengan mengunggah *file* gambar barang yang akan diretur.
3. Label pada barang yang akan diretur tidak boleh rusak, harus sama seperti pada saat barang diterima.
4. Biaya pengiriman barang dari *member* ke pihak Campus Fashion Centre ditanggung oleh *member*, biaya pengiriman kembali dari pihak Campus Fashion Centre ke *member* ditanggung oleh pihak Campus Fashion Centre.

Prosedur yang dilakukan untuk melakukan proses retur adalah sebagai berikut:

1. *Member* membuka halaman *history* retur, apabila halaman *history* retur telah terbuka, member mengklik tambah retur maka akan muncul *form* pencarian order. *Member* memasukkan id order yang akan diretur, apabila order tidak ditemukan akan muncul pemberitahuan dan bila order ditemukan akan muncul *form* tambah retur.

2. Pada halaman tambah retur member akan diminta untuk memilih barang mana yang akan diretur, berserta mengisikan jumlah barang yang diretur dan alasan barang diretur, klik tombol selesai bila telah selesai mengisi detail retur, apabila proses berhasil member akan diarahkan menuju halaman detail retur dan diminta untuk menambahkan keterangan gambar terhadap barang yang diretur.
3. *Member* menunggu status barang retur dari pihak Campus Fashion Centre, apabila barang yang diretur diterima maka member mengirimkan, dan mengisikan detail pengiriman pada halaman detail retur, dan menunggu barang dikirim kembali oleh pihak Campus Fashion Centre.

3.1.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah analisis yang dilakukan untuk mengetahui bagaimana keadaan sistem yang ada di Campus Fashion Centre, yang meliputi komponen pendukung yang telah ada, sehingga akan diusulkan kebutuhan untuk menjalankan *e-commerce* yang akan dibangun sampai dengan sistem diimplementasikan, sehingga apa yang dibangun dapat berjalan optimal.

Analsisis kebutuhan non fungsional meliputi analisis perangkat keras, analisis perangkat lunak, dan analisis pengguna.

3.1.3.1.Analisis Perangkat Keras

Perangkat keras yang terdapat saat ini pada Campus Fashion Centre yaitu satu unit komputer operasional dengan spesifikasi sebagai berikut:

1. Processor : AMD X2 555 3.0 GHz
2. Monitor : LCD 19”
3. RAM : 2 GB share with VGA
4. VGA : 1GB Onboard
5. Harddisk : 500 GB
6. LAN Card : Onboard
7. Mouse optic
8. Keyboard
9. Jaringan internet

Perangkat keras minimum yang dibutuhkan untuk menjalankan aplikasi yang akan dibangun adalah sebagai berikut:

1. Prosessor : Intel Pentium 4 2.0GHz
2. Monitor : Monitor VGA 15”
3. Memori : DDR2 512GB
4. Harddisk : 80 GB
5. Keyboard
6. Mouse
7. Printer
8. Jaringan internet ada

Perangkat keras sudah ada mencukupi untuk menjalankan aplikasi, namun untuk lebih optimal perangkat keras untuk operasional ditambah dalam hal kuantitas.

3.1.3.2.Analisis Perangkat Lunak

Untuk membangun aplikasi *e-commerce* dibutuhkan spesifikasi perangkat lunak sebagai berikut:

1. WampServer, yang mencakup web server Apache, DBMS MySQL, dan bahasa pemrograman PHP, dan framework Codeigniter.
2. NetBeans IDE 6.9.1, sebagai aplikasi pembangun aplikasi.
3. Web browser, Mozilla Firefox, Google Chrome, Internet Explorer.

Untuk menjalankan aplikasi *e-commerce* dibutuhkan spesifikasi minimal perangkat lunak yang setara dengan:

1. Sistem operasi Windows XP.
2. Browser Google Chrome, Mozilla Firefox 5, Internet Explorer 8.

Aplikasi perangkat lunak yang digunakan saat ini di Campus Fashion Centre adalah:

1. Sistem operasi Windows 7 Ultimate.
2. Browser Mozilla Firefox 16.0.2, Google Chrome, Internet Explorer 9.

Berdasarkan apa yang dipaparkan maka spesifikasi untuk perangkat lunak sudah cukup untuk menjalankan aplikasi yang akan dikembangkan.

3.1.3.3.Analisis Kebutuhan Server

Aplikasi dibangun berbasis web sehingga dibutuhkan server untuk menampung data yang disimpan, berikut spesifikasi kebutuhan server yang dibutuhkan.

1. Web *server* IIX
2. Web *space* 500MB
3. *Bandwith* 50GB
4. DBMS MySQL
5. *Dedicated* IP
6. AlphaSSL *Certificate*

3.1.3.4.Analisis Pengguna

Suatu sistem akan berjalan optimal bukan hanya karena teknologinya, namun juga harus diperhatikan sumber daya manusia yang menggunakan. Untuk itu akan dilakukan analisis mengenai sumber daya manusia yang ada di Campus Fashion Centre, SDM yang tersedia di Campus Fashion Centre dapat dilihat pada tabel 3.1.

Tabel 3. 1. Karakteristik Pengguna Campus Fashion Centre

Pengguna	Tanggung Jawab	Tingkat Pendidikan	Tingkat Keterampilan
Pimpinan	Mengawasi jalannya proses bisnis secara umum	S1	Mampu mengoperasikan komputer dengan sistem operasi Windows dan mampu menggunakan internet dengan baik.
Akunting	Mencatat transaksi keluar maupun masuk.	D3	Mampu mengoperasikan komputer dengan sistem operasi Windows dan mampu menggunakan internet dengan baik.

Pengguna	Tanggung Jawab	Tingkat Pendidikan	Tingkat Keterampilan
Gudang	Mencatat lalu lintas barang dan menentukan label harga jual.	SMA	Mampu mengoperasikan komputer dengan sistem operasi Windows dan mengenal internet.
Pramuniaga dan kasir	Melayani transaksi penjualan	SMA	Mampu mengoperasikan komputer dengan sistem operasi Windows dan mampu menggunakan internet dengan baik.

Pengguna aplikasi ini akan dibagi menjadi beberapa bagian yang mengadopsi dari yang telah berjalan dan menyesuaikan dengan apa yang dibutuhkan, berdasarkan analisis yang telah dilakukan untuk pembagian pengguna aplikasi yang akan dikembangkan dapat dilihat pada tabel 3.2.

Tabel 3. 2. Kebutuhan Pengguna Aplikasi E-Commerce

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Keterampi-lan	Pelatihan yang dibutuhkan
Pimpinan	Mengelola data pegawai, laporan dan analisis statistik kunjungan	Mengakses laporan, melakukan pengolahan data kepegawaian, dan statistik analisis kunjungan	Berpengala-man mengguna-kan sistem operasi Windows, tidak asing dengan internet	pengenalan terhadap aplikasi yang dibangun
Gudang	Mengelola data barang	Mengelola data barang dan segala atribut yang menyertai	Berpengala-man mengguna-kan sistem operasi Windows, tidak asing dengan internet	pengenalan terhadap aplikasi yang dibangun

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Keterampi-lan	Pelatihan yang dibutuhkan
Akunting dan administrasi	Mengelola transaksi, member.	Mengelola proses transaksi, retur maupun penjualan, mengelola bagian member	Berpengala-man mengguna-kan sistem operasi Windows, tidak asing dengan internet	pengenalan terhadap aplikasi yang dibangun
Database administrator	Mengelola data tetap	melakukan pengolahan data rekening, data provinsi, data kab/kota	Berpengala-man mengguna-kan sistem operasi Windows, tidak asing dengan internet	pengenalan terhadap aplikasi yang dibangun
Pengunjung	Mengakses fitur umum aplikasi	Mengakses fitur umum aplikasi	mengenal internet dan mampu mengguna-kan.	-
Member	Melakukan transaksi pembelian	melakukan proses belanja, melakukan proses retur	mengenal internet dan mampu mengguna-kan.	-

Secara umum kebutuhan yang ada dibandingkan dengan apa yang dibutuhkan untuk menjalankan sistem yang dibuat telah mencukupi.

3.1.4 Analisis *Web Analytics*

Untuk memaksimalkan proses pemasaran barang, harus diketahui tentang bagaimana perilaku dari pengunjung yang mengakses situs, dengan demikian akan dapat diperoleh masukan tentang bagaimana proses pengambilan keputusan bisnis yang akan diambil. *Web analtics* merupakan metode untuk menggambarkan pola kunjungan terhadap sebuah situs.

3.1.4.1 Pengumpulan Data

Proses pengumpulan data merupakan proses yang menentukan dalam *web analytics*. Dalam *web analtics* dikenal dua cara untuk melakukan proses pengumpulan data, *page tags*, dan *logfiles*. Dalam hal ini akan digunakan teknik *page tags*, dengan ketentuan sebagai berikut:

1. Aplikasi mencatat data dari *browser* pengunjung yaitu: id_pengunjung, id_visit, id_user, level, ip_add, waktu, halaman, browser, *platform* dan di simpan dalam tbl_log dalam *database*.
2. Aplikasi menyimpan *cookies* di *browser* pengunjung dengan nama cfc_pengunjung, yang digunakan sebagai id_pengunjung, dan cfc_visit yang digunakan sebagai id_visit.
3. *Cookies* cfc_pengunjung akan hilang selama waktu 2592000 detik, dan cfc_visit akan hilang jika browser yang digunakan pengunjung ditutup.

Alur untuk pengumpulan data adalah sebagai berikut:

1. Ketika pengunjung mengakses situs untuk pertama kali aplikasi akan melakukan pengecekan apakah telah terdapat nilai *cookies* dengan nama cfc_pengunjung dan cfc_visit di browser pengunjung. Jika tidak ditemukan maka aplikasi akan menyimpan nilai *cookies* pada browser pengunjung.
2. Aplikasi akan melakukan pengecekan apakah pegunjung terdaftar sebagai *member*, pengunjung, atau sebagai administrator, dan setelah melakukan pengecekan data akan disimpan dalam tbl_log di dalam *database*.

			id_log	id_pengunjung	id_visit	id_user	level	ip_add	waktu	halaman	browser	platform
			51	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	0	guest	202.152.204.240	2012-12-02 11:55:25	beranda	Firefox 16.0	Unknown Windows OS
			52	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	0	guest	202.152.204.240	2012-12-02 11:56:37	login	Firefox 16.0	Unknown Windows OS
			53	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	0	guest	202.152.204.240	2012-12-02 11:56:44	login	Firefox 16.0	Unknown Windows OS
			54	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	1	admin	202.152.204.240	2012-12-02 11:56:47	asset/design /content.gif	Firefox 16.0	Unknown Windows OS
			55	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	1	admin	202.152.204.240	2012-12-02 11:56:47	beranda	Firefox 16.0	Unknown Windows OS
			56	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	1	admin	202.152.204.240	2012-12-02 11:56:51	asset/design /content.gif	Firefox 16.0	Unknown Windows OS
			57	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	1	admin	202.152.204.240	2012-12-02 11:56:51	beranda	Firefox 16.0	Unknown Windows OS
			58	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	1	admin	202.152.204.240	2012-12-02 11:57:53	asset/design /content.gif	Firefox 16.0	Unknown Windows OS
			59	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	1	admin	202.152.204.240	2012-12-02 11:57:56	beranda	Firefox 16.0	Unknown Windows OS
			60	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	1	admin	202.152.204.240	2012-12-02 11:57:59	asset/design /content.gif	Firefox 16.0	Unknown Windows OS
			61	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	1	admin	202.152.204.240	2012-12-02 11:57:59	beranda	Firefox 16.0	Unknown Windows

Gambar 3. 6. Sampel Data *Log* Pada Tabel *Log*

3.1.4.2 Standart Pengukuran

Untuk mendapatkan nilai informasi yang akurat dibutuhkan satuan pengukuran yang akan membantu mendeskripsikan nilai *trend* kunjungan dalam sebuah situs. Berikut aturan pengukuran yang akan dikembangkan dalam Campus Fashion Centre:

1. Hit, merupakan jumlah *request* halaman yang berhasil terbuka oleh browser pengunjung, nilai hit merupakan pendekatan untuk menghitung jumlah halaman yang diakses oleh pengunjung. Pada saat pengunjung mengakses sebuah halaman dalam situs maka akan dihitung sebagai satu hit.
2. Kunjungan, merupakan nilai pendekatan untuk menghitung seberapa banyak jumlah pengunjung yang menghabiskan waktu dalam situs. Satu pengunjung dapat melakukan satu atau lebih kunjungan. Berikut aturan untuk proses penghitungan kunjungan:
   a. Pegunjung melakukan *request* pertama kali pada satu kunjungan maka aplikasi akan menyimpan nilai *cookies* dengan nama cfc_visit dalam browser pengunjung.
   b. *Request* pengunjung dalam satu sesi kunjungan akan disimpan dengan nilai kunjungan yang sama.

c. Ketika pengunjung keluar dari browser maka *session* cfc_visit akan hilang.
d. Nilai perhitungan kunjungan akan dihitung sebagai jumlah kunjungan dalam periode tertentu.

3. Pengunjung, merupakan pendekatan untuk menghitung jumlah orang yang mengakses dalam sebuah situs, berikut aturan untuk perhitungan pengunjung:
   a. Pengunjung melakukan akses untuk pertama kali pada situs, maka aplikasi akan menyimpan nilai *cookies* unik pada browser pengunjung.
   b. *Cookies* akan disimpan dalam browser meskipun browser telah ditutup. Nilai *cookies* merupakan nilai acak unik yang dibuat oleh aplikasi, dan tidak mengandung informasi pribadi dari pengunjung.
   c. Setiap kali pengunjung yang datang dengan menggunakan browser yang telah memiliki nilai *cookies* pengunjung maka akan dianggap sebagai pengunjung yang sama.
   d. Nilai perhitungan pengunjung akan dihitung sebagai jumlah dalam periode tertentu.

Untuk melakukan perhitungan pengunjung ditentukan berdasarkan periode tertentu, sehingga tidak terdapat kerancuan dalam proses perhitungan, berikut simulasi perhitungan pengunjung dalam situs Campus Fashiion Centre. Pada tabel 3.3. merupakan simulasi data kunjungan.

Tabel 3. 3. Simulasi Data Kunjungan

Periode	**Waktu**			**Pengunjung**		
1	Bulan ke-1	Minggu ke-1	Hari ke-1	Avinsh	Dennis	Matt
2	Bulan ke-1	Minggu ke-1	Hari ke-2	Dennis	Matt	Dennis
3	Bulan ke-1	Minggu ke-1	Hari ke-3	Matt	Matt	
4	Bulan ke-1	Minggu ke-2	Hari ke-1	Matt	Ian	
5	Bulan ke-1	Minggu ke-2	Hari ke-2	Ian	Jim	
6	Bulan ke-1	Minggu ke-3	Hari ke-1	Jim	Avinash	Bryan
7	Bulan ke-2	Minggu ke-1	Hari ke-1	Jim	Avinash	Bryan
				Angie	Jennifer	Michelle

Jika periode pertama dihitung maka akan menampilkan nilai bulan ke-1, minggu ke-1, dan hari ke-1 maka akan menghasilkan data kunjungan seperti pada tabel 3.4. Harian merupakan periode perhitungan pengunjung yang dihitung berdasarkan waktu hari, mingguan merupakan periode perhitungan pengunjung yang dihitung berdasar waktu tiap minggu, dan unik merupakan perhitungan pengunjung yang memiliki nilai unik.

Tabel 3. 4. Data Pengunjung Satu Periode

Periode	Total
Harian	3
Mingguan	3
Bulanan	3
Unik	3

Hasil perhitungan pada tabel 3.4 merupakan perhitungan untuk satu periode awal. Data masih konsisten. Perhitungan periode pertama dan kedua maka hasil perhitungan menjadi tampak pada tabel 3.5.

Tabel 3. 5. Data Pengunjung Dua Periode

Periode	Total
Harian	5
Mingguan	3
Bulanan	3
Unik	3

Dalam satu hari pada periode ke-2 denish berkunjung dua kali, namun dihitung sebagai satu pengunjung, sehingga dapat disimpulkan bahwa jika dalam satu hari satu orang berkunjung dua kali akan tetap dihitung sebagai satu pengunjung. Perhitungan periode pertama hingga perode ketiga, tampak pada tabel 3.6.

Tabel 3. 6. Data Pengunjung Tiga Periode

Periode	Total
Harian	6
Mingguan	3

Periode	Total
Bulanan	3
Unik	3

Dalam perhitungan periode pertama hingga periode ke-3 jika dilakukan perhitungan yang sama menghasilkan nilai pengunjung harian total menjadi 6, namun jika dilihat terhadap data yang sebenarnya maka pengunjung hanya terdapat 3 pengunjung. Oleh karena itu perhitungan harian tidak akan memiliki nilai akurat jika digunakan untuk menghitung periode lebih dari satu hari, bergitu juga untuk perhitungan waktu mingguan dan bulanan.

3.1.4.3 Analisis Aktivitas

Analisis aktivitas pengunjung merupakan pengolahan data *log* yang akan diolah sehingga memiliki nilai informasi bagi pihak Fashion Campus Centre. Analisis aktivitas akan lebih di utamakan pada proses yang mendukung dalam hal pengambilan keputusan bisnis dalam bidang pemasaran dan penjualan, yaitu: analisis akivitas berdasarkan waktu akses akan dibagi menjadi berdasarkan waktu akses harian, waktu akses mingguan, dan waku akses bulanan. Selain aktivitas berdasarkan waktu akses juga akan dilakukan analisis mengenai akses terhadap barang, yang diproyeksikan terhadap kategori, merk, dan supplier barang. Aktivitas juga dianalisis berdasarkan profil pengunjung, yang diproyeksikan berdasarkan asal kunjungan, range usia pengunjung, dan jenis kelamin pengunjung. Untuk digunakan sebagai perbandingan dibutuhkan nilai terhadap penjualan barang yang terjadi di Campus Fashion Centre, sehingga juga dibutuhkan analisis aktivitas penjualan barang, yang akan diproyeksikan terhadap kategori, merk dan suplier.

Tabel 3. 7. Data Log

Id	Id Pengunjung	Id Kunjungan	Waktu
1	c81e728d9d4c2f636f067f89cc14862c	50b9eb56eaca61354361686	2012-11-01
3	c4ca4238a0b923820dcc509a6f75849b	50b9eb417492b1354361665	2012-11-01
4	c4ca4238a0b923820dcc509a6f75849b	50b9eb417492b1354361665	2012-12-01
5	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	2012-12-01
6	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	2012-12-01

Id	Id Pengunjung	Id Kunjungan	Waktu
7	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	2012-12-01
8	c4ca4238a0b923820dcc509a6f75849b	50b9ece2a72151354362082	2012-12-01
44	c81e728d9d4c2f636f067f89cc14862c	50b9ee4f085901354362447	2012-12-02
45	eccbc87e4b5ce2fe28308fd9f2a7baf3	50baa52fd634a1354409263	2012-12-02
265	6cdd60ea0045eb7a6ec44c54d29ed402	50bbeffb836131354493947	2012-12-03
266	6cdd60ea0045eb7a6ec44c54d29ed402	50bbeffb836131354493947	2012-12-03
267	6cdd60ea0045eb7a6ec44c54d29ed402	50bbeffb836131354493947	2012-12-03
268	6cdd60ea0045eb7a6ec44c54d29ed402	50bbeffb836131354493947	2012-12-03
536	2838023a778dfaecdc212708f721b788	50bd3f9998e031354579865	2012-12-04
537	2838023a778dfaecdc212708f721b788	50bd3f9998e031354579865	2012-12-04
538	7bcdf75ad237b8e02e301f4091fb6bc8	50bd4aecb16281354582764	2012-12-04
539	5737034557ef5b8c02c0e46513b98f90	50bd4aeccd24c1354582764	2012-12-04
540	9b72e31dac81715466cd580a448cf823	50bd4aecdf7161354582764	2012-12-04
541	16c222aa19898e5058938167c8ab6c57	50bd4aed21c281354582765	2012-12-04
635	6a10bbd480e4c5573d8f3af73ae0454b	50be3ee1bf7dc1354645217	2012-12-05
636	c5ab0bc60ac7929182aadd08703f1ec6	50be3ee279e601354645218	2012-12-05
637	a532400ed62e772b9dc0b86f46e583ff	50be3ee34b0071354645219	2012-12-05
638	4c27cea8526af8cfee3be5e183ac9605	50be6d78225151354657144	2012-12-05
639	4c27cea8526af8cfee3be5e183ac9605	50be6d782255c1354657144	2012-12-05
898	c81e728d9d4c2f636f067f89cc14862c	50c0a0e7614041354801383	2012-12-06
899	c81e728d9d4c2f636f067f89cc14862c	50c0a0e7614041354801383	2012-12-06
900	c81e728d9d4c2f636f067f89cc14862c	50c0a0e7614041354801383	2012-12-06
901	c81e728d9d4c2f636f067f89cc14862c	50c0a0e7614041354801383	2012-12-06
902	c81e728d9d4c2f636f067f89cc14862c	50c0a0e7614041354801383	2012-12-06
931	eccbc87e4b5ce2fe28308fd9f2a7baf3	50c0b226dff781354805798	2012-12-07
932	eccbc87e4b5ce2fe28308fd9f2a7baf3	50c0b226dff781354805798	2012-12-07
933	eccbc87e4b5ce2fe28308fd9f2a7baf3	50c0b226dff781354805798	2012-12-07
934	eccbc87e4b5ce2fe28308fd9f2a7baf3	50c0b226dff781354805798	2012-12-07
1347	eccbc87e4b5ce2fe28308fd9f2a7baf3	50c180a9830bc1354858665	2012-12-08
1348	eccbc87e4b5ce2fe28308fd9f2a7baf3	50c180a9830bc1354858665	2012-12-08
1349	eccbc87e4b5ce2fe28308fd9f2a7baf3	50c180a9830bc1354858665	2012-12-08
1350	eccbc87e4b5ce2fe28308fd9f2a7baf3	50c180a9830bc1354858665	2012-12-08

1. Analisis aktivitas waktu akses

   Analisis aktivitas waktu akses digunakan untuk mengindikasikan kecenderungan waktu akses terhadap situs, sehingga dapat digunakan sebagai alat bantu pengambilan keputusan waktu belanja yang tepat dan waktu memasarkan barang, memberikan diskon, dan waktu *maintenance system*.

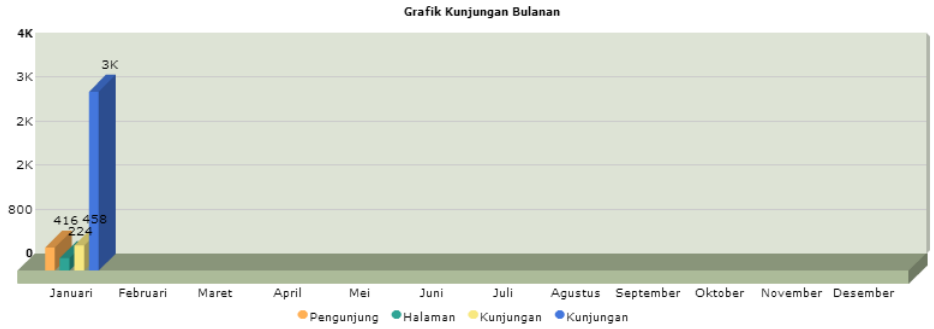
Dari data log kunjungan pada tabel 3.7 dan akan dihitung bagaimana proyeksi kunjungan terhadap waktu akses. Maka akan didapat informasi sebagai berikut:

Tabel 3. 8. Proyeksi Data Kunjungan Harian

Waktu	Pengunjung	Kunjungan
2012-11-01	2	2
2012-12-01	3	2
2012-12-02	2	2
2012-12-03	1	1
2012-12-04	6	3
2012-12-05	4	4
2012-12-06	1	1
2012-12-07	1	1
2012-12-08	1	1

Tabel 3. 9. Proyeksi Data Kunjungan Bulanan

Waktu	Pengunjung	Kunjungan
2012-11	2	2
2012-12	17	15



Gambar 3. 7. Contoh Proyeksi Data Kunjungan Bulanan

2. Analisis aktivitas akses terhadap barang

   Aktivitas akses terhadap barang mengindikasikan kecenderungan kesukaan pengunjung terhadap barang tertentu. Dengan mengetahui *trend* akses terhadap barang akan membantu pihak Campus Fashion Centre untuk melakukan pengambilan keputusan dalam hal pengadaan barang baru.

Data kunjungan terhadap barang didapatkan dari data *log* yang telah dilakukan proses transformasi terlebih dahulu, berikut adalah proses transformasi data log menjadi data kunjungan barang:

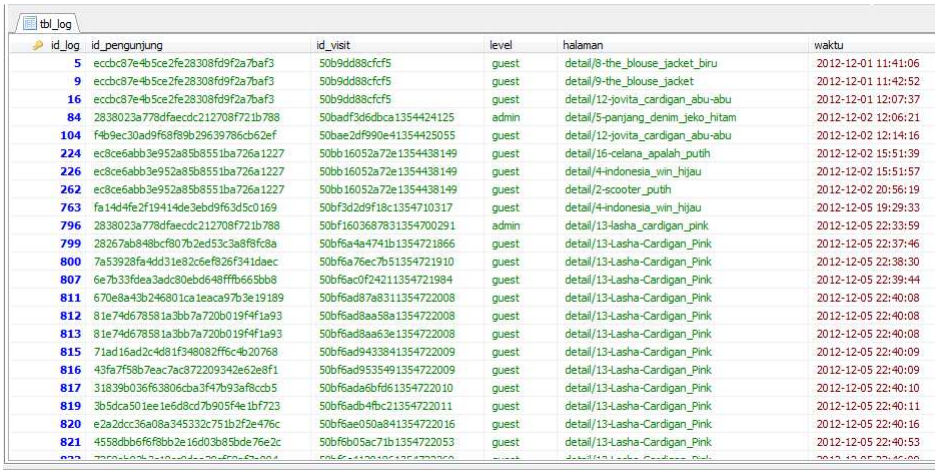
a. Data *source* diambil dari data log yang berisi *field* id_log, id_pengunjung, id_visit, id_user, level, ip_add, waktu, halaman, browser, *platform*.
b. Untuk proses proyeksi kunjungan terhadap data barang diambil beberapa *field* yang digunakan, yaitu: id_log, id_pengunjung, id_visit, level,halaman, dan waktu akses. Data setelah proses seleksi dapat dilihat pada gambar 3.7

tbl_log

id_log	id_pengunjung	id_visit	level	halaman	waktu
1	c81e728d9d4c2f636f067f89cc14862c	50b9eb56eaca61354361686	admin	beranda	2012-11-01 11:36:10
2	c81e728d9d4c2f636f067f89cc14862c	50b9eb56eaca61354361686	admin	kategori/1-T-Shirt	2012-12-01 11:36:27
3	c4ca4238a0b923820dcc509a6f75849b	50b9eb417492b1354361665	guest	beranda	2012-11-01 11:37:59
4	c4ca4238a0b923820dcc509a6f75849b	50b9eb417492b1354361665	guest	faq/tracking	2012-12-01 11:38:01
5	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	detail/8-the_blouse_jacket_biru	2012-12-01 11:41:06
6	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	template/bootstrap-1/js/jquery.jqzoom.js	2012-12-01 11:41:06
7	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	beranda	2012-12-01 11:41:06
8	c4ca4238a0b923820dcc509a6f75849b	50b9ece2a72151354362082	guest	beranda	2012-12-01 11:41:22
9	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	detail/9-the_blouse_jacket	2012-12-01 11:42:52
10	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	template/bootstrap-1/js/jquery.jqzoom.js	2012-12-01 11:42:52
11	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	beranda	2012-12-01 11:42:52
12	c81e728d9d4c2f636f067f89cc14862c	50b9ee4f085901354362447	admin	beranda	2012-12-01 11:47:27
13	c81e728d9d4c2f636f067f89cc14862c	50b9ee4f085901354362447	admin	asset/design/content.gif	2012-12-01 11:54:38
14	c81e728d9d4c2f636f067f89cc14862c	50b9ee4f085901354362447	admin	beranda	2012-12-01 11:54:38
15	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	wanita	2012-12-01 12:07:26
16	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	detail/12-jovita_cardigan_abu-abu	2012-12-01 12:07:37
17	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	template/bootstrap-1/js/jquery.jqzoom.js	2012-12-01 12:07:37
18	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	beranda	2012-12-01 12:07:38
19	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	warna/9-abu-abu	2012-12-01 12:07:50
20	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	kategori/1-T-Shirt	2012-12-01 12:08:01
21	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9f36666eb01354363750	guest	beranda	2012-12-01 12:09:10
22	c4ca4238a0b923820dcc509a6f75849b	50b9ece2a72151354362082	guest	beranda	2012-12-01 12:11:25
23	c81e728d9d4c2f636f067f89cc14862c	50b9ee4f085901354362447	admin	asset/design/content.gif	2012-12-01 12:35:38
24	c81e728d9d4c2f636f067f89cc14862c	50b9ee4f085901354362447	admin	beranda	2012-12-01 12:35:38
25	c81e728d9d4c2f636f067f89cc14862c	50b9ee4f085901354362447	admin	beranda	2012-12-01 14:04:19

Gambar 3. 8. Data Source Hasil Seleksi

c. Setelah dilakukan proses seleksi data, data yang dianggap tidak bermanfaat dan tidak memiliki makna dihilanggkan. Dalam kasus ini disimpan merupakan data yang memiliki isi *field* halaman dengan pola detail/[angka]-[teks], data selain pola tersebut akan dihilangkan.

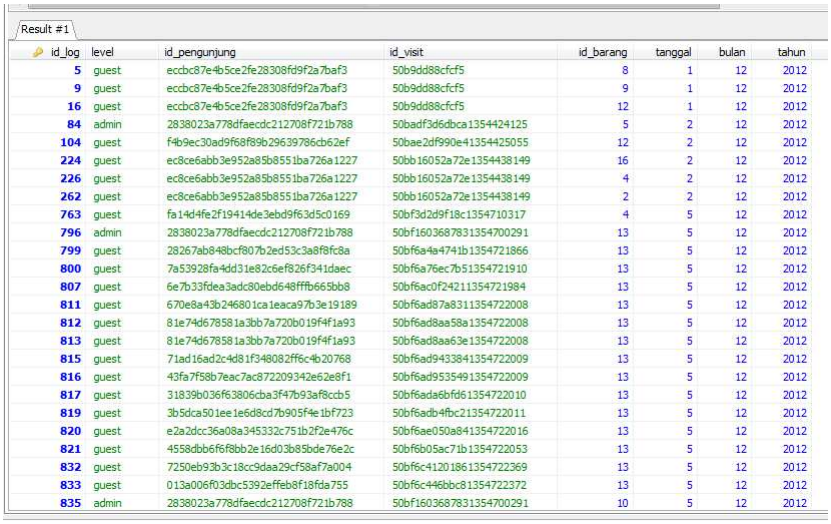


tbl_log

id_log	id_pengunjung	id_visit	level	halaman	waktu
5	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	detail/8-the_blouse_jacket_biru	2012-12-01 11:41:06
9	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	detail/9-the_blouse_jacket	2012-12-01 11:42:52
16	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	guest	detail/12-jovita_cardigan_abu-abu	2012-12-01 12:07:37
84	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	admin	detail/5-panjang_denim_jeko_hitam	2012-12-02 12:06:21
104	f4b9ec30ad9f68f89b29639786cb62ef	50bae2df990e41354425055	guest	detail/12-jovita_cardigan_abu-abu	2012-12-02 12:14:16
224	ec8ce6abb3e952a85b8551ba726a1227	50bb16052a72e1354438149	guest	detail/16-celana_apalah_putih	2012-12-02 15:51:39
226	ec8ce6abb3e952a85b8551ba726a1227	50bb16052a72e1354438149	guest	detail/4-indonesia_win_hijau	2012-12-02 15:51:57
262	ec8ce6abb3e952a85b8551ba726a1227	50bb16052a72e1354438149	guest	detail/2-scooter_putih	2012-12-02 20:56:19
763	fa14d4fe2f19414de3ebd9f63d5c0169	50bf3d2d9f18c1354710317	guest	detail/4-indonesia_win_hijau	2012-12-05 19:29:33
796	2838023a778dfaecdc212708f721b788	50bf1603687831354700291	admin	detail/13-lasha_cardigan_pink	2012-12-05 22:33:59
799	28267ab848bcf807b2ed53c3a8f8fc8a	50bf6a4a4741b1354721866	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:37:46
800	7a53928fa4dd31e82c6ef826f341daec	50bf6a76ec7b51354721910	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:38:30
807	6e7b33fdea3adc80ebd648fffb665bb8	50bf6ac0f24211354721984	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:39:44
811	670e8a43b246801ca1eaca97b3e19189	50bf6ad87a8311354722008	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:08
812	81e74d678581a3bb7a720b019f4f1a93	50bf6ad8aa58a1354722008	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:08
813	81e74d678581a3bb7a720b019f4f1a93	50bf6ad8aa63e1354722008	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:08
815	71ad16ad2c4d81f348082ff6c4b20768	50bf6ad9433841354722009	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:09
816	43fa7f58b7eac7ac872209342e62e8f1	50bf6ad9535491354722009	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:09
817	31839b036f63806cba3f47b93af8ccb5	50bf6ada6bfd61354722010	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:10
819	3b5dca501ee1e6d8cd7b905f4e1bf723	50bf6adb4fbc21354722011	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:11
820	e2a2dcc36a08a345332c751b2f2e476c	50bf6ae050a841354722016	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:16
821	4558dbb6f6f8bb2e16d03b85bde76e2c	50bf6b05ac71b1354722053	guest	detail/13-Lasha-Cardigan_Pink	2012-12-05 22:40:53

Gambar 3. 9. Data Setelah Proses Filtering

d. Data dalam tahap ini sudah dianggap konsisten, namun masih belum dapat diproyeksikan menjadi aktivitas kunjungan barang, *field* halaman akan ditransformasi menjadi id barang dengan memecah *field* halaman dengan pola detail/[id_barang]-[teks] maka diambil id barang, dan juga merubah format waktu menjadi format tanggal. Pada gambar 3.9 adalah data setelah proses transformasi.



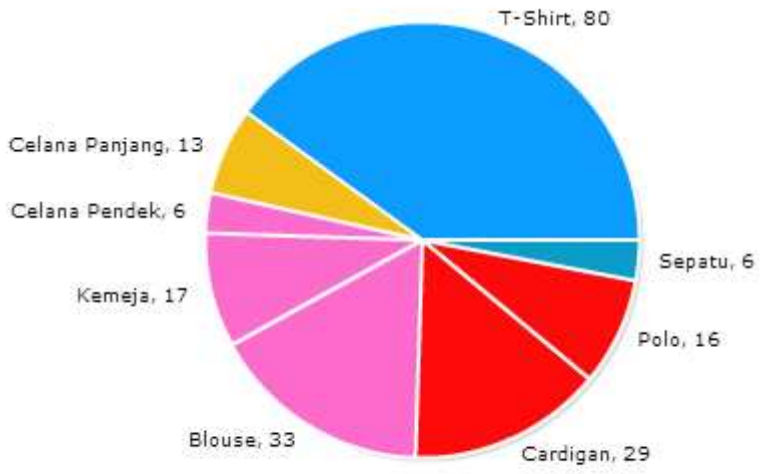
Result #1

id_log	level	id_pengunjung	id_visit	id_barang	tanggal	bulan	tahun
5	guest	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	8	1	12	2012
9	guest	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	9	1	12	2012
16	guest	eccbc87e4b5ce2fe28308fd9f2a7baf3	50b9dd88cfcf5	12	1	12	2012
84	admin	2838023a778dfaecdc212708f721b788	50badf3d6dbca1354424125	5	2	12	2012
104	guest	f4b9ec30ad9f68f89b29639786cb62ef	50bae2df990e41354425055	12	2	12	2012
224	guest	ec8ce6abb3e952a85b8551ba726a1227	50bb16052a72e1354438149	16	2	12	2012
226	guest	ec8ce6abb3e952a85b8551ba726a1227	50bb16052a72e1354438149	4	2	12	2012
262	guest	ec8ce6abb3e952a85b8551ba726a1227	50bb16052a72e1354438149	2	2	12	2012
763	guest	fa14d4fe2f19414de3ebd9f63d5c0169	50bf3d2d9f18c1354710317	4	5	12	2012
796	admin	2838023a778dfaecdc212708f721b788	50bf1603687831354700291	13	5	12	2012
799	guest	28267ab848bcf807b2ed53c3a8f8fc8a	50bf6a4a4741b1354721866	13	5	12	2012
800	guest	7a53928fa4dd31e82c6ef826f341daec	50bf6a76ec7b51354721910	13	5	12	2012
807	guest	6e7b33fdea3adc80ebd648fffb665bb8	50bf6ac0f24211354721984	13	5	12	2012
811	guest	670e8a43b246801ca1eaca97b3e19189	50bf6ad87a8311354722008	13	5	12	2012
812	guest	81e74d678581a3bb7a720b019f4f1a93	50bf6ad8aa58a1354722008	13	5	12	2012
813	guest	81e74d678581a3bb7a720b019f4f1a93	50bf6ad8aa63e1354722008	13	5	12	2012
815	guest	71ad16ad2c4d81f348082ff6c4b20768	50bf6ad9433841354722009	13	5	12	2012
816	guest	43fa7f58b7eac7ac872209342e62e8f1	50bf6ad9535491354722009	13	5	12	2012
817	guest	31839b036f63806cba3f47b93af8ccb5	50bf6ada6bfd61354722010	13	5	12	2012
819	guest	3b5dca501ee1e6d8cd7b905f4e1bf723	50bf6adb4fbc21354722011	13	5	12	2012
820	guest	e2a2dcc36a08a345332c751b2f2e476c	50bf6ae050a841354722016	13	5	12	2012
821	guest	4558dbb6f6f8bb2e16d03b85bde76e2c	50bf6b05ac71b1354722053	13	5	12	2012
832	guest	7250eb93b3c18cc9daa29cf58af7a004	50bf6c412018611354722369	13	5	12	2012
833	guest	013a006f03dbc5392effeb8f18fda755	50bf6c446bbc81354722372	13	5	12	2012
835	admin	2838023a778dfaecdc212708f721b788	50bf1603687831354700291	10	5	12	2012

Gambar 3. 10. Data Hasil Transformasi

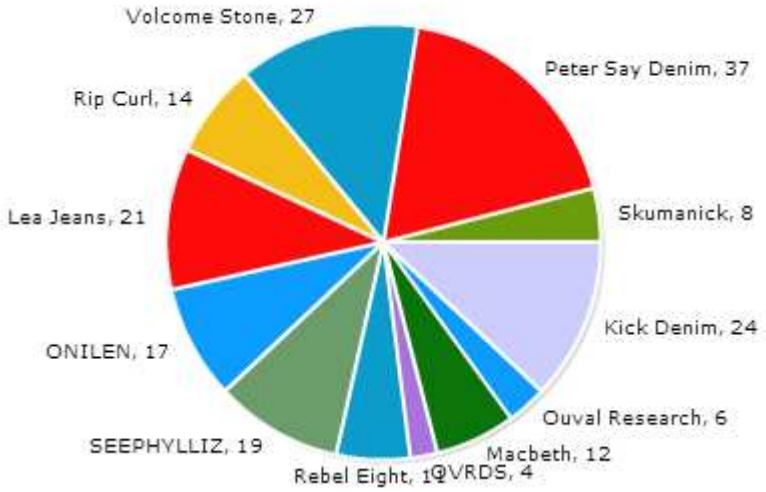
Setelah diperoleh data kunjungan barang, data akan diproyeksikan berdasarkan pengelompokan kunjungan berdasarkan data kategori, data

merk, dan data supplier, dengan menggunakan standart pengukuran hit, kunjungan, dan pengunjung.



Gambar 3. 11. Proyeksi Kunjungan Barang Terhadap Kategori

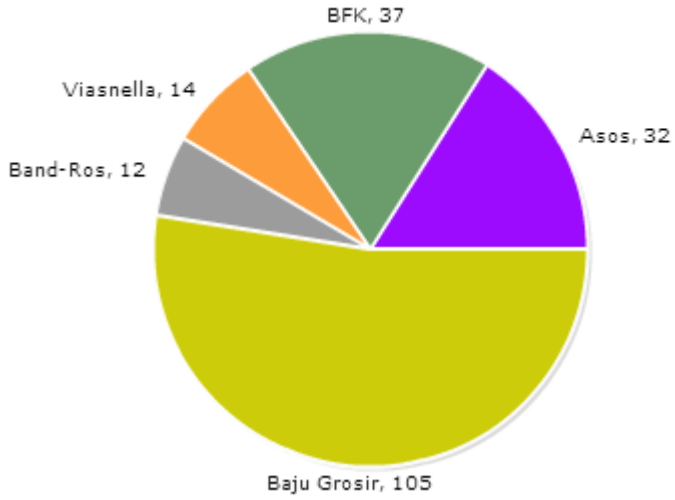
Data kunjugan barang di proyeksikan terhadap jenis kategori barang, hal ini dibutuhkan untuk keperluan menentukan keputusan belanja barang kedepannya, sehingga akan membantu pimpinan untuk menentukan kategori produk apa yang menjadi *trend* yang sering dikunjungi oleh pengunjung. Dari grafik pada gambar 3.11 dapat diambil kesimpulan bahwa T-Shirt adalah kategori yang paling sering dikunjungi maka untuk kedepannya bisa dijadikan pertimbangan untuk belanja barang dengan kategori T-Shirt lebih banyak.



Gambar 3. 12. Proyeksi Kunjungan Barang Terhadap Merk

Data kunjungan diproyeksikan terhadap jenis merk barang. Hal ini dibutuhkan untuk membantu memilah merk apa yang menjadi *trend* dalam waktu tertentu, dari grafik pada gambar 3.12 dapat dilihat merk Peter Say Denim, Volcome Stone dan Kick Denim menjadi tiga besar merk yang paling sering dikunjungi, dengan demikian dapat menjadi masukan bagi pimpinan untuk memilih ke tiga merk tersebut untuk keperluan belanja mendatang.

Data kunjungan barang juga diproyeksikan terhadap suplier, dengan pemetaan data kunjungan barang terhadap suplier diharapkan dapat membantu memetakan suplier yang potensial, sehingga untuk kedepannya akan dapat dijadikan parameter dalam menentukan keputusan saat berbelanja. Pada grafik gambar 3.13 dapat dilihat bahwa Baju Grosir merupakan suplier dengan nilai kunjungan paling besar, dengan demikian akan dapat disimpulkan bahwa Baju Grosir merupakan suplier yang potensial dilihat dari parameter kunjungan.

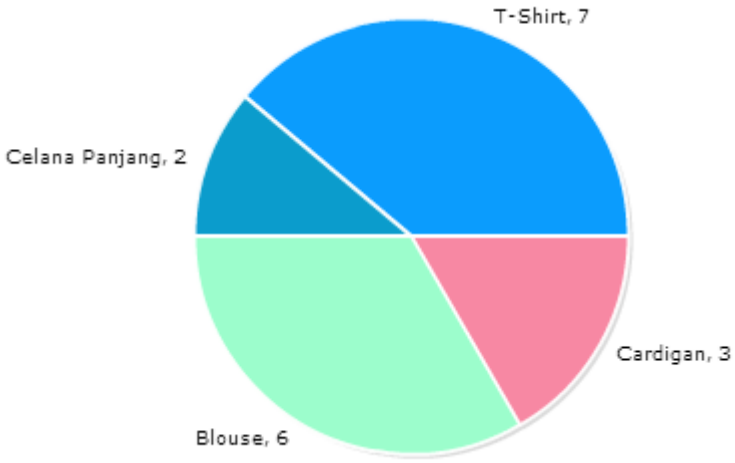


Gambar 3. 13. Proyeksi Kunjungan Terhadap Suplier

3. Analisis aktivitas penjualan

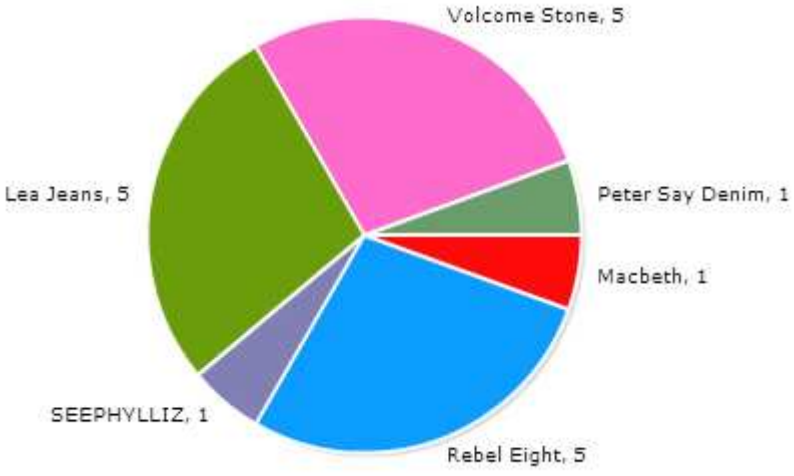
   Penjualan merupakan data yang paling dapat dijadikan tolak ukur *trend* belanja konsumen, sehingga dengan data penujualan didapatkan informasi mengenai kecenderungan belanja member, yang digunakan sebagai pembanding terhadap data proyeksi kunjungan barang.

   Aktivitas penjualan barang diproyeksikan terhadap jenis kategori barang tertentu. Pada proyeksi kunjungan barang kategori t-shirt mendominasi kunjungan barang, dan jika dilihat pada gambar 3.14 aktivitas penjualan kategori t-shirt juga menjadi *trend* penjualan. Dengan demikian dapat dikatakan bahwa kategori t-shirt menjadi trend pada periode tersebut, dan hal ini dapat dijadikan acuan bagi pimpinan untuk periode kedepan berbelanja kategori t-shirt lebih diutamakan.



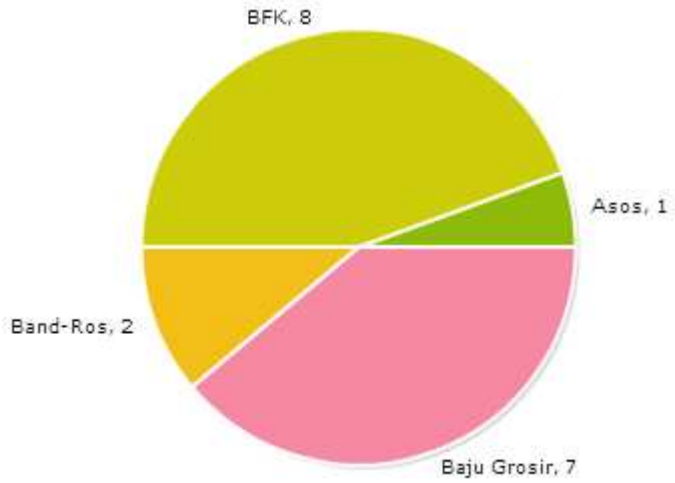
Gambar 3. 14. Proyeksi Belanja Terhadap Kategori

Proyeksi belanja berdasarkan kategori dapat dilihat pada gambar 3.15, dari data tersebut tampak bahwa barang yang paling banyak terjual adalah Volcome Stone, Lea Jeans, Rebel Eight, sedangkan dari data kunjungan barang yang paling sering diakses oleh pengunjung berasal dari merk Peter Say Denim, Volcome dan Kick Denim, sehingga data tersebut dapat diambil kesimpulan bahwa untuk belanja memperhatikan beberapa merk tersebut.



Gambar 3. 15. Proyeksi Belanja Terhadap Kategori

Aktivitas penjualan juga diproyeksikan terhadap suplier untuk melihat suplier mana yang cukup potensial jika dilihat dari penjualan tiap barang, sehingga untuk belanja kedepan dapat dijadikan pertimbangan dalam berbelanja. Dari gambar 3.16 dapat dilihat bahwa BFK dan Baju Grosir merupakan suplier yang potensial jika dilihat berdasarkan aktivitas belanja dalam waktu tertentu.

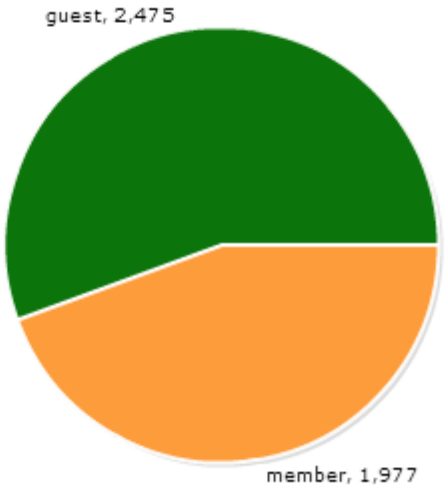


Gambar 3. 16. Proyeksi Aktivitas Penjualan Terhadap Suplier

4. Analisis aktivitas berdasarkan profil pengunjung

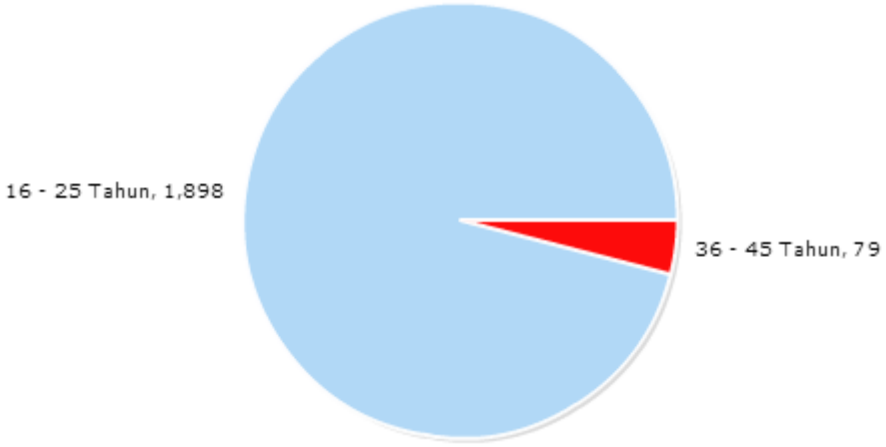
   Untuk memaksimalkan proses pemasaran barang, harus mengetahui bagaimana profil pengunjung yang datang berkunjung pada situs, dengan demikian akan memudahkan dalam menentukan strategi pemasaran.

   Pada grafik gambar 3.17 dapat dilihat bahwa kunjungan terbesar adalah dari yang bukan member, dari data tersebut maka perlu dilakukan pengambilan keputusan untuk mengajak pengunjung yang belum mendaftar untuk mendaftar sebagai member, sehingga pada akhirnya akan menambah peluang untuk berbelanja.



Gambar 3. 17. Proyeksi Kunjungan Berdasarkan Level Pengunjung

Segment penjualan barang mulai dari remaja hingga dewasa, sehingga dianggap perlu untuk mengetahui bagaimana *range* usia member yang mengunjungi situs, hal itu dapat dilihat pada grafik gambar 3.18, dari informasi grafik didapat informasi, bahwa *member* yang mengunjungi situs sebagian besar berada pada *range* usia 16-25tahun, sehingga perlu dipertimbangkan juga untuk belanja periode kedepan lebih menekankan pada *mode* untuk range usia 16-25tahun.



Gambar 3. 18. Proyeksi Kunjungan Berdasarkan Level Pengunjung

Untuk tujuan pemasaran perlu dipertimbangkan untuk mengetahui dari mana asal daerah *member* yang berkunjung ke situs, hal ini akan berguna pada saat target perluasan pasar, dengan demikian akan tau wilayah mana yang potensial dijadikan target pemasaran.



Gambar 3. 19. Proyeksi Kunjungan Member berdasarkan Wilayah

3.1.5 Analisis Kebutuhan Fungsional

Analisis fungsional merupakan tindakan mengidentifikasi proses yang akan berjalan didalam sistem yang akan dikembangkan dan menjelaskan kebutuhan yang diperlukan. Sehingga sistem dapat berjalan seperti yang diharapkan.

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

3.1.5.1 Identifikasi Aktor

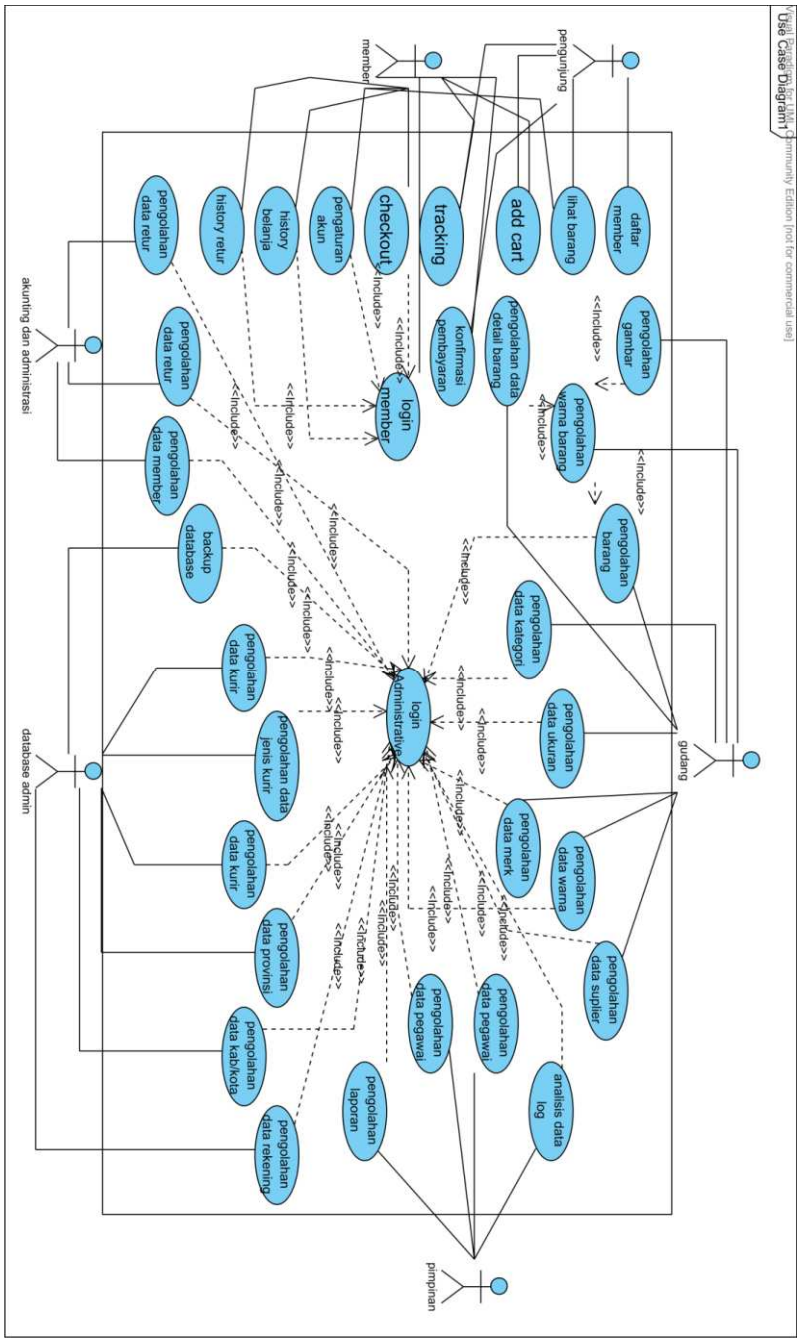
Aktor yang diidentifikasi terlibat dalam E-Commerce Campus Fashion Centre yaitu *database* administrator, pimpinan, bagian gudang, akunting dan administrasi, member dan pengunjung.

3.1.5.2 Diagram *Use case*

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 aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar.

Gambar 3.20 adalah diagram *use case* untuk *e-commerce* pada Campus Fashion Centre.



Gambar 3. 20. Diagram *Use Case*

3.1.5.3 Skenario *Use Case*

Skenario *use case* menjelaskan interaksi antara aktor dan *use case* dalam melakukan sebuah proses tertentu. Berikut adalah skenario *use case* untuk Campus Fashion Centre berdasarkan diagram *use case* pada gambar 3.20.

1. Skenario Use Case *Login* Administratif

   Interaksi antara aktor akunting dan administrasi, *database* administrator, gudang, dan pimpinan dengan *use case login* Administratif dapat dilihat pada tabel 3.10.

Tabel 3. 10. Skenario *Use Case Login* Administratif

Identifikasi			
Nama	*Login* Administratif		
Tujuan	Validasi Untuk mengakses hak khusus bagi administrator		
Deskripsi	Validasi atau autentikasi sistem untuk setiap aktor, untuk memperoleh hak khusus sebagai administrator.		
Aktor	*database* administrator, gudang, akunting dan administrasi, pimpinan		
Skenario Utama			
Kondisi Awal	Sistem menampilkan *form login* administratif		
Aksi Aktor		Reaksi Sistem	
1	Mengisi *form login* administratif	2	Melakukan validasi data masukan, dan mengecek ke *database* pegawai.
		3	Bila validasi berhasil dan data ditemukan di *database* maka akan menuju menu utama, tergantung hak akses masing–masing aktor.
Skenario Alternatif (*login* gagal)			
Aksi Aktor		Reaksi Sistem	
		4	tetap berada pada *form login* dan proses *login* gagal.
Kondisi Akhir	Berhasil masuk ke dalam sistem dan menampilkan menu sesuai hak akses masing–masing aktor		

2. Skenario *Use Case Login Member*

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

Tabel 3. 11. Skenario *Use Case Login Member*

Identifikasi			
Nama	*Login Member*		
Tujuan	Validasi Untuk mengakses hak khusus sebagai *member*		
Deskripsi	Validasi atau autentikasi untuk setiap aktor, untuk memperoleh hak khusus sebagai *member*.		
Aktor	*member*		
Skenario Utama			
Kondisi Awal	Sistem menampilkan *form login member*		
Aksi Aktor		Reaksi Sistem	
1	Mengisi *form login member*	2	Melakukan validasi data masukan, dan mengecek ke *database member*.
		3	Bila validasi berhasil dan data ditemukan di *database* maka akan menuju halaman utama *member*.
Skenario Alternatif (*login* gagal)			
Aksi Aktor		Reaksi Sistem	
		4	tetap berada pada *form login* dan menampilkan pesan kesalahan.
Kondisi Akhir	Berhasil masuk ke dalam sistem dan menuju halaman utama *member*.		

3. Skenario *Use Case* Pengolahan Data Rekening

   Interaksi antara aktor *database* administrator dengan *use case* pengolahan data rekening dapat dilihat pada tabel 3.12.

Tabel 3. 12. Skenario Use Case Pengolahan Data Rekening

Identifikasi			
Nama	Pengolahan Data Rekening		
Tujuan	Mengelola data rekening		
Deskripsi	proses mengelola data rekening yang digunakan untuk transaksi, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	*database* administrator		
Skenario Utama			
Kondisi Awal	menampilkan data rekening beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah rekening.	2	Menampilkan *form* tambah rekening.
3	Mengisi *form* rekening dan	4	Validasi data masukan, jika data

	menekan tombol simpan.		berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data rekening, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data rekening
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data rekenig.	6	Menampilkan form ubah data rekening yang akan diubah.
7	Mengubah *field* yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman rekening, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data rekening.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data rekening.	10	Menampilkan pesan konfirmasi, OK dan Cancel.
11	Memilih tombol OK pada pesan konfirmasi	12	Data rekening terpilih akan dihapus dari *database*, dan kembali ke halaman data rekening.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman menu rekening.
Kondisi Akhir	Pengolahan data rekening berhasil		

4. Skenario *Use Case* Pengolahan Data Provinsi

   Interaksi antara aktor *database* administrator dengan *use case* pengolahan data provinsi dapat dilihat pada tabel 3.13.

Tabel 3. 13. Skenario *Use Case* Pengolahan Data Provinsi

Identifikasi		
Nama	Pengolahan Data Provinsi	
Tujuan	Mengelola data provinsi	
Deskripsi	proses mengelola data provinsi yang digunakan untuk pengolahan data alamat, mencakup proses lihat, tambah, ubah, dan hapus data.	
Aktor	*database* administrator	
Skenario Utama		
Kondisi Awal	menampilkan data provinsi beserta menu untuk operasi.	
Aksi Aktor		Reaksi Sistem

1	Memilih menu tambah provinsi.	2	Menampilkan *form* tambah provinsi.
3	Mengisi form provinsi dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data provinsi, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data provinsi
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data provinsi.	6	Menampilkan *form* ubah dan data provinsi yang akan diubah.
7	Mengubah *field* yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman provinsi, jika validasi gagal akan muncul pesan kesalahan dan tetap pada form ubah data provinsi.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data provinsi.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol ya pada pesan konfirmasi	12	Data provinsi terpilih akan dihapus dari *database* dan kembali ke halaman data provinsi.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman provinsi.
Kondisi Akhir	Pengolahan data provinsi berhasil		

5. Skenario *Use Case* Pengolahan Data Kab/Kota

Interaksi antara aktor *database* administrator dengan *use case* pengolahan data kab/kota dapat dilihat pada tabel 3.14.

Tabel 3. 14. Skenario *Use Case* Pengolahan Data Kab/Kota

Identifikasi	
Nama	Pengolahan Data Kab/Kota
Tujuan	Mengelola data kab/kota
Deskripsi	proses mengelola data kab/kota yang digunakan untuk pengolahan data alamat, mencakup proses lihat, tambah, ubah, hapus dan cari data.
Aktor	*database* administrator
Skenario Utama	

Kondisi Awal	menampilkan data kab/kota beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah kab/kota.	2	Menampilkan *form* tambah kab/kota.
3	Mengisi form kab/kota dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data kab/kota, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data kab/kota
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data kab/kota.	6	Menampilkan *form* ubah dan data kab/kota yang akan diubah.
7	Mengubah *field* yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman kab/kota, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data kab/kota.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data kab/kota.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data kab/kota terpilih akan dihapus dari database dan kembali ke halaman data kab/kota.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman kab/kota.
Skenario Alternatif (cari data)			
14	Masukkan kata kunci yang akan dicari pada *field* cari, dan tekan tombol cari.	15	Sistem akan mengakses *database* dan mencari berdasarkan kata kunci yang dimasukkan, jika data ditemukan maka akan ditampilkan.
Kondisi Akhir	Pengolahan data kab/kota berhasil		

6. Skenario *Use Case* Pengolahan Data Kurir

   Interaksi antara aktor *database* administrator dengan *use case* pengolahan data kurir dapat dilihat pada tabel 3.15.

Tabel 3. 15. Skenario *Use Case* Pengolahan Data Kurir

Identifikasi			
Nama	Pengolahan Data Kurir		
Tujuan	Mengelola data kurir		
Deskripsi	proses mengelola data kurir yang digunakan untuk pengolahan data pengiriman , mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	*database* administrator		
Skenario Utama			
Kondisi Awal	menampilkan data kurir beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah kurir.	2	Menampilkan *form* tambah kurir.
3	Mengisi *form* kurir dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data kurir, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data kurir
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data kurir.	6	Menampilkan *form* ubah dan data kurir yang akan diubah.
7	Mengubah *field* yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman kurir, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data kurir.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data kurir.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data kurir terpilih akan dihapus beserta data jenis kurir dan ongkir kurir yang dibawahnya dari *database* dan kembali ke halaman data kurir.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman kurir.
Kondisi Akhir	Pengolahan data kurir berhasil		

7. Skenario *Use Case* Pengolahan Data Jenis Kurir

   Interaksi antara aktor *database* administrator dengan *use case* pengolahan data jenis kurir dapat dilihat pada tabel 3.16.

Tabel 3. 16. Skenario Use Case Pengolahan Data Jenis Kurir

Identifikasi			
Nama	Pengolahan Data Jenis Kurir		
Tujuan	Mengelola data jenis kurir		
Deskripsi	proses mengelola data jenis kurir yang digunakan untuk pengolahan data pengiriman, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	*database* administrator		
Skenario Utama			
Kondisi Awal	menampilkan data jenis kurir beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah jenis kurir.	2	Menampilkan *form* tambah jenis kurir.
3	Mengisi *form* jenis kurir dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data kurir, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data jenis kurir
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data jenis kurir.	6	Menampilkan *form* ubah dan data jenis kurir yang akan diubah.
7	Mengubah *field* yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman jenis kurir, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data jenis kurir.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data jenis kurir.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data jenis kurir terpilih akan dihapus dari *database* dan kembali ke halaman data jenis kurir.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman jenis kurir.
Kondisi Akhir	Pengolahan data jenis kurir berhasil		

8. Skenario *Use Case* Pengolahan Data Ongkir Kurir

Interaksi antara aktor *database* administrator dengan *use case* pengolahan data ongkir kurir dapat dilihat pada tabel 3.17.

Tabel 3. 17. Skenario *Use Case* Pengolahan Ongkir Kurir

Identifikasi			
Nama	Pengolahan Data Ongkir Kurir		
Tujuan	Mengelola data ongkir kurir		
Deskripsi	proses mengelola data ongkir kurir yang digunakan untuk pengolahan data pengiriman, mencakup proses lihat, tambah, ubah, hapus dan cari data.		
Aktor	*database* administrator		
Skenario Utama			
Kondisi Awal	menampilkan data ongkir kurir beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah ongkir kurir.	2	Menampilkan *form* tambah ongkir kurir.
3	Mengisi *form* ongkir kurir dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data ongkir kurir, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data ongkir kurir
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data ongkir kurir.	6	Menampilkan *form* ubah dan data ongkir kurir yang akan diubah.
7	Mengubah *field* yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman ongkir kurir, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data ongkir kurir.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data ongkir kurir.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data ongkir kurir terpilih akan dihapus dari *database* dan kembali

			ke halaman data ongkir kurir.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman ongkir kurir.
Skenario Alternatif (cari data)			
14	Masukkan kata kunci yang akan dicari pada field cari, dan tekan tombol cari.	15	Sistem akan mengakses *database* dan mencari berdasarkan kata kunci yang dimasukkan, jika data ditemukan maka akan ditampilkan, jika tidak ditemukan akan menampilkam pemberitahuan.
Kondisi Akhir	Pengolahan data ongkir kurir berhasil		

9. Skenario *Use Case* Backup Data

   Interaksi antara aktor *database* administrator dengan *use case backup database* dapat dilihat pada tabel 3.18.

Tabel 3. 18. Skenario *Use Case Backup* Data

Identifikasi			
Nama	*Backup Database*		
Tujuan	menduplikasi data yang ada pada *database*, untuk mengantisipasi ada kerusakan data.		
Deskripsi	proses mengduplikasi data pada semua database dan disimpan pada file berformat mysql.		
Aktor	*database* administrator		
Skenario Utama			
Kondisi Awal	menampilkan menu *backup database*		
Aksi Aktor		Reaksi Sistem	
1	Pada menu *backup*, pilih tombol *backup*.	2	Membaca data yang ada pada MySQL, dan menyalin data yang ada pada file berformat .zip.
		3	Menyimpan data file pada *hard drive*.
Kondisi Akhir	Data pada *database* berhasil di *backup* menjadi file .zip.		

10. Skenario *Use Case* Pengolahan Data Suplier

Interaksi antara aktor bagian gudang dengan *use case* pengolahan data suplier dapat dilihat pada tabel 3.19.

Tabel 3. 19. Skenario *Use Case* Pengolahan Data Suplier

Identifikasi			
Nama	Pengolahan Data Suplier		
Tujuan	Mengelola data suplier		
Deskripsi	proses mengelola data suplier yang digunakan untuk pengolahan data barang, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data suplier beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah suplier.	2	Menampilkan *form* tambah suplier.
3	Mengisi *form* suplier dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data suplier, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data supplier.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data suplier.	6	Menampilkan *form* ubah dan data suplier yang akan diubah.
7	Mengubah *field* yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman suplier, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data suplier.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data suplier.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data suplier terpilih akan dihapus dari *database* dan kembali ke halaman data suplier.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman suplier.
Kondisi Akhir	Pengolahan data suplier berhasil		

11. Skenario *Use Case* Pengolahan Data Merk

Interaksi antara aktor bagian gudang dengan *use case* pengolahan data merk dapat dilihat pada tabel 3.20.

Tabel 3. 20. Skenario Use Case Pengolahan Data Merk

Identifikasi			
Nama	Pengolahan Data Merk		
Tujuan	Mengelola data merk		
Deskripsi	proses mengelola data merk yang digunakan untuk pengolahan data barang, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data merk beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah merk.	2	Menampilkan *form* tambah merk.
3	Mengisi *form* merk dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data merk, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data merk.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data merk.	6	Menampilkan *form* ubah dan data merk yang akan diubah.
7	Mengubah merk yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman merk, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data merk.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data merk.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data merk terpilih akan dihapus dari *database* dan kembali ke halaman data merk.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman merk.
Kondisi Akhir	Pengolahan data merk berhasil		

12. Skenario *Use Case* Pengolahan Data Warna

Interaksi antara aktor bagian gudang dengan *use case* pengolahan data warna dapat dilihat pada tabel 3.21.

Tabel 3. 21. Skenario *Use Case* Pengolahan Data Warna

Identifikasi			
Nama	Pengolahan Data Warna		
Tujuan	Mengelola data warna		
Deskripsi	proses mengelola data warna yang digunakan untuk pengolahan data barang, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data warna beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah warna.	2	Menampilkan *form* tambah warna.
3	Mengisi *form* warna dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data warna, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data warna.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data warna.	6	Menampilkan *form* ubah dan data warna yang akan diubah.
7	Mengubah warna yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman warna, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data warna.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data warna.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data merk terpilih akan dihapus dari *database* dan kembali ke halaman data warna.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman warna.
Kondisi Akhir	Pengolahan data warna berhasil		

13. Skenario *Use Case* Pengolahan Data Kategori

Interaksi antara aktor bagian gudang dengan *use case* pengolahan data kategori dapat dilihat pada tabel 3.22.

Tabel 3. 22. Skenario *Use Case* Pengolahan Data Kategori

Identifikasi			
Nama	Pengolahan Data Kategori		
Tujuan	Mengelola data kategori		
Deskripsi	proses mengelola data kategori yang digunakan untuk pengolahan data barang, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data kategori beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah kategori.	2	Menampilkan *form* tambah kategori.
3	Mengisi *form* kategori dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data kategori, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data kategori.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data kategori.	6	Menampilkan *form* ubah dan data kategori yang akan diubah.
7	Mengubah kategori yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman kategori, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data kategori.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data kategori.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data kategori terpilih akan dihapus beserta data barang dan ukuran dibawahnya dari *database* dan kembali ke halaman data kategori.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman kategori.
Kondisi Akhir	Pengolahan data kategori berhasil		

14. Skenario *Use Case* Pengolahan Data Ukuran

Interaksi antara aktor bagian gudang dengan *use case* pengolahan data ukuran dapat dilihat pada tabel 3.23.

Tabel 3. 23. Skenario *Use Case* Pengolahan Data Ukuran

Identifikasi			
Nama	Pengolahan Data Ukuran		
Tujuan	Mengelola data ukuran		
Deskripsi	proses mengelola data ukuran yang digunakan untuk pengolahan data barang, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data ukuran beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah ukuran.	2	Menampilkan *form* tambah ukuran.
3	Mengisi *form* ukuran dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data ukuran, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data ukuran.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data ukuran.	6	Menampilkan *form* ubah dan data ukuran yang akan diubah.
7	Mengubah ukuran yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman ukuran, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data ukuran.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data ukuran.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data ukuran terpilih akan dihapus dari *database* dan kembali ke halaman data ukuran.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman ukuran.
Kondisi Akhir	Pengolahan data ukuran berhasil		

15. Skenario *Use Case* Pengolahan Data Barang

Interaksi antara aktor bagian gudang dengan use case pengolahan data barang dapat dilihat pada tabel 3.24.

Tabel 3. 24. Skenario *Use Case* Pengolahan Data Barang

Identifikasi			
Nama	Pengolahan Data Barang		
Tujuan	Mengelola data barang		
Deskripsi	proses mengelola data barang yang digunakan untuk pengolahan data proses transaksi, mencakup proses lihat, tambah, ubah, hapus dan cari data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data barang beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah barang.	2	Menampilkan *form* tambah barang.
3	Mengisi *form* barang dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data barang, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data barang.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data barang.	6	Menampilkan *form* ubah dan data barang yang akan diubah.
7	Mengubah barang yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman barang, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data barang.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data barang.	10	Menampilkan pesan konfirmasi, OK dan Cancel.
11	Memilih tombol OK pada pesan konfirmasi	12	Data barang terpilih akan dihapus dari *database* dan kembali ke halaman data barang.
12	Memilih tombol Cancel pada pesan konfirmasi.	13	Kembali ke halaman barang.
Skenario Alternatif (cari data)			

14	Masukkan kata kunci yang akan dicari pada field cari, dan tekan tombol cari.	15	Sistem akan mengakses *database* dan mencari berdasarkan kata kunci yang dimasukkan, jika data ditemukan maka akan ditampilkan, jika tidak ditemukan akan menampilkam pemberitahuan.
Kondisi Akhir	Pengolahan data barang berhasil		

16. Skenario *Use Case* Pengolahan Data Warna Barang

Interaksi antara aktor bagian gudang dengan *use case* pengolahan data warna barang dapat dilihat pada tabel 3.25.

Tabel 3. 25. Skenario *Use Case* Pengolahan Data Warna Barang

Identifikasi			
Nama	Pengolahan Data Warna Barang		
Tujuan	Mengelola data warna barang		
Deskripsi	proses mengelola data warna barang yang digunakan untuk pengolahan data proses transaksi, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data warna barang beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu warna barang.	2	Menampilkan *form* tambah warna barang.
3	Mengisi memilih warna barang dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan gambar di simpan dan akan mengarah pada halaman data warna barang, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data warna barang.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data warna barang.	6	Menampilkan *form* ubah dan data warna barang yang akan diubah.
7	Memilih warna baru, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman warna barang, jika

			validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data warna barang.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data warna barang.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data gambar barang terpilih akan dihapus dari *database* dan kembali ke halaman data warna barang.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman warna barang.
Kondisi Akhir	Pengolahan data warna barang berhasil		

17. Skenario *Use Case* Pengolahan Data Detail Barang

Interaksi antara aktor bagian gudang dengan *use case* pengolahan data detail barang dapat dilihat pada tabel 3.26.

Tabel 3. 26. Skenario *Use Case* Pengolahan Data Detail Barang

Identifikasi			
Nama	Pengolahan Data Detail Barang		
Tujuan	Mengelola data detail barang		
Deskripsi	proses mengelola data detail barang yang digunakan untuk pengolahan data proses transaksi, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data detail barang beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah detail barang.	2	Menampilkan *form* tambah detail barang.
3	Mengisi memilih detail barang barang dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil disimpan ke *database* dan akan mengarah pada halaman data detail barang, jika validasi gagal akan muncul pesan kesalahan, tetap berada *form* tambah data detail barang.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan	6	Menampilkan *form* ubah data

	diubah pada data detail barang.		detail barang yang akan diubah.
7	Mengubah *field* detail baru, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman detail barang, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data detail barang.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data detail barang.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data detail barang terpilih akan dihapus dari *database* dan kembali ke halaman data detail barang.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman detail barang.
Kondisi Akhir	Pengolahan data detail barang berhasil		

18. Skenario *Use Case* Pengolahan Data Gambar Barang

Interaksi antara aktor bagian gudang dengan use case pengolahan data gambar barang dapat dilihat pada tabel 3.27.

Tabel 3. 27. Skenario *Use Case* Pengolahan Data Gambar

Identifikasi			
Nama	Pengolahan Data Gambar Barang		
Tujuan	Mengelola data gambar barang		
Deskripsi	proses mengelola data gambar barang yang digunakan untuk pengolahan data proses transaksi, mencakup proses lihat, tambah, dan hapus data.		
Aktor	Bagian Gudang		
Skenario Utama			
Kondisi Awal	menampilkan data gambar barang beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu gambar barang yang akan di unggah, tekan tombol tambah	2	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan gambar di simpan dan akan mengarah pada halaman data gambar barang, jika validasi gagal akan muncul pesan kesalahan.
Skenario Alternatif (hapus data)			

3	Memilih data yang akan dihapus pada data gambar barang.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data gambar barang terpilih akan dihapus dari *database* dan kembali ke halaman data gambar barang.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman gambar barang.
Kondisi Akhir	Pengolahan data gambar barang berhasil		

19. Skenario *Use Case* Pengolahan Data Jabatan

Interaksi antara aktor pimpinan dengan *use case* pengolahan data jabatan dapat dilihat pada tabel 3.28.

Tabel 3. 28. Skenario *Use Case* Pengolahan Data Jabatan

Identifikasi			
Nama	Pengolahan Data Jabatan		
Tujuan	Mengelola data jabatan		
Deskripsi	proses mengelola data jabatan yang digunakan untuk pengolahan data pegawai, mencakup proses lihat, tambah, ubah, dan hapus data.		
Aktor	Pimpinan		
Skenario Utama			
Kondisi Awal	menampilkan data jabatan beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah jabatan.	2	Menampilkan *form* tambah jabatan.
3	Mengisi *form* jabatan dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data jabatan, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data jabatan.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data jabatan.	6	Menampilkan *form* ubah dan data jabatan yang akan diubah.
7	Mengubah jabatan yang perlu diubah, setelah selesai tekan tombol simpan.	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman jabatan, jika validasi gagal akan muncul pesan kesalahan dan tetap pada *form*

			ubah data jabatan.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data jabatan.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data jabatan terpilih akan dihapus dari *database* dan kembali ke halaman data jabatan.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman jabatan.
Kondisi Akhir	Pengolahan data jabatan berhasil		

20. Skenario *Use Case* Pengolahan Data Pegawai

Interaksi antara aktor pimpinan dengan *use case* pengolahan data pegawai dapat dilihat pada tabel 3.29.

Tabel 3. 29. Skenario *Use Case* Pengolahan Data Pegawai

Identifikasi			
Nama	Pengolahan Data Pegawai		
Tujuan	Mengelola data pegawai		
Deskripsi	proses mengelola data pegawai yang digunakan untuk pengolahan data, mencakup proses lihat, tambah, ubah, dan dan hapus data.		
Aktor	Pimpinan		
Skenario Utama			
Kondisi Awal	menampilkan data pegawai beserta menu untuk operasi.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu tambah pegawai.	2	Menampilkan *form* tambah pegawai.
3	Mengisi *form* pegawai dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil masuk ke *database* dan akan mengarah pada halaman data pegawai, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* tambah data pegawai.
Skenario Alternatif (ubah data)			
Aksi Aktor		Reaksi Sistem	
5	Memilih data yang akan diubah pada data pegawai.	6	Menampilkan *form* ubah dan data pegawai yang akan diubah.
7	Mengubah pegawai yang perlu diubah, setelah selesai tekan tombol	8	Validasi data, jika data berhasil divalidasi maka akan menuju halaman pegawai, jika validasi

	simpan.		gagal akan muncul pesan kesalahan dan tetap pada *form* ubah data pegawai.
Skenario Alternatif (hapus data)			
9	Memilih data yang akan dihapus pada data pegawai.	10	Menampilkan pesan konfirmasi, OK dan *Cancel*.
11	Memilih tombol OK pada pesan konfirmasi	12	Data pegawai terpilih akan dihapus dari *database* dan kembali ke halaman data pegawai.
12	Memilih tombol *Cancel* pada pesan konfirmasi.	13	Kembali ke halaman pegawai.
Kondisi Akhir	Pengolahan data pegawai berhasil		

21. Skenario *Use Case* Pengolahan Laporan

Interaksi antara aktor pimpinan dengan *use case* pengolahan laporan dapat dilihat pada tabel 3.30.

Tabel 3. 30. Skenario *Use Case* Pengolahan Laporan

Identifikasi			
Nama	Pengolahan Laporan		
Tujuan	Mengelola laporan tiap divisi		
Deskripsi	proses mengelola laporan, laporan yang dikelola mengenai transaksi penjulan dan stok barang.		
Aktor	Pimpinan		
Skenario Utama			
Kondisi Awal	menampilkan menu laporan.		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu laporan transaksi penjualan	2	sistem mengakses data penjualan ke *database* dan menampilkan data global transaksi.
3	Memilih laporan penjualan berdasarkan waktu tertentu.	4	sistem mengakses data penjualan ke *database* dan menampilkan berdasarkan waktu dipilih.
Skenario Alternatif (laporan barang)			
5	Memilih menu laporan barang	6	sistem mengakses data barang ke *database* dan menampilkan data global stok barang berdasarkan kategori.
7	Memilih laporan barang berdasarkan waktu tertentu.	8	sistem mengakses data barang ke *database* dan menampilkan berdasarkan waktu dipilih.
Kondisi Akhir	Pengolahan laporan berhasil		

22. Skenario *Use Case* Pengolahan Analisis *Log*

Interaksi antara aktor pengunjung dengan *use case* analisis *log* pimpinan dapat dilihat pada tabel 3.31.

Tabel 3. 31. Skenario *Use Case* Pengolahan Analisis *Log*

Identifikasi			
Nama	Analisis Log		
Tujuan	Melakukan pengolahan data log		
Deskripsi	Proses pimpinan melihat informasi yang didapatkan dari data log		
Aktor	Pimpinan		
Skenario Utama			
Kondisi Awal	Halaman awal		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu analisis *log*	2	Melakkan proses pengambilan data dari database, dan melakukan pengolahan dan menampilkan dalam bentuk grafik
Kondisi Akhir	Data log berhasil ditampilkan		

23. Skenario *Use Case* Daftar *Member*

Interaksi antara aktor pengunjung dengan *use case* daftar *member* dapat dilihat pada tabel 3.32.

Tabel 3. 32. Skenario *Use Case* Daftar *Member*

Identifikasi			
Nama	Daftar Member		
Tujuan	Melakukan proses daftar		
Deskripsi	Proses pengunjung mendaftar menjadi *member*, untuk dapat melakukan proses transaksi.		
Aktor	Pengunjung		
Skenario Utama			
Kondisi Awal	Halaman awal		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu daftar *member*.	2	Menampilkan *form* isian untuk daftar menjadi *member*.
3	Mengisi *form* yang disediakan, jika selesai tekan tombol daftar	4	Melakukan validasi input data, jika isian *form* tidak lengkap atau salah maka akan muncul pemberitahuan, dan tetap di menu *form* daftar

			member. Jika isian benar, data akan disimpan dan akan muncul pemberitahuan.
		5	Sistem akan membuat *notifikasi* melalui *email* yang digunakan untuk mendaftar.
Kondisi Akhir	Daftar member berhasil		

24. Skenario *Use Case* Lihat Barang

Interaksi antara aktor pengunjung dan *member* dengan *use case* lihat barang dapat dilihat pada tabel 3.33.

Tabel 3. 33. Skenario *Use Case* Lihat Barang

Identifikasi			
Nama	Lihat Barang		
Tujuan	Melihat detail barang		
Deskripsi	Proses melihat detail, untuk memudahkan konsumen mengetahui barang yang ada dijual.		
Aktor	Pengunjung, *Member*		
Skenario Utama			
Kondisi Awal	Halaman barang		
Aksi Aktor		Reaksi Sistem	
1	Memilih barang yang akan dilihat.	2	Mengakses *database* dan menampilkan detail barang yang dipilih.
Kondisi Akhir	Proses lihat barang berhasil		

25. Skenario *Use Case Tracking*

Interaksi antara aktor pengunjung dan *member* dengan *use case tracking* dapat dilihat pada tabel 3.34.

Tabel 3. 34. Skenario *Use Case Tracking*

Identifikasi	
Nama	*Tracking*
Tujuan	Mengecek status pengiriman pesanan
Deskripsi	Proses melihat status pengiriman barang yang telah dilakukan oleh member
Aktor	Pengunjung, *Member*
Skenario Utama	
Kondisi Awal	Halaman barang

Aksi Aktor		Reaksi Sistem	
1	Memilih menu *tracking*	2	Mengakses *database* dan menampilkan status pengiriman
Kondisi Akhir	Proses *tracking* berhasil		

26. Skenario *Use Case* Pengaturan Akun

Interaksi antara aktor *member* dengan *use case* pengaturan akun dapat dilihat pada tabel 3.35.

Tabel 3. 35. Skenario *Use Case* Pengaturan Akun

Identifikasi			
Nama	Pengaturan Akun		
Tujuan	Operasi ubah data pribadi		
Deskripsi	Proses mengelola akun berkaitan dengan operasi ubah data pribadi, yaitu:data *password* dan data pribadi		
Aktor	*Member*		
Skenario Utama			
Kondisi Awal	Halaman *member*		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu ubah data *password*	2	Menampilkan form ubah data *password*
3	Mengisi *form* isian dan menekan tombol simpan.	4	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil diubah dan akan mengarah pada halaman *member*, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* ubah *password*.
Skenario Alternatif (ubah data pribadi)			
Aksi Aktor		Reaksi Sistem	
5	Memilih menu ubah data pribadi	6	Menampilkan *form* ubah data pribadi
7	Mengisi *form* isian dan menekan tombol simpan.	8	Validasi data masukan, jika data berhasil divalidasi maka akan berhasil diubah dan akan mengarah pada halaman *member*, jika validasi gagal akan muncul pesan kesalahan dan tetap berada *form* ubah data pribadi.
Kondisi Akhir	Pengaturan akun berhasil		

27. Skenario *Use Case* Add Cart

Interaksi antara aktor pengunjung, *member* dengan *use case add cart* dapat dilihat pada tabel 3.36.

Tabel 3. 36. Skenario *Use Case Add Cart*

Identifikasi			
Nama	*Add Cart*		
Tujuan	Menambahkan barang ke keranjang belanja		
Deskripsi	Proses menambahkan keranjang belanja, untuk selanjutnya dilakukan transaksi.		
Aktor	*Member*, Pengunjung		
Skenario Utama			
Kondisi Awal	Halaman barang detail		
Aksi Aktor		Reaksi Sistem	
1	Memilih ukuran barang		
2	Tekan tombol beli	3	Nilai akan di simpan dalam *session* sementara, jika tidak dilakukan transaksi selanjutnya selama waktu tertentu maka *session* akan dihapus
4	Merubah kuantitas	5	Jika input berupa angka dan lebih kecil dari stok yang tersedia, maka keranjang belanja akan diubah, jika tidak maka akan ada pemberitahuan
Kondisi Akhir	Tambah ke keranjang belanja berhasil		

28. Skenario *Use Case* Checkout

Interaksi antara aktor *member* dengan *use case checkout* dapat dilihat pada tabel 3.37.

Tabel 3. 37. Skenario *Use Case Checkout*

Identifikasi	
Nama	*Checkout*
Tujuan	Proses pemesanan barang
Deskripsi	Menambahkan barang ke daftar belanja dan melakukan transaksi pembelian.
Aktor	Member
Skenario Utama	
Kondisi Awal	Halaman utama
Aksi Aktor	Reaksi Sistem

1	Memilih menu keranjang belanja	2	Menampilkan halaman keranjang belanja, jika keranjang belanja kosong maka akan ada pemberitahuan.
3	Melakukan konfirmasi pesanan, dengan mengecek apakah benar data yang dipesan.		
4	Mengcek jumlah belanja, dan merubah jika kurang atau salah. Jika telah selesai tekan tombol lanjut transaksi.	5	Melakukan kalkulasi mengenai jumlah pesanan. Dan menampilkan *form* tujuan pengiriman barang.
6	Mengisi form pengiriman barang, jika talah selesai tekan tombol lanjut.	7	Melakukan kalkulasi biaya ongkos kirim beserta biaya total transaksi. Dan menampilkan *form* cara pembayaran.
8	Mengisi *form* cara pembayaran, tekan tombol lanjut.	9	Menampilkan konfirmasi persetujuan transaksi.
10	Jika setuju tekan tombol *checkout*.	11	Data pesanan disimpan dalam *database* pemesanan barang selesai dan dapat melakukan transaksi menuju tahap selanjutnya, yaitu konfirmasi pembayaran.
Kondisi Akhir	Proses *checkout* berhasil		

29. Skenario *Use Case History* Belanja

Interaksi antara aktor *member* dengan *use case history* belanja dapat dilihat pada tabel 3.38.

Tabel 3. 38. Skenario *Use Case History* Belanja

Identifikasi		
Nama	*History* Belanja	
Tujuan	Mengelola transaksi belanja	
Deskripsi	*History* belanja merupakan proses yang berkaitan dengan transaksi membeli yang dilakukan *member*, meliputi proses mencari transaks hingga mengubah status transaksi belanja	
Aktor	*Member*	
Skenario Utama		
Kondisi Awal	Halaman utama *member*	
Aksi Aktor		Reaksi Sistem

1	Memilih menu *history* belanja	2	Menampilkan halaman *history* belanja secara keseluruhan.
3	klik id order	4	Mengakses *database* dan melakukan proses cari berdasarkan id order jika ditemukan akan menampilkan transaksi, jika tidak ditemukan akan muncul pemberitahuan
Skenario Alternatif (konfirmasi barang diterima)			
5	Klik transaksi berstatus "dikirim"	6	Menampilkan detail transaksi dipilih yang bestatus "dikirim" dan menampilkan konfirmasi ubah status menjadi "diterima"
7	Klik konfirmasi diterima	8	Melakukan perubahan status transaksi dari "dikirim" menjadi "diterima" dan kembali menuju halaman detail transaksi
Kondisi Akhir	Pengolahan *history* belanja berhasil		

30. Skenario *Use Case* Konfirmasi Pembayaran

Interaksi antara aktor *member* dan pengunjung dengan *use case* konfirmasi pembayaran dapat dilihat pada tabel 3.39.

Tabel 3. 39. Skenario *Use Case* Konfirmasi Pembayaran

Identifikasi			
Nama	Konfirmasi Pembayaran		
Tujuan	Proses konfirmasi pembayaran		
Deskripsi	*member* yang telah melakukan proses transaksi melakukan proses konfirmasi jika telah melakukan pembayaran, sehigga dapat diproses pengiriman		
Aktor	*Member*, Pengunjung		
Skenario Utama			
Kondisi Awal	Halaman utama		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu konfirmasi pembayaran	2	Menampilkan halaman konfirmasi pembayaran
3	Mengisi semua *field* yang disediakan klik tombol konfirmasi	4	Melakukan pengecekan di *database*, jika ditemukan, maka akan merubah status order menjadi konf, dan memberikan notifikasi, jika tidak ditemukan akan member pesan kesalahan
Kondisi Akhir	Konfirmasi pembayaran berhasil		

31. Skenario *Use Case History* Retur

Interaksi antara aktor *member* dengan *use case history* retur dapat dilihat pada tabel 3.40.

Tabel 3. 40. Skenario *Use Case* History Retur

Identifikasi			
Nama	*History* Retur		
Tujuan	Mengelola proses retur jika dibutuhkan		
Deskripsi	Proses retur merupakan proses mengembalikan barang yang telah dibeli karena hal – hal tertentu.		
Aktor	*Member*		
Skenario Utama			
Kondisi Awal	Halaman utama member		
Aksi Aktor		Reaksi Sistem	
1	Pilih menu *history* retur	2	Mengakses *database*, jika ada data retur maka akan menampilkan data retur
3	Klik tambah retur	4	Menampilkan *form* pencarian id order
5	Memasukkan id order yang akan dilakukan retur	6	Mengakses *database* dan melakukan pencarian, jika data ditemukan dan memenuhi kriteria untuk retur maka akan ditampilkan, jika tidak maka akan memberikan pemberitahuan
7	Mengisi *form* alasan retur pada barang akan yang diretur, dan kuantitas barang yang akan diretur, setelah selesai klik selesai	8	Melakukan penambahan data retur baru, menuju halaman detail retur
9	Klik edit pada detail barang retur	10	Tampil *form* data detail barang retur
11	Merubah data detail barang retur, dan menambahkan gambar, klik ubah jika selesai	12	Mengakses *database* dan merubah data detail barang, dan melakukan penambahan gambar, jika berhasil menuju halaman detail retur, jika gagal akan memberikan informasi kesalahan
Skenario Alternatif (status konfirmasi)			
		13	Menampilkan detail retur, jika detail barang retur ada yang diterima maka akan menampilkan

			form pengisian konfirmasi kirim barang retur, jika tidak maka akan menampilkan pesan retur ditolak
14	Mengisi form konfirmasi pengiriman, jika telah selesai klik simpan	15	Mengakses *database*, menambahkan data konfirmasi pengiriman dan status retur berubah menjadi kirim
Kondisi Akhir	Proses retur berhasil		

32. Skenario *Use Case* Pengolahan Data Penjualan

Interaksi antara aktor akunting dan administrasi dengan *use case* pengolahan data penjualan dapat dilihat pada tabel 3.41.

Tabel 3. 41. Skenario *Use Case* Pengolahan Data Penjualan

Identifikasi			
Nama	Pengolahan Data Penjualan		
Tujuan	Mengelola transaksi penjualan		
Deskripsi	Merupakan pengelolaan transaksi penjualan kepada *member*, meliputi pencarian dan pengelolaan status.		
Aktor	Akunting dan administrasi		
Skenario Utama			
Kondisi Awal	Halaman utama akunting dan administrasi		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu data penjualan	2	Menampilkan data transaksi penjualan secara global, dan form pencarian
3	Masukkan id order untuk dilakukan pencarian	4	Mengakses *database*, jika data ditemukan maka akan menampilkan data yang dicari, jika tidak akan diberikan pemberitahuan.
Skenario Alternatif (konfirmasi pembayaran diterima)			
5	Klik detail transaksi berstatus “konfirmasi bayar”	6	Menampilkan detail transaksi dipilih yang berstatus “konfirmasi bayar”, dan menampilkan tombol konfirmasi “terima” dan konfirmasi “tolak”
7	Klik konfirmasi “terima”.	8	Melakukan perubahan status transaksi dari konfirmasi bayar menjadi konfirmasi dibayar dan kembali ke detail transaksi.
9	Klik konfirmasi “tolak”	10	Melakukan perubahan status transaksi dari konfirmasi bayar

			menjadi konfirmasi “pesanan baru” dan kembali ke detail transaksi.
Skenario Alternatif (konfirmasi pengiriman barang)			
11	Klik detail transaksi berstatus “order bayar”	12	Menampilkan detail transaksi dipilih yang berstatus “dibayar”, dan menampilkan tombol konfirmasi pengiriman.
13	Klik “tambah detail pengiriman”	14	Menampilkan form pengisian detail transaksi pengiriman
15	Mengisi form detail transaksi,lalu tekan tombol simpan	16	Menyimpan data detail pengiriman barang, dan status transaksi berubah menjadi “dikirim”
Skenario Alternatif (konfirmasi barang diterima)			
17	Klik transaksi berstatus “dikirim”	18	Menampilkan detail transaksi dipilih yang bestatus “dikirim” dan menampilkan konfirmasi ubah status menjadi “diterima”
19	Klik “tambah detail pengiriman”	20	Menampilkan form pengisian detail transaksi pengiriman
19	Ubah status pengiriman menjadi diterima	20	Melakukan perubahan status transaksi dari “dikirim” menjadi “diterima” dan kembali menuju halaman detail transaksi
Kondisi Akhir	Pengolahan data penjualan berhasil		

33. Skenario *Use Case* Pengolahan Retur

Interaksi antara aktor akunting dan administrasi dengan *use case* pengelolaan retur dapat dilihat pada tabel 3.42.

Tabel 3. 42. Skenario *Use Case* Pegolahan Retur

Identifikasi			
Nama	Pengolahan retur		
Tujuan	Mengelola data retur yang dilakukan oleh *member*		
Deskripsi	Melakukan pengelolaan transaksi retur yang dilakukan *member*		
Aktor	akunting dan administrasi		
Skenario Utama			
Kondisi Awal	Halaman utama Akunting dan administrasi		
Aksi Aktor		Reaksi Sistem	
1	Memilih menu retur	2	Menampilkan menu retur, dan seluruh data retur yang dilakukan *member*.
3	Klik detail	4	Mengakses *database* dan mencari

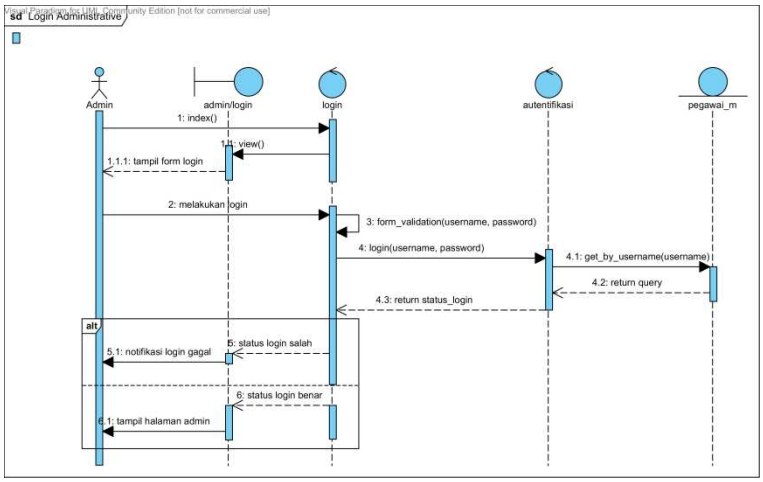
			data retur yang dicari, dan menampilkan data retur yang dicari
5	Melihat data detail barang yang di retur, dan alasan retur, klik ubah	6	Mengakses *database* dan menampilkan *form* ubah status barang retur
7	Mengubah status barang menjadi diterima untuk retur, atau ditolak untuk retur.	8	Sistem mengubah status barang retur menjadi ditolak atau diterima dan memberikan notifikasi
9	Mengubah status retur dari new menjadi konf	10	Sistem mengubah status retur dari new manjadi konf
Skenario Alternatif (konfirmasi pengiriman barang diterima)			
11	Mengubah status retur dari kirim menjadi terima	12	Sistem mengubah status retur dari kirim manjadi terima
Skenario Alternatif (konfirmasi pengiriman barang dikirim kembali)			
13	Mengisi form jasa kurir yang digunakan, dan no resi pengiriman dan merubah status menjadi kembali	14	Sistem mengubah status retur dari diterima menjadi kembali, dan menambahkan detail pengiriman
Kondisi Akhir	Pengelolan retur berhasil		

3.1.5.4 Diagram *Sequence*

Diagram *sequence* menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa pesan (*message*) yang digambarkan terhadap waktu. Diagram *sequence* untuk Campus Fashion Centre adalah sebagai berikut:

1. Diagram *Sequence Login* Administratif

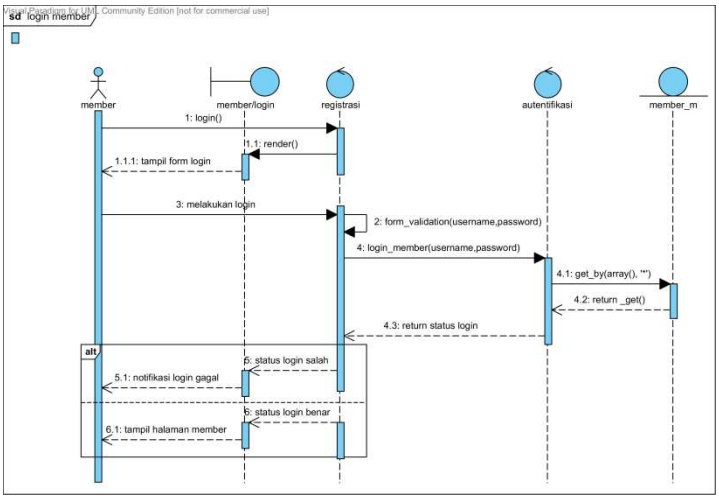
   Diagram *sequence login* admin dapat dilihat pada gambar 3.21.



Gambar 3. 21 Diagram *Sequence Login* Administratif

2. Diagram *Sequence Login* Member

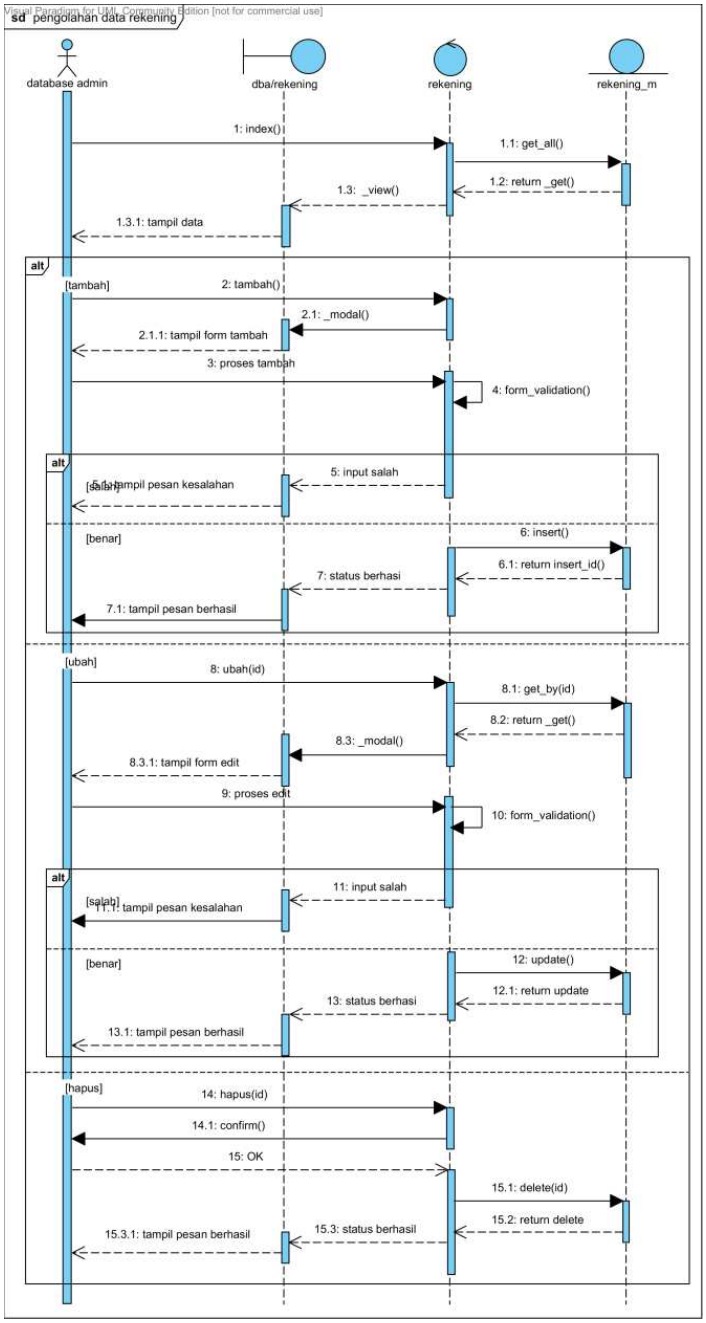
   Diagram *sequence login* member dapat dilihat pada gambar 3.22.



Gambar 3. 22 Diagram *Sequence Login* Member

3. Diagram *Sequence* Pengolahan Data Rekening

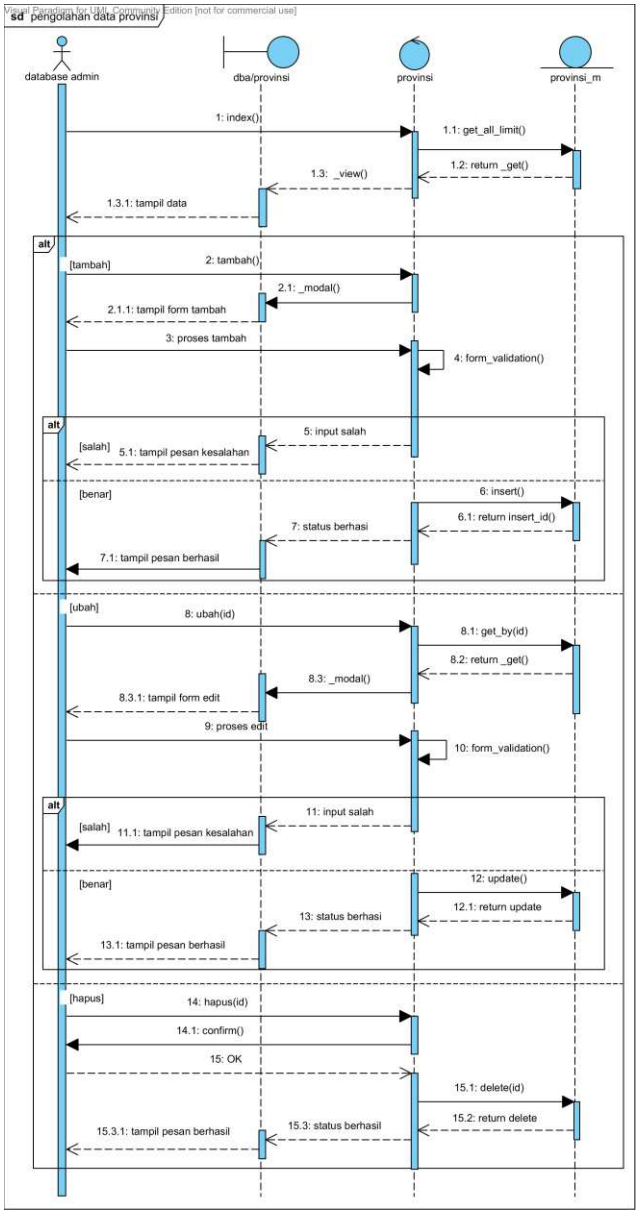
Diagram *sequence* pengolahan data rekening dapat dilihat pada gambar 3.23.



Gambar 3. 23 Diagram *Sequence* Pengolahan Data Rekening

4. Diagram *Sequence* Pengolahan Data Provinsi

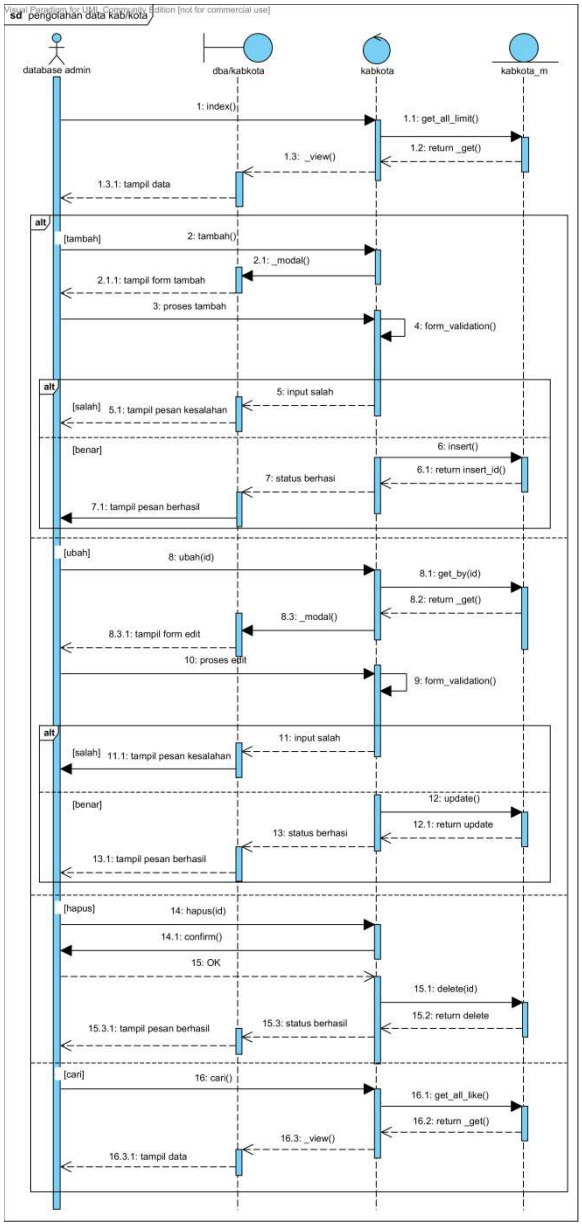
   Diagram *sequence* pengolahan data provinsi dapat dilihat pada gambar 3.24.



Gambar 3. 24 Diagram *Sequence* Pengolahan Data Provinsi

5. Diagram *Sequence* Pengolahan Data Kabupaten Kota

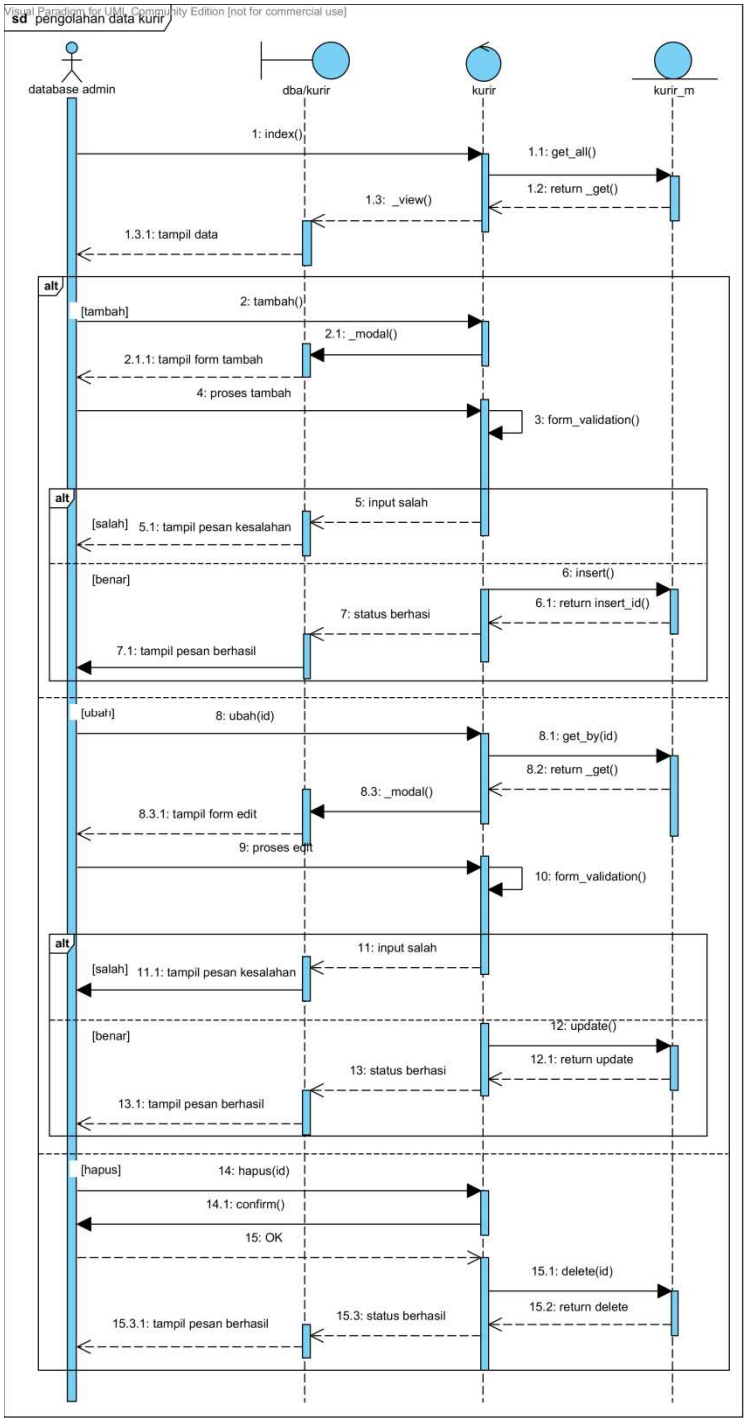
   Diagram *sequence* pengolahan data kabupaten kota dapat dilihat pada gambar 3.25.



Gambar 3. 25 Diagram *Sequence* Pengolahan Data Kabupaten Kota

6. Diagram *Sequence* Pengolahan Data Kurir

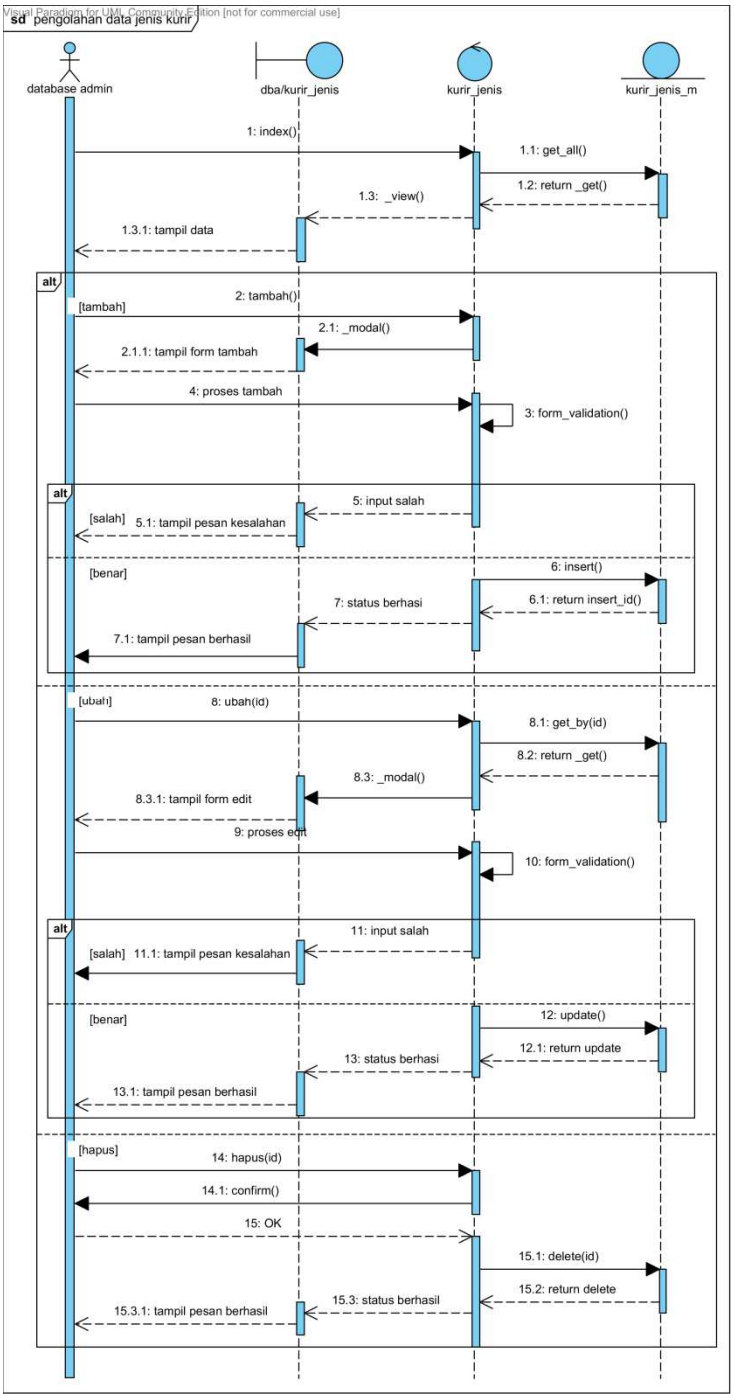
Diagram *sequence* pengolahan data kurir dapat dilihat pada gambar 3.26.



Gambar 3. 26 Diagram *Sequence* Pengolahan Data Kurir

7. Diagram *Sequence* Pengolahan Data Jenis Kurir

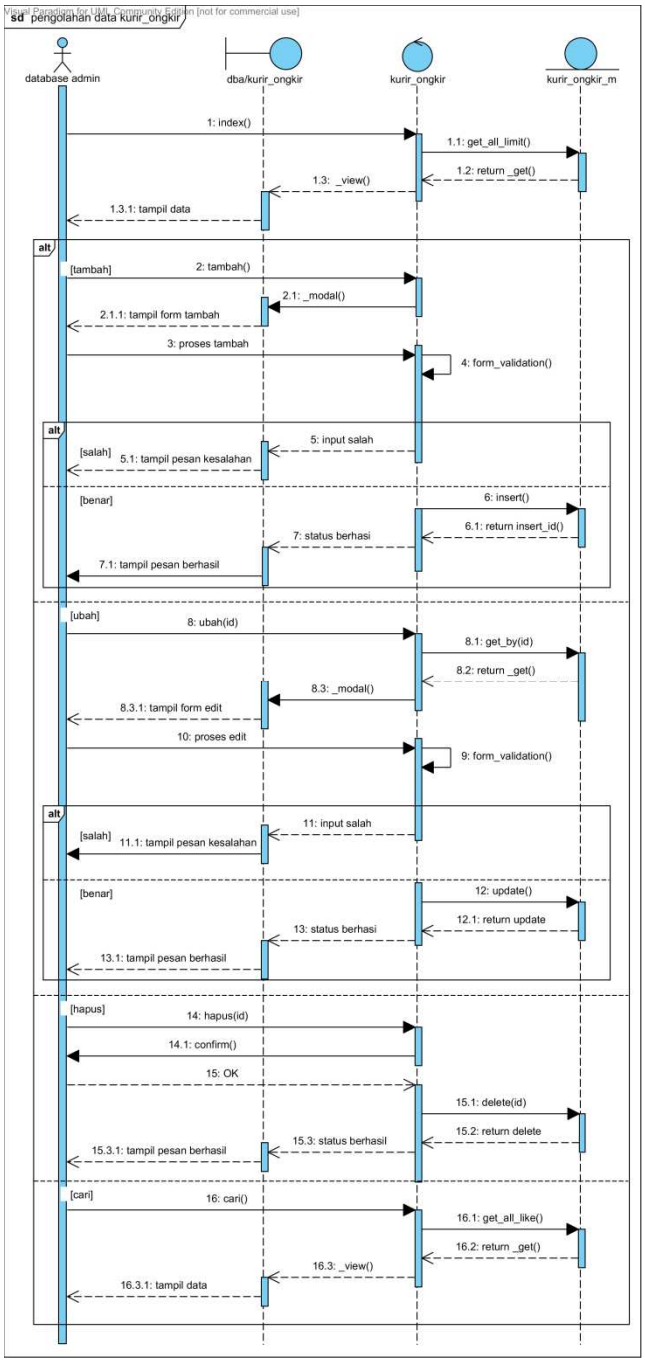
Diagram *sequence* pengolahan data jenis kurir dapat dilihat pada gambar 3.27.



Gambar 3. 27 Diagram *Sequence* Pengolahan Data Jenis Kurir

8. Diagram *Sequence* Pengolahan Data Ongkir Kurir

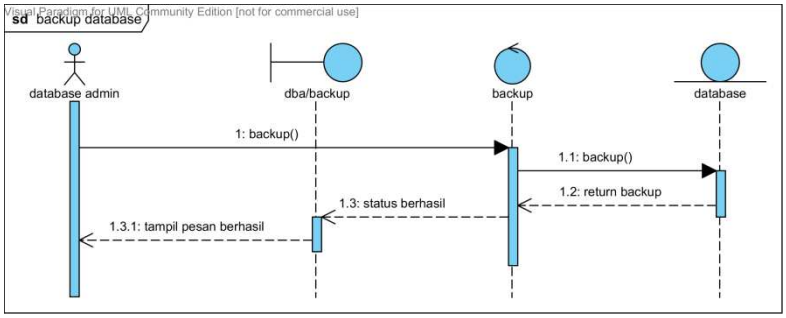
Diagram *sequence* pengolahan data ongkir kurir dapat dilihat pada gambar 3.28.



Gambar 3. 28 Diagram *Sequence* Pengolahan Data Ongkir Kurir

9. Diagram *Sequence* Backup Database

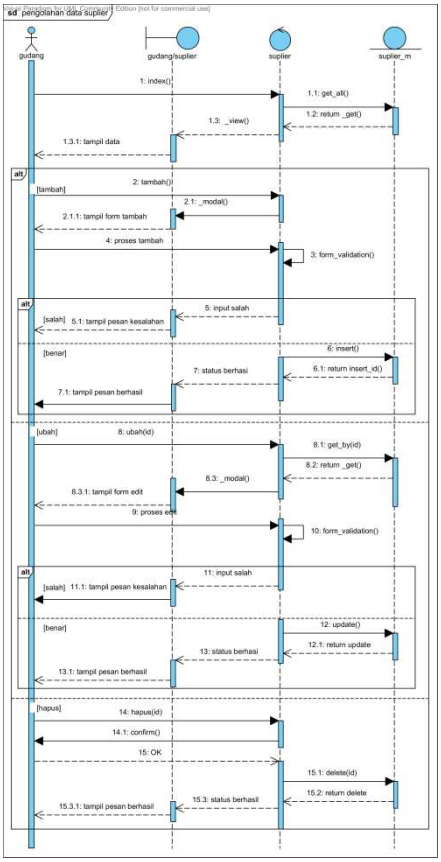
   Diagram *sequence backup* database dapat dilihat pada gambar 3.29.



Gambar 3. 29 Diagram *Sequence* Backup Database

10. Diagram *Sequence* Pengolahan Data Suplier

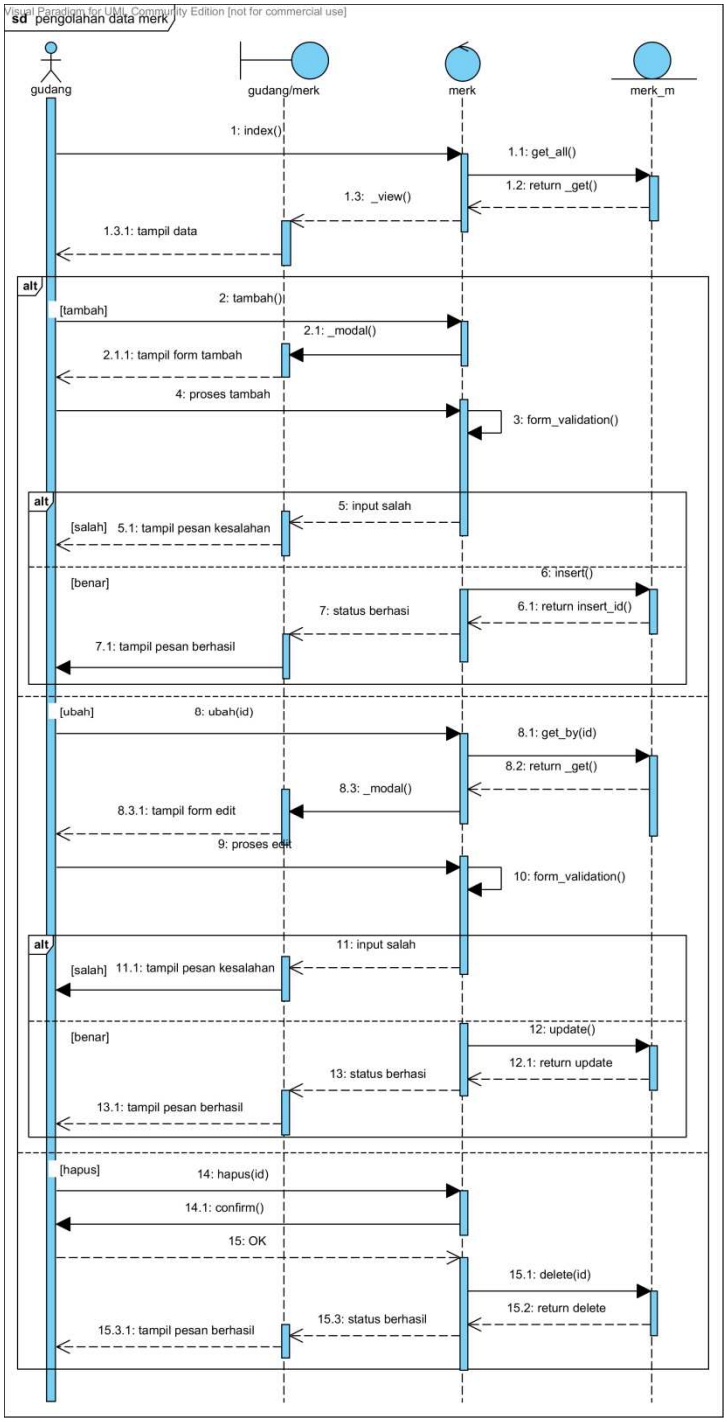
    Diagram *sequence* pengolahan data suplier dapat dilihat pada gambar 3.30.



Gambar 3. 30 Diagram *Sequence* Pengolahan Data Suplier

11. Diagram *Sequence* Pengolahan Data Merk

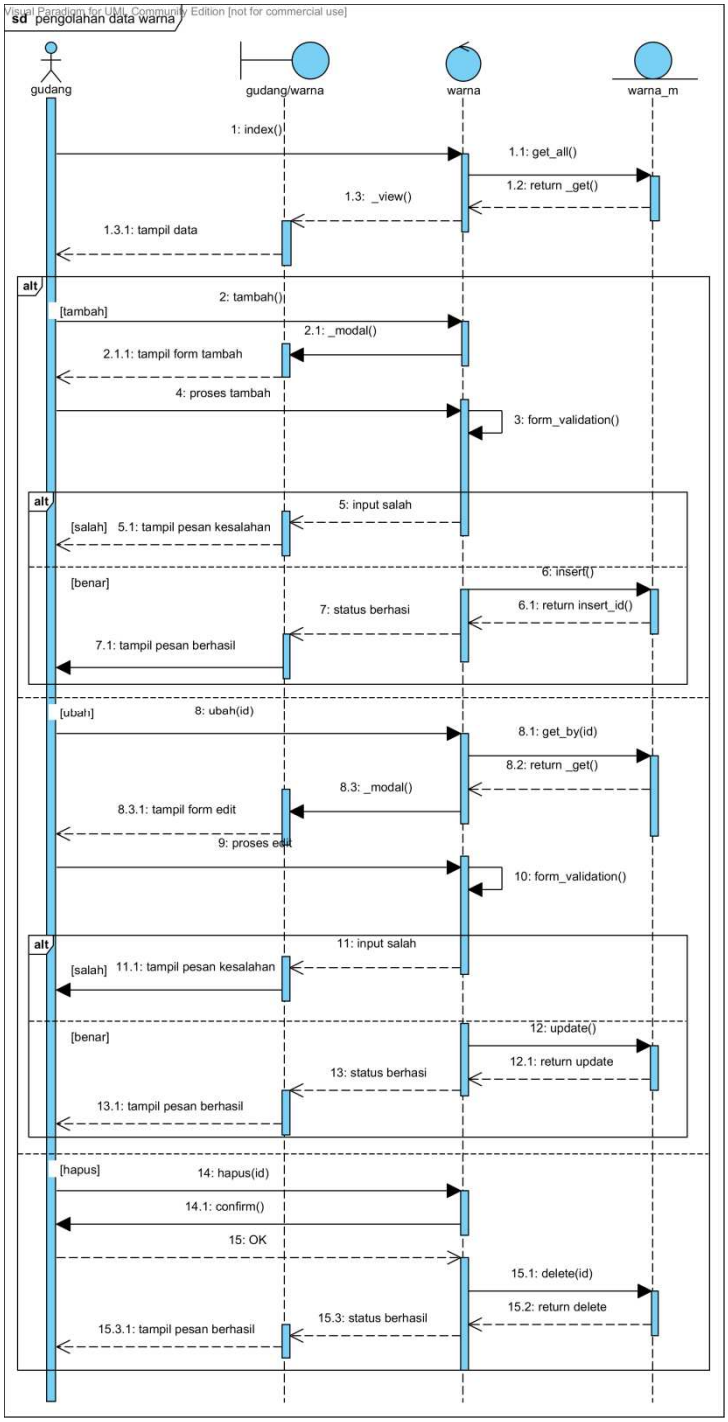
Diagram *sequence* pengolahan data merk dapat dilihat pada gambar 3.31.



Gambar 3. 31 Diagram *Sequence* Pengolahan Data Merk

12. Diagram *Sequence* Pengolahan Data Warna

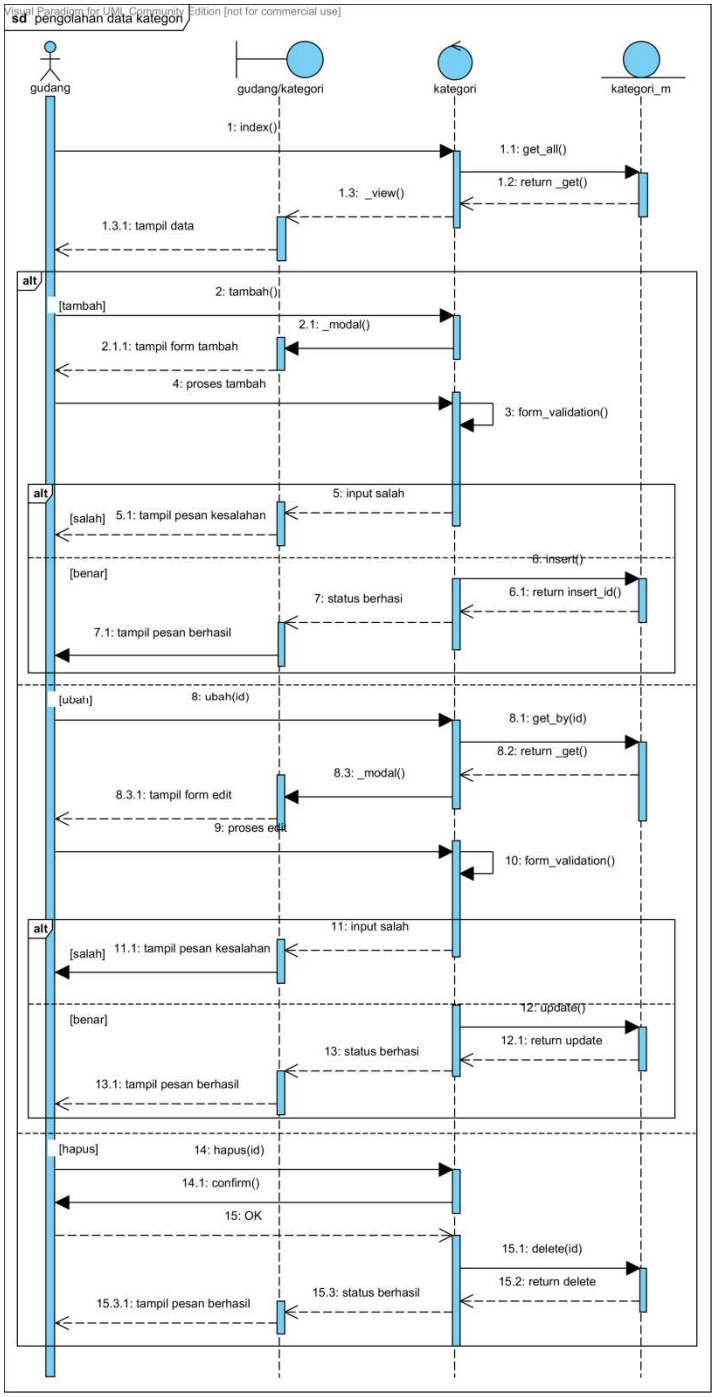
Diagram *sequence* pengolahan data warna dapat dilihat pada gambar 3.32.



Gambar 3. 32 Diagram *Sequence* Pengolahan Data Warna

13. Diagram *Sequence* Pengolahan Data Kategori

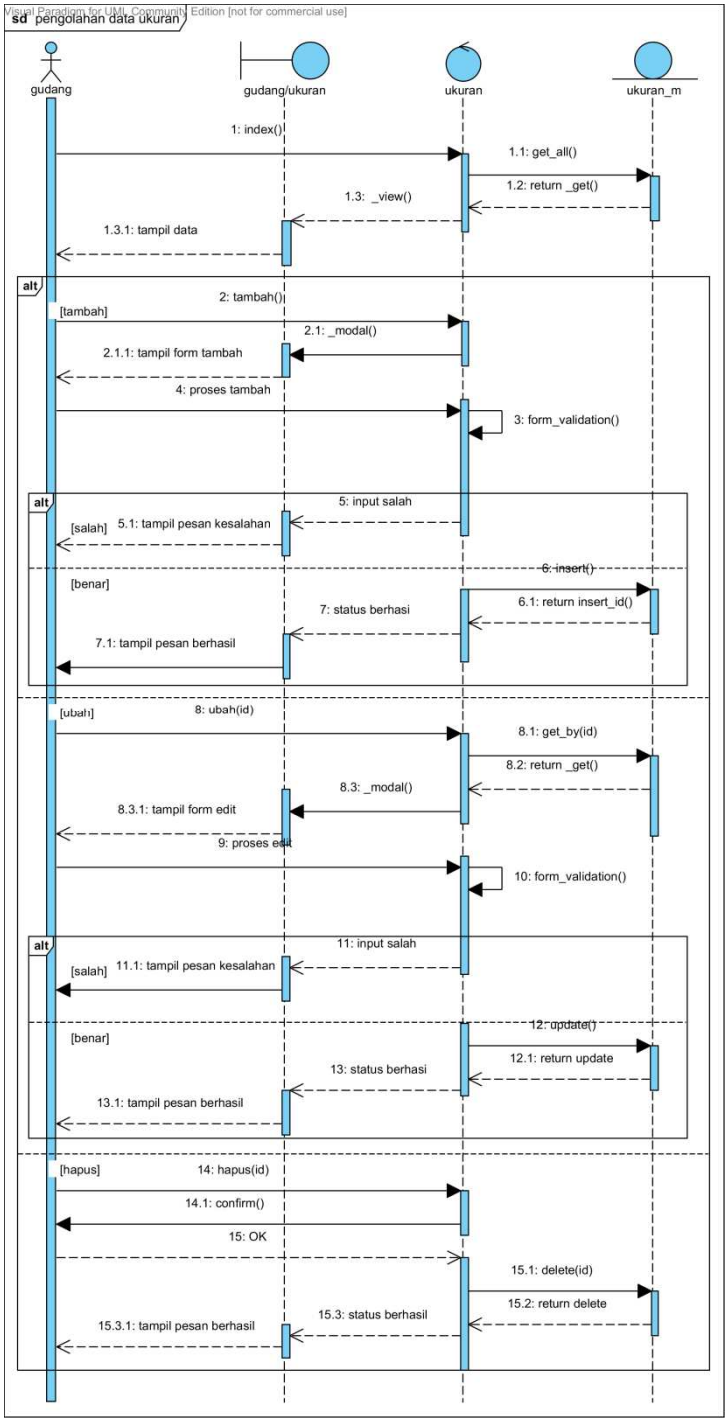
Diagram *sequence* pengolahan data kategori dapat dilihat pada gambar 3.33.



Gambar 3. 33 Diagram *Sequence* Pengolahan Kategori

14. Diagram *Sequence* Pengolahan Data Ukuran

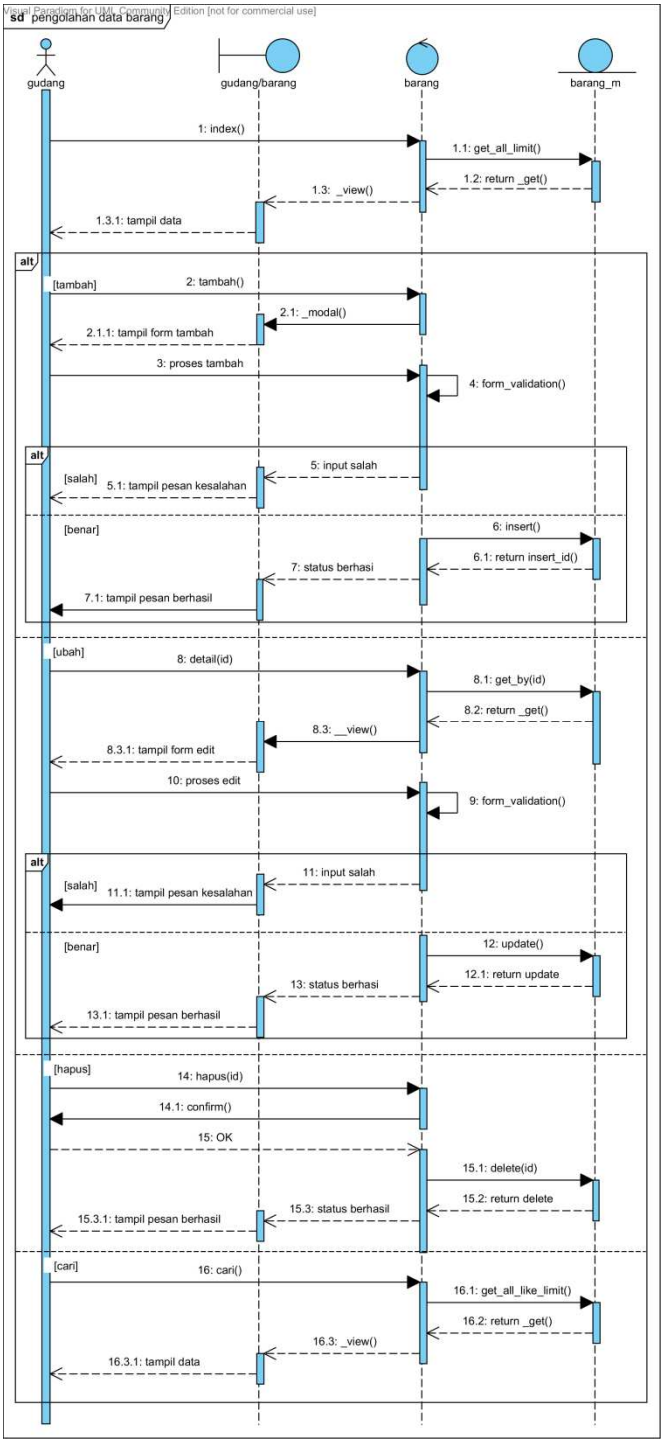
Diagram *sequence* pengolahan data ukuran dapat dilihat pada gambar 3.34.



Gambar 3. 34 Diagram *Sequence* Pengolahan Data Ukuran

15. Diagram *Sequence* Pengolahan Data Barang

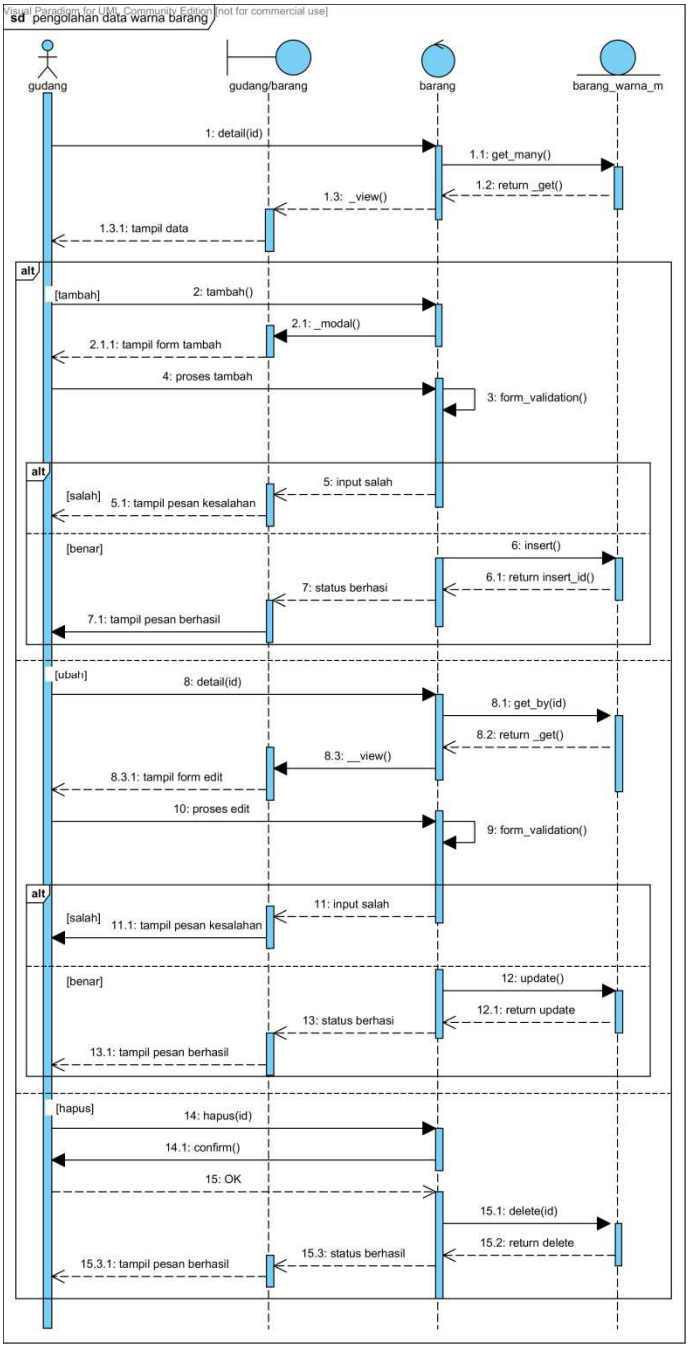
Diagram *sequence* pengolahan data barang dapat dilihat pada gambar 3.35.



Gambar 3. 35 Diagram Sequence Pengolahan Data Barang

16. Diagram *Sequence* Pengolahan Data Warna Barang

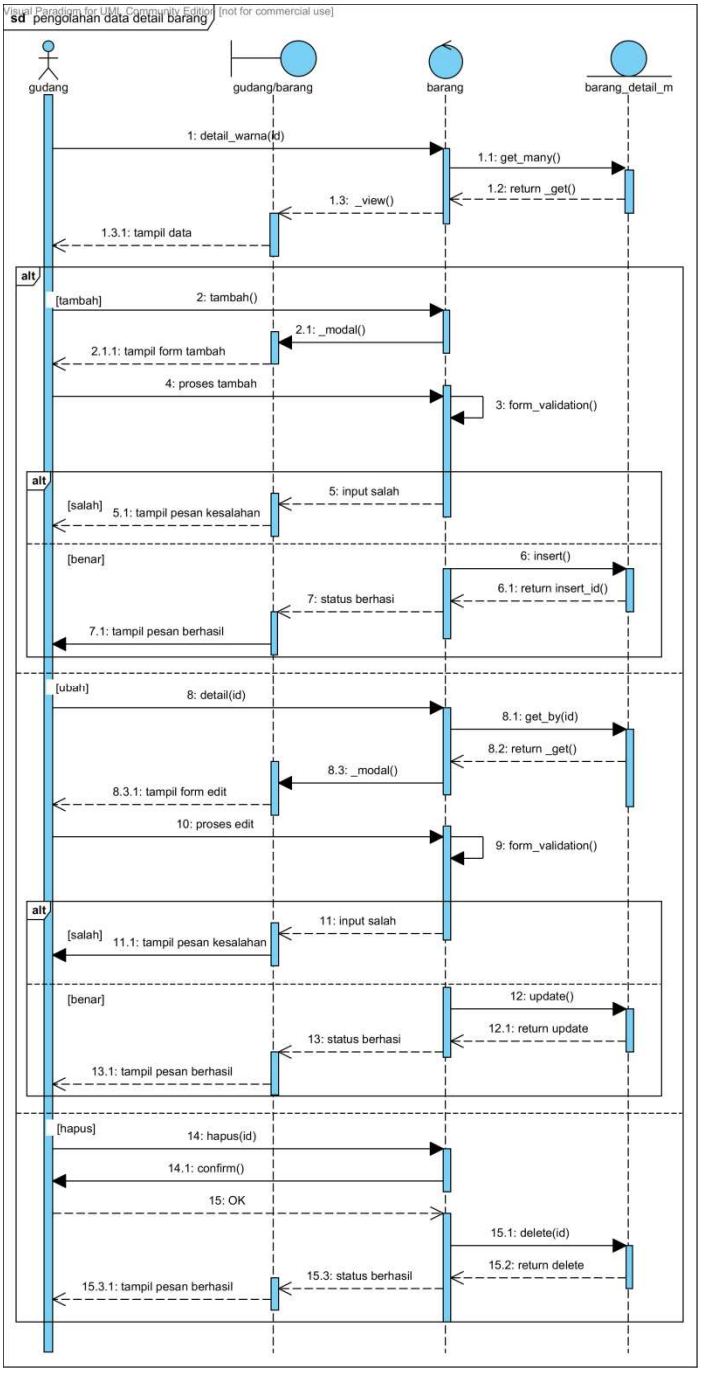
Diagram *sequence* pengolahan data warna barang dapat dilihat pada gambar 3.36.



Gambar 3. 36 Diagram *Sequence* Pengolahan Data Warna Barang

17. Diagram *Sequence* Pengolahan Data Detail Barang

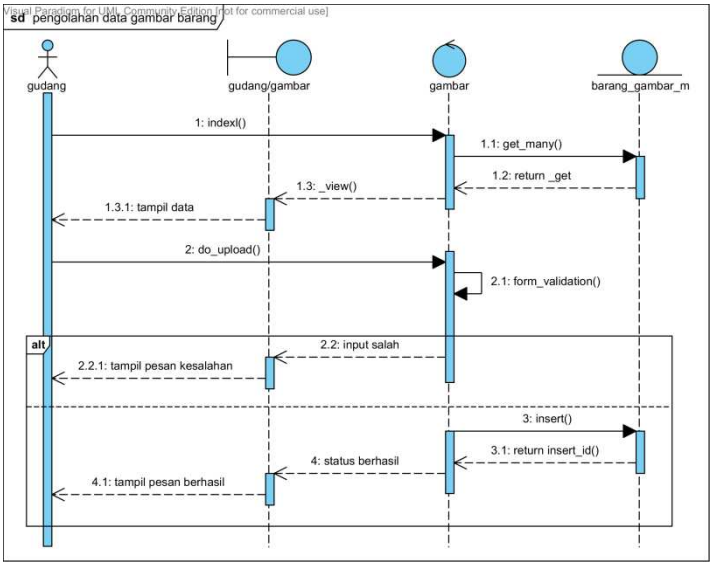
Diagram *sequence* pengolahan data detail barang dapat dilihat pada gambar 3.37.



Gambar 3. 37 Diagram *Sequence* Pengolahan Data Detail Barang

18. Diagram *Sequence* Pengolahan Data Gambar Barang

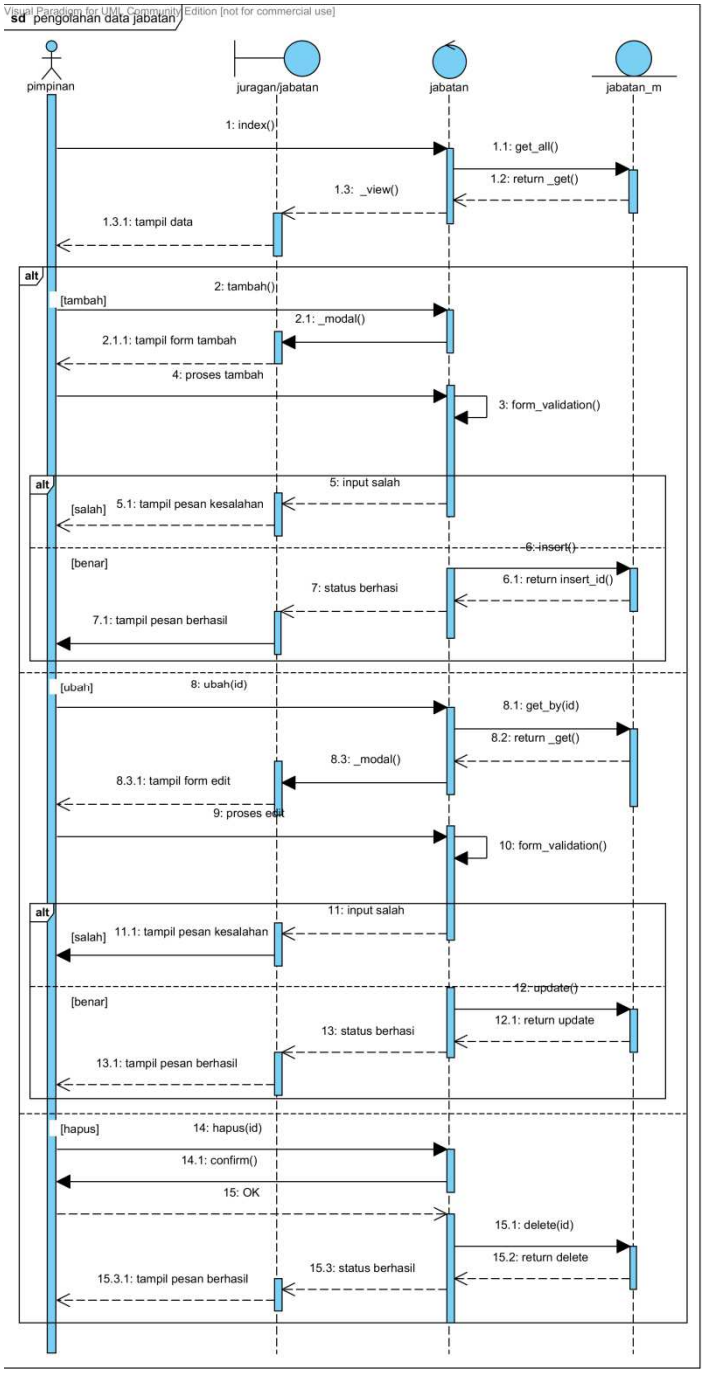
    Diagram *sequence* pengolahan data gambar barang dapat dilihat pada gambar 3.38.



Gambar 3. 38 Diagram *Sequence* Pengolahan Data Gambar Barang

19. Diagram *Sequence* Pengolahan Data Jabatan

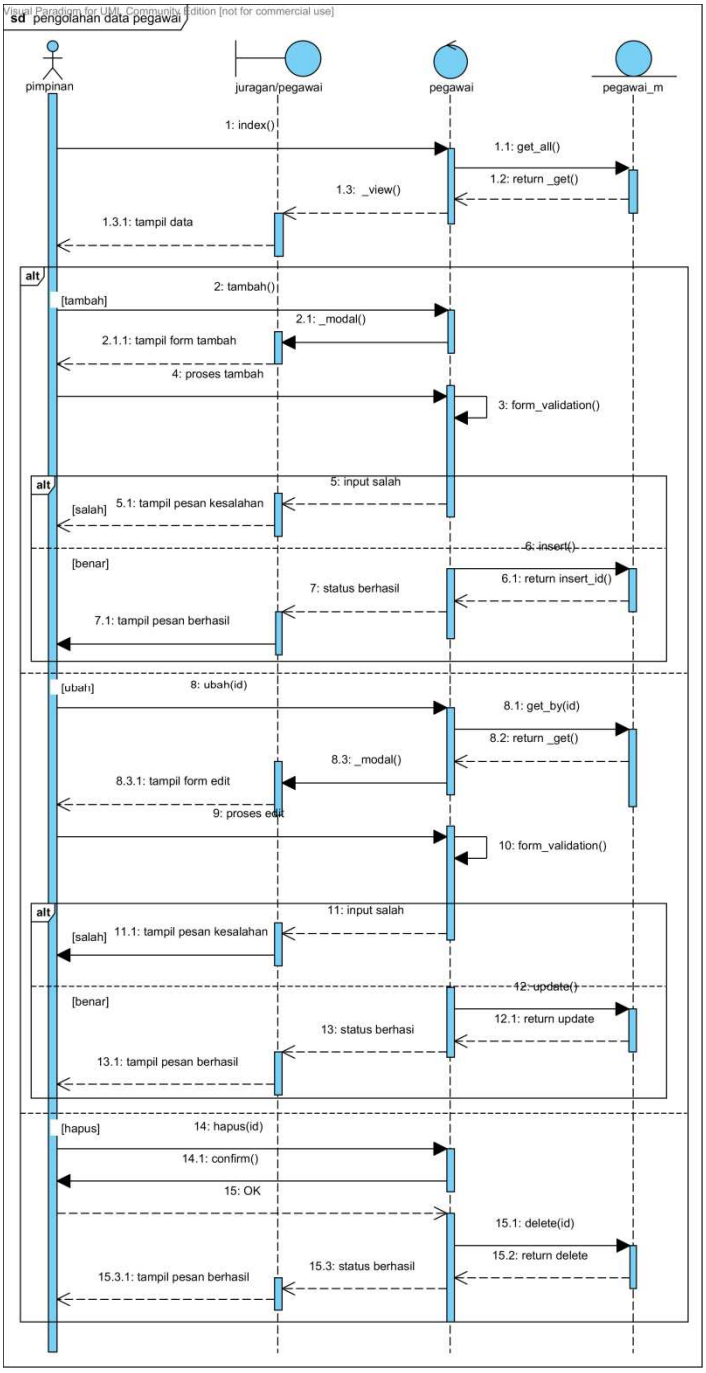
Diagram *sequence* pengolahan data jabatan dapat dilihat pada gambar 3.39.



Gambar 3. 39 Diagram *Sequence* Pengolahan Data Jabatan

20. Diagram *Sequence* Pengolahan Data Pegawai

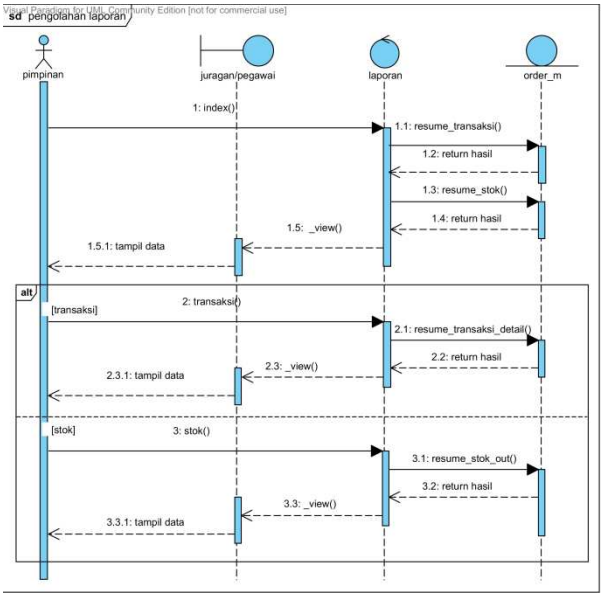
Diagram *sequence* pengolahan data pegawai dapat dilihat pada gambar 3.40.



Gambar 3. 40 Diagram *Sequence* Pengolahan Data Pegawai

21. Diagram *Sequence* Pengolahan Data Laporan

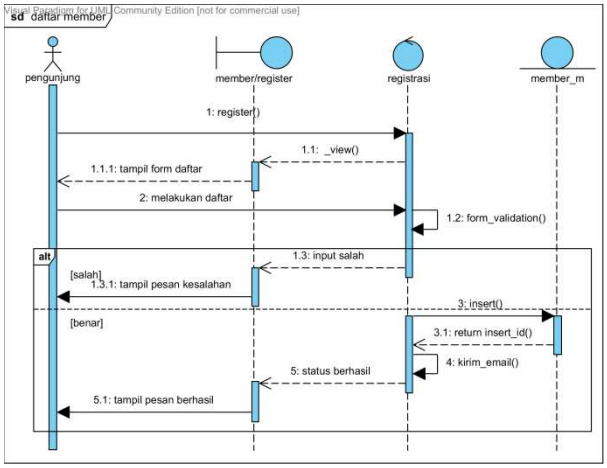
Diagram *sequence* pengolahan data laporan dapat dilihat pada gambar 3.41.



Gambar 3. 41 Diagram *Sequence* Pengolahan Data Laporan

22. Diagram *Sequence* Pengolahan Daftar Member

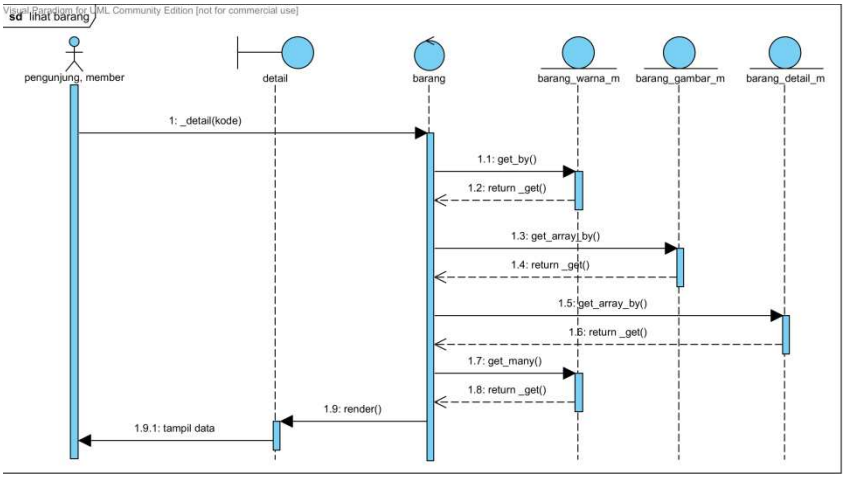
Diagram *sequence* pengolahan daftar member dapat dilihat pada gambar 3.42.



Gambar 3. 42 Diagram *Sequence* Pengolahan Daftar Member

23. Diagram *Sequence* Pengolahan Lihat Barang

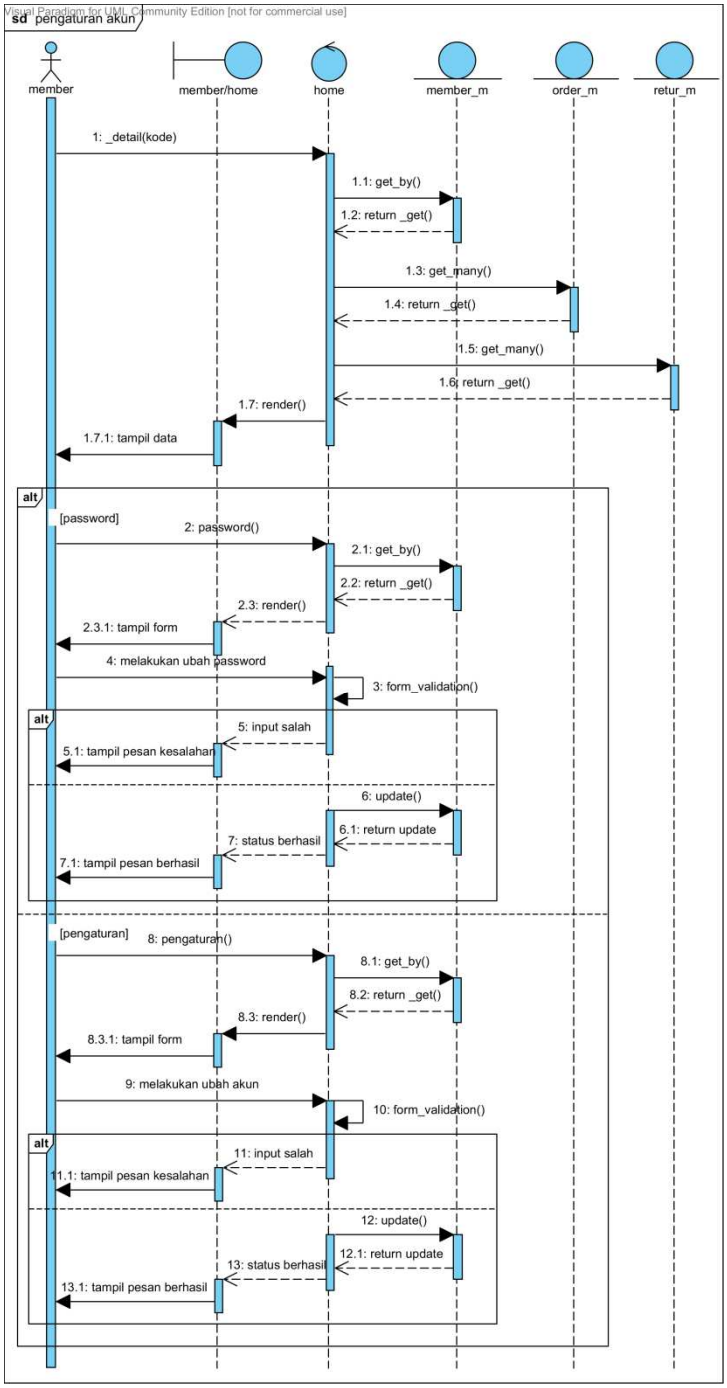
Diagram *sequence* pengolahan lihat barang dapat dilihat pada gambar 3.43.



Gambar 3. 43 Diagram *Sequence* Pengolahan Lihat Barang

24. Diagram *Sequence* Pengolahan Pengaturan Akun

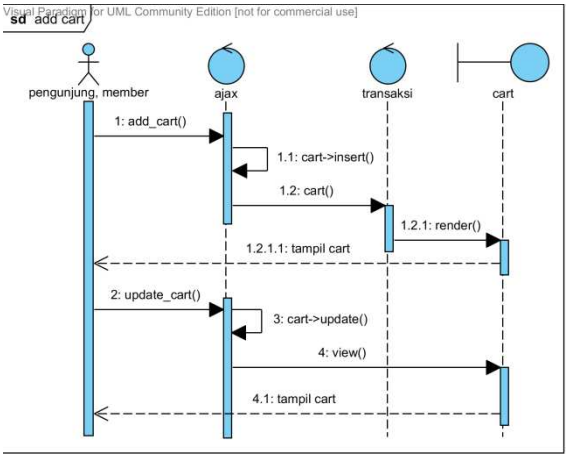
Diagram *sequence* pengolahan pengaturan akun dapat dilihat pada gambar 3.44.



Gambar 3. 44 Diagram *Sequence* Pengolahan Pengaturan Akun

25. Diagram *Sequence* Pengolahan *Add Cart*

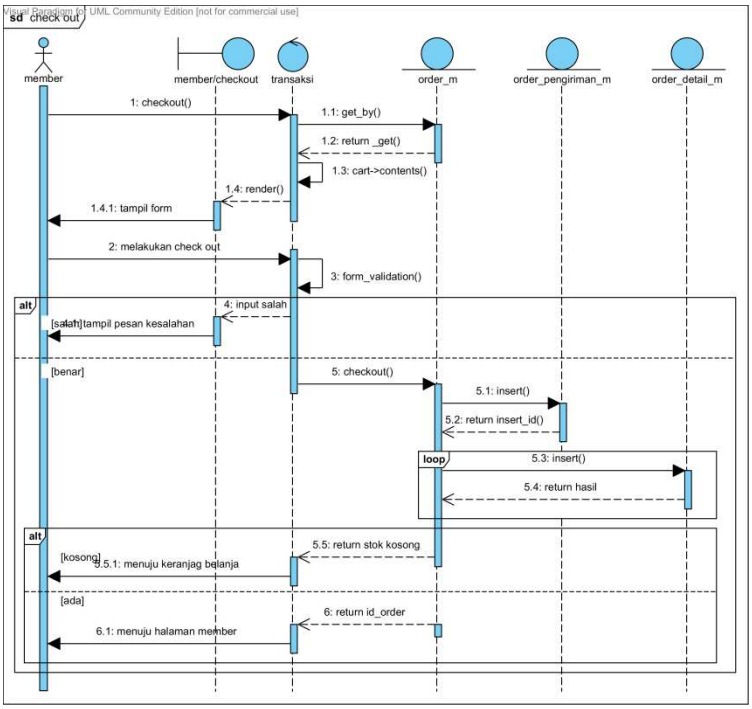
Diagram *sequence* pengolahan *add cart* dapat dilihat pada gambar 3.45.



Gambar 3. 45 Diagram *Sequence* Pengolahan *Add Cart*

26. Diagram *Sequence* Pengolahan *Checkout*

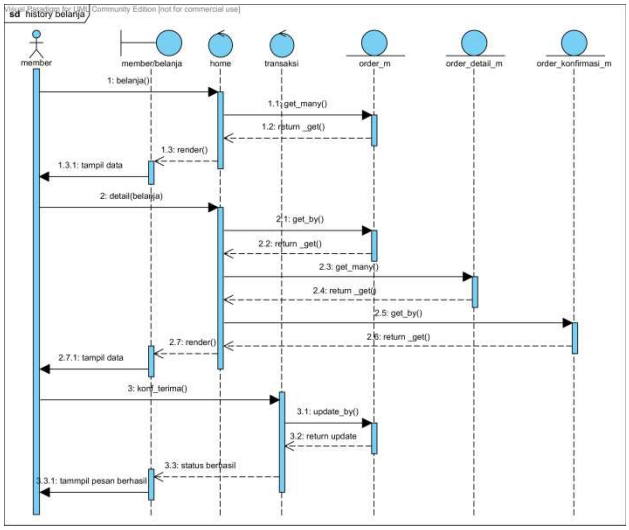
Diagram *sequence* pengolahan *checkout* dapat dilihat pada gambar 3.46.



Gambar 3. 46 Diagram *Sequence* Pengolahan *Checkout*

27. Diagram *Sequence* Pengolahan *History* Belanja

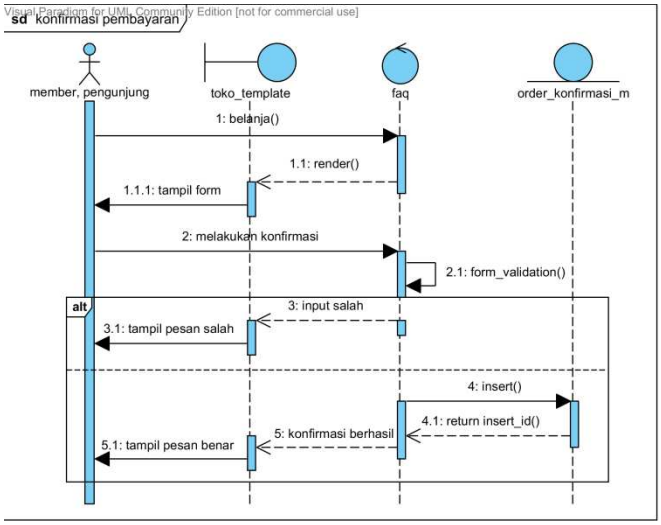
Diagram *sequence* pengolahan *history* belanja dapat dilihat pada gambar 3.47.



Gambar 3. 47 Diagram *Sequence* Pengolahan *History* Retur

28. Diagram *Sequence* Pengolahan Konfirmasi Pembayaran

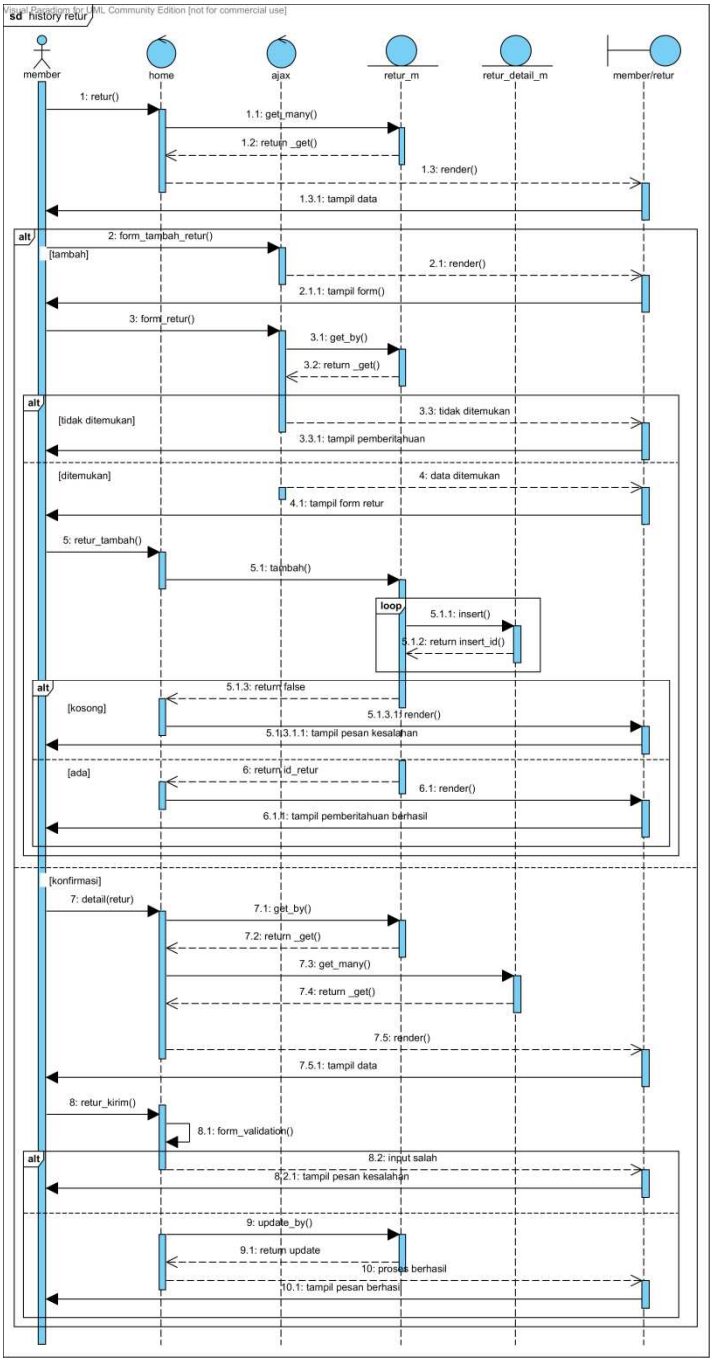
Diagram *sequence* pengolahan konfirmasi pembayaran dapat dilihat pada gambar 3.48.



Gambar 3. 48 Diagram *Sequence* Pengolahan Konfirmasi Pembayaran

29. Diagram *Sequence* Pengolahan *History* Retur

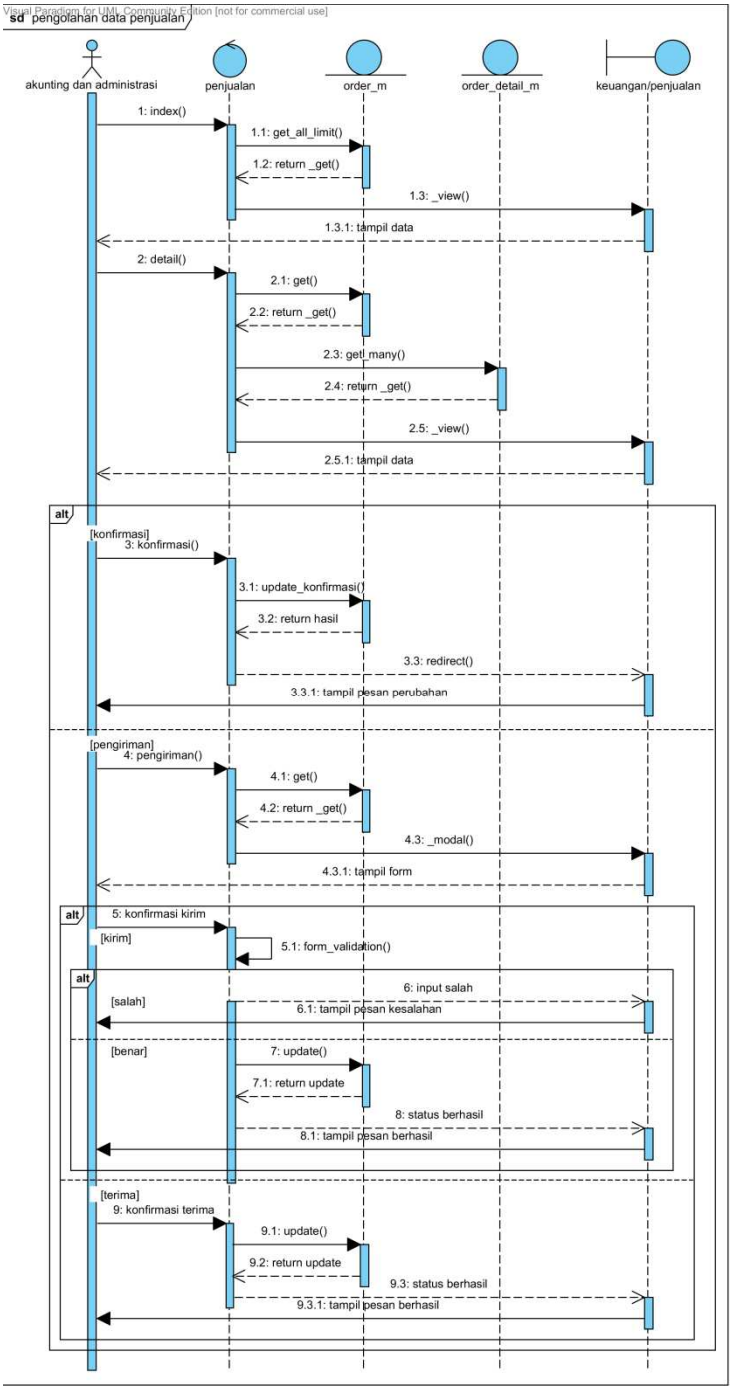
Diagram *sequence* pengolahan *history* retur dapat dilihat pada gambar 3.49.



Gambar 3. 49 Digram *Sequence* Pengolahan *History* Retur

30. Diagram *Sequence* Pengolahan Data Penjualan

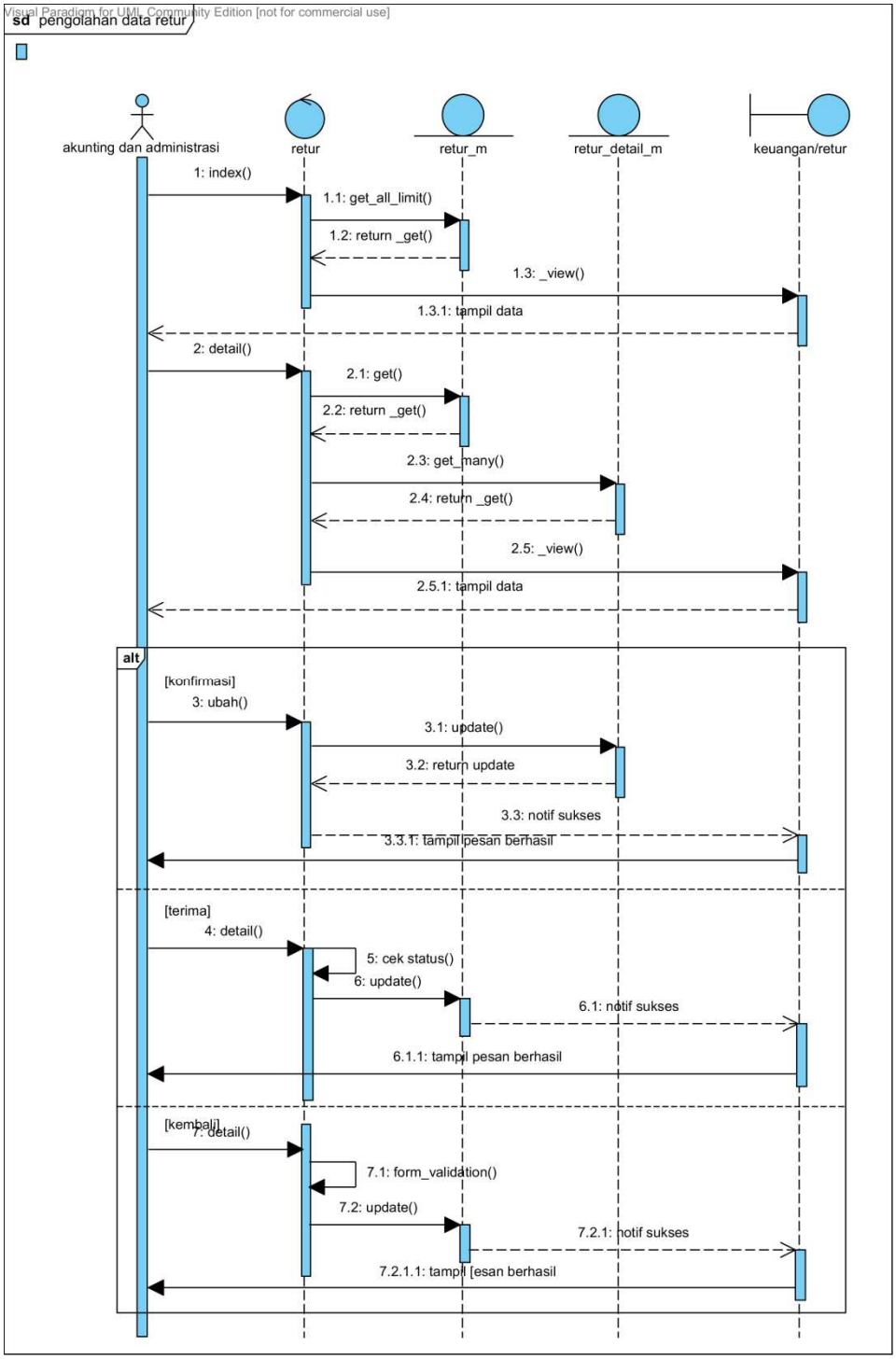
Diagram *sequence* pengolahan data penjualan dapat dilihat pada gambar 3.50.



Gambar 3. 50 Diagram *Sequence* Pengolahan Data Penjualan

31. Diagram *Sequence* Pengolahan Data Retur

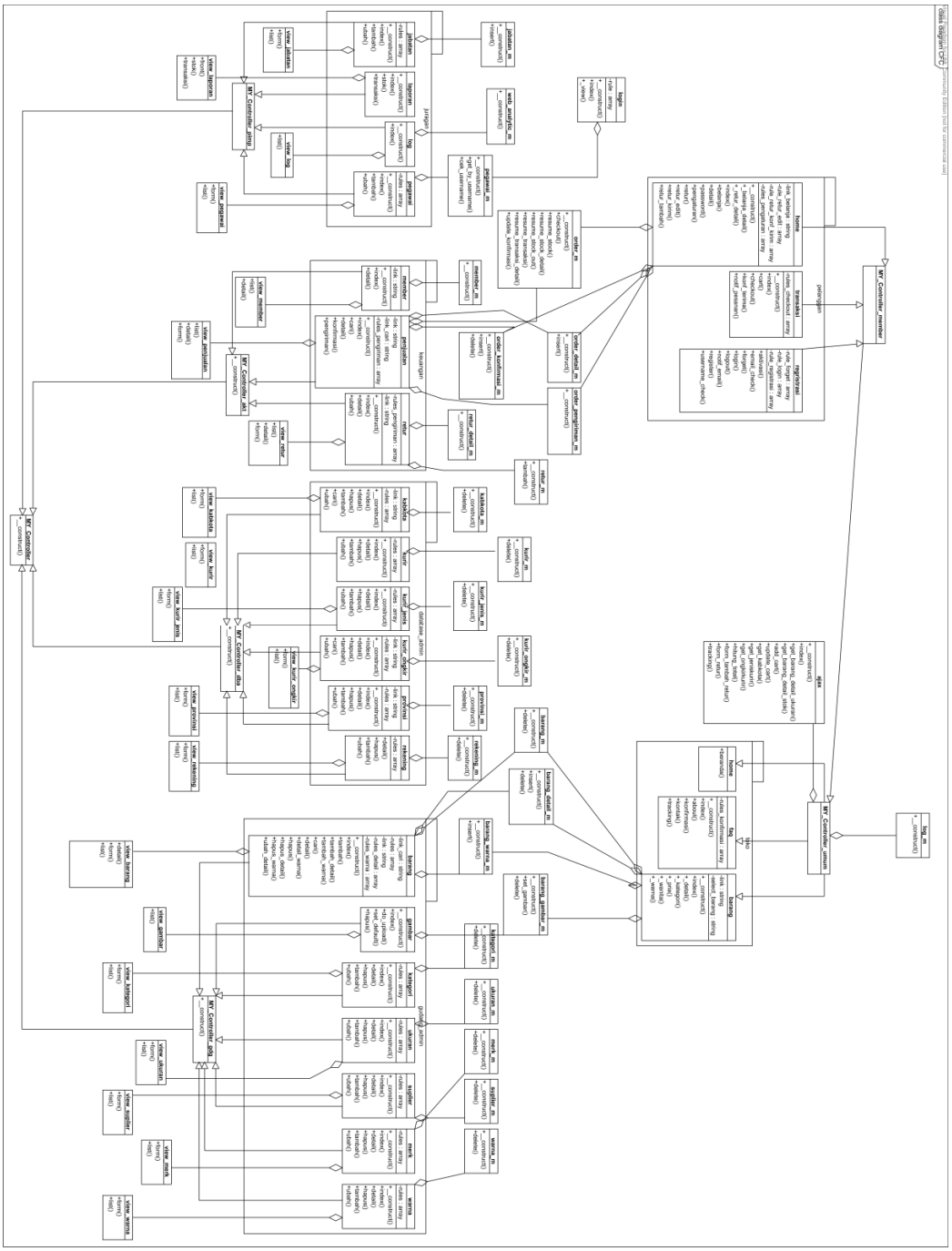
Diagram *sequence* pengolahan data retur dapat dilihat pada gambar 3.51.



Gambar 3. 51 Diagram *Sequence* Pengolahan Data Retur

3.1.5.5 Diagram *Class*

Class diagram mendeskripsikan jenis–jenis objek dalam sistem dan berbagai macam hubungan statis yang ada. *Class* diagram *e-commerce* Campus Fashion Centre dapat dilihat pada gambar 3.52.



Gambar 3. 52. *Class* Diagram *E-Commerce* Campus Fashion Centre

Setiap *class* memiliki jenis *class* masing–masing seperti terlihat pada tabel 3.43.

Tabel 3. 43. Deskripsi *Class*

No	Nama	Jenis
1	kab_kota	Control
2	kurir	Control
3	kurir_jenis	Control
4	kurir_ongkir	Control
5	provinsi	Control
6	rekening	Control
7	barang	Control
8	gambar	Control
9	kategori	Control
10	merk	Control
11	supplier	Control
12	ukuran	Control
13	warna	Control
14	jabatan	Control
15	laporan	Control
16	log	Control
17	pegawai	Control
18	dashboard	Control
19	member	Control
20	penjualan	Control
21	retur	Control
22	home	Control
23	regirstrasi	Control
24	transaksi	Control
25	barang	Control
26	faq	Control
27	home	Control
28	ajax	Control
29	login	Control
30	barang_detail_m	Entities
31	barang_gambar_m	Entities
32	barang_m	Entities
33	barang_warna_m	Entities
34	jabatan_m	Entities
35	kabkota_m	Entities
36	kategori_m	Entities
37	kurir_jenis_m	Entities
38	kurir_m	Entities

No	Nama	Jenis
39	kurir_ongkir_m	Entities
40	log_m	Entities
41	member_m	Entities
42	merk_m	Entities
43	order_detail_m	Entities
44	order_konfirmasi_m	Entities
45	order_m	Entities
46	order_pengiriman_m	Entities
47	pegawai_m	Entities
48	provinsi_m	Entities
49	rekening_m	Entities
50	retur_detail_m	Entities
51	retur_m	Entities
52	supplier_m	Entities
53	ukuran_m	Entities
54	warna_m	Entities
55	web_analytic_m	Entities
56	view_kabkota	Boundary
57	view_kurir	Boundary
58	view_kurir_jenis	Boundary
59	view_kurir_ongkir	Boundary
60	view_provinsi	Boundary
61	view_rekening	Boundary
62	view_barang	Boundary
63	view_gambar	Boundary
64	view_kategori	Boundary
65	view_merk	Boundary
66	view_suplier	Boundary
67	view_ukuran	Boundary
68	view_warna	Boundary
69	view_jabatan	Boundary
70	view_laporan	Boundary
71	view_log	Boundary
72	view_pegawai	Boundary
73	view_member	Boundary
74	view_penjualan	Boundary
75	view_retur	Boundary
76	view_belanja	Boundary
77	view_retur	Boundary
78	view_toko	Boundary

Deskripsi tiap *class* untuk data tersebut adalah sebagai berikut:

1. *Class* Kab_kota

   Class Kab_kota merupakan *class* yang mengatur tentang data kabupaten kota, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Kab_kota, yaitu:

   Deskripsi operasi *class* Kab_kota dapat dilihat pada tabel 3.44.

Tabel 3. 44. Atribut *Class* Kab_kota

No	Nama	Tipe
1	link	string
2	rules	array

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

Tabel 3. 45 Tabel Operasi *Class* Kab_kota

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat awal *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data kabkota
cari()	Prosedur yang digunakan untuk mencari data kabkota, dan menampilkan jika data ditemukan.
tambah()	Prosedur untuk melakukan proses penambahan data kabkota berdasarkan data masukan dari *form* tambah dan akan memanggil *class* kabkota_m untuk melakukan proses tambah data kabkota dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data kabkota, akan memanggil *class* kabkota_m untuk mengambil data kemudian memanggil *class* view_kabkota untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data kabkota, akan menampilkan pesan konfirmasi apakah data kabkota akan dihapus, jika ya maka akan memanggil kabkota_m untuk melakukan penghapusan data pada *database*.

2. *Class* Kurir

   Class Kurir merupakan *class* yang mengatur tentang data kurir, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Kurir, yaitu:

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

Tabel 3. 46 Tabel Atribut *Class* Kurir

No	Nama	Tipe
1	rules	array

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

Tabel 3. 47 Tabel Operasi Class Kurir

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data kurir
tambah()	Prosedur untuk melakukan proses penambahan data kurir berdasarkan data masukan dari form tambah dan akan memanggil *class* kurir_m untuk melakukan proses tambah data kurir dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data kurir, akan memanggil *class* kurir_m untuk mengambil data kemudian memanggil *class* view_kurir untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data kurir, akan menampilkan pesan konfirmasi apakah data kurir akan dihapus, jika ya maka akan memanggil kurir_m untuk melakukan penghapusan data pada *database*.

3. *Class* Kurir_jenis

   Class Kurir_jenis merupakan *class* yang mengatur tentang data jenis kurir, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Kurir_jenis, yaitu:

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

Tabel 3. 48 Tabel Atribut *Class* Kurir_jenis

No	Nama	Tipe
1	rules	array

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

Tabel 3. 49 Tabel Operasi *Class* Kurir_jenis

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat awal *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data jenis kurir
tambah()	Prosedur untuk melakukan proses penambahan data jenis kurir berdasarkan data masukan dari *form* tambah dan akan memanggil *class* kurir_jenis_m untuk melakukan proses tambah data jenis kurir dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data kurir, akan memanggil *class* kurir_jenis_m untuk mengambil data kemudian memanggil *class* view_kurir_jenis untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data jenis kurir, akan menampilkan pesan konfirmasi apakah data jenis kurir akan dihapus, jika ya maka akan memanggil kurir_jenis_m untuk melakukan penghapusan data pada *database*.

4. *Class* Kurir_ongkir

   Class Kurir_ongkir merupakan *class* yang mengatur tentang data ongkir kurir, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Kurir_ongkir, yaitu:

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

Tabel 3. 50 Tabel Atribut *Class* Kurir_ongkir

No	Nama	Tipe
1	rules	array
2	link	string

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

Tabel 3. 51 Tabel Operasi *Class* Kurir_ongkir

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data ongkir kurir

Nama	Keterangan
tambah()	Prosedur untuk melakukan proses penambahan data ongkir kurir berdasarkan data masukan dari *form* tambah dan akan memanggil *class* kurir_ongkir_m untuk melakukan proses tambah data ongkir kurir dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data ongkir kurir, akan memanggil *class* kurir_ongkir_m untuk mengambil data kemudian memanggil *class* view_kurir_ongkir untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data ongkir kurir, akan menampilkan pesan konfirmasi apakah data ongkir kurir akan dihapus, jika ya maka akan memanggil kurir_ongkir_m untuk melakukan penghapusan data pada *database*.
cari()	Prosedur yang digunakan untuk mencari data ongkir kurir, dan menampilkan jika data ditemukan.

5. *Class* Provinsi

Class Provinsi merupakan *class* yang mengatur tentang data provinsi, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Provinsi, yaitu:

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

Tabel 3. 52 Tabel Atribut *Class* Provinsi

No	Nama	Tipe
1	rules	array
2	link	string

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

Tabel 3. 53 Tabel Operasi *Class* Provinsi

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat class diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data provinsi
tambah()	Prosedur untuk melakukan proses penambahan data provinsi berdasarkan data masukan dari *form* tambah dan akan memanggil *class* provinsi_m untuk melakukan proses tambah data provinsi dan menyimpannya pada *database*.

Nama	Keterangan
ubah()	Prosedur untuk melakukan perubahan data provinsi, akan memanggil *class* provinsi_m untuk mengambil data kemudian memanggil *class* view_provinsi untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data provinsi, akan menampilkan pesan konfirmasi apakah data provinsi akan dihapus, jika ya maka akan memanggil provinsi_m untuk melakukan penghapusan data dari *database*.

6. *Class* Rekening

Class Rekening merupakan *class* yang mengatur tentang data rekening, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Rekening, yaitu:

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

Tabel 3. 54 Tabel Atribut *Class* Rekening

No	Nama	Tipe
1	rules	array

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

Tabel 3. 55 Tabel Operasi *Class* Rekening

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data rekening
tambah()	Prosedur untuk melakukan proses penambahan data rekening berdasarkan data masukan dari *form* tambah dan akan memanggil *class* rekening_m untuk melakukan proses tambah data rekening dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data rekening, akan memanggil *class* rekening_m untuk mengambil data kemudian memanggil *class* view_rekening untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data rekening, akan menampilkan pesan konfirmasi apakah data rekening akan dihapus, jika ya maka akan memanggil rekening_m untuk melakukan penghapusan data pada *database*.

7. *Class* Barang

Class Barang merupakan *class* yang mengatur tentang data barang, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Barang, yaitu:

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

Tabel 3. 56 Tabel Atribut Class Barang

No	Nama	Tipe
1	rules	array
2	rules_detail	array
3	rules_warna	array
4	link	string
5	link_cari	string

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

Tabel 3. 57 Tabel Operasi *Class* Barang

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data barang
tambah()	Prosedur untuk melakukan proses penambahan data barang berdasarkan data masukan dari *form* tambah dan akan memanggil *class* barang_m untuk melakukan proses tambah data barang dan menyimpannya pada *database*.
tambah_detail()	Prosedur untuk melakukan proses penambahan data detail barang berdasarkan data masukan dari *form* tambah detail barang dan akan memanggil *class* barang_detail_m untuk melakukan proses tambah data detail barang dan menyimpannya pada *database*.
tambah_warna()	Prosedur untuk melakukan proses penambahan data warna barang berdasarkan data masukan dari *form* tambah warna barang dan akan memanggil *class* barang_warna_m untuk melakukan proses tambah data warna barang dan menyimpannya pada *database*.
cari()	Prosedur yang digunakan untuk mencari data barang, dan menampilkan jika data ditemukan.
detail()	Prosedur yang digunakan untuk menampilkan data barang secara detail.
detail_warna()	Prosedur yang digunakan untuk menampilkan data barang warna secara detail.

Nama	Keterangan
hapus()	Prosedur untuk menghapus data barang, akan menampilkan pesan konfirmasi apakah data barang akan dihapus, jika ya maka akan memanggil barang_m untuk melakukan penghapusan data pada *database*.
hapus_detail()	Prosedur untuk menghapus data detail barang, akan menampilkan pesan konfirmasi apakah data detail barang akan dihapus, jika ya maka akan memanggil barang_detail_m untuk melakukan penghapusan data pada *database*.
hapus_warna()	Prosedur untuk menghapus data warna barang, akan menampilkan pesan konfirmasi apakah data warna barang akan dihapus, jika ya maka akan memanggil barang_warna_m untuk melakukan penghapusan data pada *database*.
ubah_detail()	Prosedur untuk melakukan perubahan data detail barang, akan memanggil class barang_detail_m untuk mengambil data kemudian memanggil *class* view_barang untuk menampilkan *form* ubah.

8. *Class* Gambar

Class Gambar merupakan *class* yang mengatur tentang data gambar, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Gambar, yaitu:

Class gambar tidak memiliki atribut.

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

Tabel 3. 58 Tabel Atribut *Class* Gambar

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data gambar
do_upload()	Prosedur untuk melakukan proses penambahan data gambar berdasarkan data masukan dari *form* tambah dan akan memanggil *class* barang_gambar_m untuk melakukan proses tambah data gambar dan menyimpannya pada *database*.
hapus()	Prosedur untuk menghapus data gambar, akan menampilkan pesan konfirmasi apakah data gambar akan dihapus, jika ya maka akan memanggil barang_gambar_m untuk melakukan penghapusan data pada *database*.

9. *Class* Kategori

Class Kategori merupakan *class* yang mengatur tentang data kategori, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Rekening, yaitu:

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

Tabel 3. 59 Tabel Atribut *Class* Kategori

No	Nama	Tipe
1	rules	array

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

Tabel 3. 60 Tabel Operasi *Class* Kategori

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data kategori
tambah()	Prosedur untuk melakukan proses penambahan data kategori berdasarkan data masukan dari *form* tambah dan akan memanggil *class* kategori_m untuk melakukan proses tambah data kategori dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data kategori, akan memanggil *class* kategori_m untuk mengambil data kemudian memanggil *class* view_kategori untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data kategori, akan menampilkan pesan konfirmasi apakah data kategori akan dihapus, jika ya maka akan memanggil kategori_m untuk melakukan penghapusan data pada *database*.

10. *Class* Merk

Class Merk merupakan *class* yang mengatur tentang data merk, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Merk, yaitu:

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

Tabel 3. 61 Tabel Atribut *Class* Merk

No	Nama	Tipe
1	rules	array

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

Tabel 3. 62 Tabel Operasi *Class* Merk

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data merk
tambah()	Prosedur untuk melakukan proses penambahan data merk berdasarkan data masukan dari *form* tambah dan akan memanggil *class* merk_m untuk melakukan proses tambah data merk dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data merk, akan memanggil *class* merk_m untuk mengambil data kemudian memanggil *class* view_merk untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data merk, akan menampilkan pesan konfirmasi apakah data merk akan dihapus, jika ya maka akan memanggil merk_m untuk melakukan penghapusan data pada *database*.

11. *Class* Suplier

Class Suplier merupakan *class* yang mengatur tentang data suplier, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Suplier, yaitu:

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

Tabel 3. 63 Tabel Atribut *Class* Suplier

No	Nama	Tipe
1	rules	array

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

Tabel 3. 64 Tabel Operasi *Class* Suplier

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data suplier
tambah()	Prosedur untuk melakukan proses penambahan data suplier berdasarkan data masukan dari *form* tambah dan akan memanggil *class* suplier_m untuk melakukan proses tambah data suplier dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data suplier, akan memanggil *class* suplier_m untuk mengambil data kemudian memanggil *class* view_suplier untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data suplier, akan menampilkan pesan konfirmasi apakah data suplier akan dihapus, jika ya maka akan memanggil suplier_m untuk melakukan penghapusan data pada *database*.

12. *Class* Ukuran

Class Ukuran merupakan *class* yang mengatur tentang data ukuran, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Ukuran, yaitu:

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

Tabel 3. 65 Tabel Atribut *Class* ukuran

No	Nama	Tipe
1	rules	array

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

Tabel 3. 66 Tabel Operasi *Class* Ukuran

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data ukuran
tambah()	Prosedur untuk melakukan proses penambahan data ukuran berdasarkan data masukan dari *form* tambah dan akan memanggil class ukuran_m untuk melakukan proses tambah data ukuran dan menyimpannya pada *database*.

Nama	Keterangan
hapus()	Prosedur untuk menghapus data ukuran, akan menampilkan pesan konfirmasi apakah data suplier akan dihapus, jika ya maka akan memanggil ukuran_m untuk melakukan penghapusan data pada *database*.

13. *Class* Warna

Class Warna merupakan *class* yang mengatur tentang data warna, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Warna, yaitu:

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

Tabel 3. 67 Tabel Atribut *Class* Warna

No	Nama	Tipe
1	rules	array

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

Tabel 3. 68 Tabel Operasi *Class* Warna

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat awal *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data warna
tambah()	Prosedur untuk melakukan proses penambahan data warna berdasarkan data masukan dari *form* tambah dan akan memanggil *class* warna_m untuk melakukan proses tambah data warna dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data warna, akan memanggil *class* warna_m untuk mengambil data kemudian memanggil class view_warna untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data warna, akan menampilkan pesan konfirmasi apakah data warna akan dihapus, jika ya maka akan memanggil warna_m untuk melakukan penghapusan data pada *database*.

14. *Class* Jabatan

Class Jabatan merupakan *class* yang mengatur tentang data jabatan, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Jabatan, yaitu:

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

Tabel 3. 69 Tabel Atribut *Class* Jabatan

No	Nama	Tipe
1	rules	array

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

Tabel 3. 70 Tabel Operasi *Class* Jabatan

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data jabatan
tambah()	Prosedur untuk melakukan proses penambahan data jabatan berdasarkan data masukan dari *form* tambah dan akan memanggil *class* jabatan_m untuk melakukan proses tambah data jabatan dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data jabatan, akan memanggil *class* jabatan_m untuk mengambil data kemudian memanggil *class* view_jabatan untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data jabatan, akan menampilkan pesan konfirmasi apakah data jabatan akan dihapus, jika ya maka akan memanggil jabatan_m untuk melakukan penghapusan data pada *database*.

15. Class Laporan

Class Laporan merupakan *class* yang mengatur tentang laporan, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Laporan, yaitu:

Class laporan tidak memiliki atribut.

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

Tabel 3. 71 Tabel Operasi *Class* Laporan

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data jabatan
stok()	Prosedur yang melakukan pengolahan laporan stok, akan mengakses *class* order_m dan menampilkan stok barang yang telah terjual
transaksi()	Prosedur yang melakukan pengolahan laporan transaksi, akan mengakses *class* order_m dan menampilkan transaksi penjualan

16. Class Log

    Class Log merupakan *class* yang mengatur tentang data *log*, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Log, yaitu:

    C*lass log* tidak memiliki attribut.

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

Tabel 3. 72 Tabel Operasi *Class* Log

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat class diakses
index()	Akan menampilkan bagan–bagan, serta nilai statistik dari data kunjungan yang terjadi pada situs.

17. *Class* Pegawai

    Class Pegawai merupakan *class* yang mengatur tentang data pegawai, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Pegawai, yaitu:

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

Tabel 3. 73 Tabel Atribut *Class* Pegawai

No	Nama	Tipe
1	rules	array

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

Tabel 3. 74 Tabel Operasi *Class* Pegawai

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data pegawai
tambah()	Prosedur untuk melakukan proses penambahan data pegawai berdasarkan data masukan dari *form* tambah dan akan memanggil *class* pegawai_m untuk melakukan proses tambah data pegawai dan menyimpannya pada *database*.
ubah()	Prosedur untuk melakukan perubahan data pegawai, akan memanggil *class* pegawai_m untuk mengambil data kemudian memanggil *class* view_pegawai untuk menampilkan *form* ubah.
hapus()	Prosedur untuk menghapus data pegawai, akan menampilkan pesan konfirmasi apakah data pegawai akan dihapus, jika ya maka akan memanggil pegawai_m untuk melakukan penghapusan data pada *database*.

18. *Class* Dashboard

Class Dashboard merupakan *class* yang mengatur tentang data pengaturan profil pegawai, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Dashboard, yaitu:

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

Tabel 3. 75 Tabel Atribut Dashboard

No	Nama	Tipe	Keterangan
1	Rules_pass	array	
2	Rules_profil	array	

Deskripsi operasi *class* Dashboard dapat dilihat pada tabel 3.76.

Tabel 3. 76 Tabel Operasi *Class* Dashboard

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data pegawai
password()	Prosedur untuk melakukan proses penambahan data pegawai berdasarkan data masukan dari form tambah dan akan memanggil *class* pegawai_m untuk melakukan proses tambah data pegawai dan menyimpannya .

Nama	Keterangan
profil()	Prosedur untuk melakukan perubahan data pegawai, akan memanggil *class* pegawai_m untuk mengambil data kemudian memanggil *class* view_pegawai untuk menampilkan *form* ubah.

19. *Class* Member

Class Member merupakan *class* yang mengatur tentang data *member*, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Member, yaitu:

Deskripsi operasi *class* Member dapat dilihat pada tabel 3.77.

Tabel 3. 77 Tabel Atribut *Class Member*

No	Nama	Tipe
1	link	string

Deskripsi operasi *class* Member dapat dilihat pada tabel 3.78.

Tabel 3. 78 Tabel Operasi *Class Member*

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data *member*
detail()	Prosedur akan mengakses member_m dan menampilkan detail dari data *member*

20. *Class* Penjualan

Class Penjualan merupakan *class* yang mengatur tentang data penjualan, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Penjualan, yaitu:

Deskripsi operasi *class* Penjualan dapat dilihat pada tabel 3.79.

Tabel 3. 79 Tabel Atribut Class Penjualan

No	Nama	Tipe
1	rules_pengiriman	array
2	link	string
3	link_cari	string

Deskripsi operasi *class* Penjualan dapat dilihat pada tabel 3.80.

Tabel 3. 80 Tabel Operasi *Class* Penjualan

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data penjualan
cari()	Prosedur yang digunakan untuk mencari data penjualan, dan menampilkan jika data ditemukan.
detail()	Prosedur akan mengakses order_m dan menampilkan detail dari data penjualan
konfirmasi()	Prosedur untuk melakukan konfirmasi pembayaran, akan mengakses *class* order_m dan melakukan perubahan status
pengiriman()	Prosedur untuk melakukan konfirmasi pengiriman order, akan melakukan akses pada *class* order_pengiriman_m

21. *Class* Retur

Class Retur merupakan *class* yang mengatur tentang data retur, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Retur, yaitu:

Deskripsi operasi *class* Retur dapat dilihat pada tabel 3.81.

Tabel 3. 81 Tabel Atribut *Class* Retur

No	Nama	Tipe
1	rules_pengiriman	array
2	link	string

Deskripsi operasi *class* Retur dapat dilihat pada tabel 3.82.

Tabel 3. 82 Tabel Operasi *Class* Retur

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan data retur
detail()	Prosedur akan mengakses retur_m dan menampilkan detail dari data retur
ubah()	Prosedur melakukan perubahan status barang retur detail, mengakses *class* retur_detail_m dan merubah nilai status pada *database*

22. *Class* Home

Class Home merupakan *class* yang mengatur tentang halaman *member*, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Home, yaitu:

Deskripsi operasi *class* Home dapat dilihat pada tabel 3.83.

Tabel 3. 83 Tabel Atribut *Class* Home

No	Nama	Tipe
1	link_belanja	string
2	rule_retur_edit	array
3	rule_retur_konf_kirim	array
4	rules_pengaturan	array

Deskripsi operasi *class* Home dapat dilihat pada tabel 3.84.

Tabel 3. 84 Tabel Operasi *Class* Home

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
_belanja_detail()	Prosedur akan mengakses order_m dan menampilkan detail dari data belanja
_retur_detail()	Prosedur akan mengakses retur_m dan menampilkan detail dari data retur
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil, dan menampilkan halaman utama *member*
belanja ()	Merupakan Prosedur yang menampilkan *history* belanja member, mengakses *class* order_m
detail()	prosedur inisialisasi dalam proses tampil detail
password()	Prosedur pengaturan perubahan *password member*
pengaturan()	Prosedur pengaturan perubahan data pribadi *member*
retur()	Merupakan Prosedur yang menampilkan *history* retur member, mengakses *class* retur_m
retur_edit()	Prosedur yang menangani perubahan data barang detail retur, akan mengakases *class* retur_detail_m dan melakukan perubahan pada *database*
retur_kirim()	Prosedur yang menangani proses perubahan data pengiriman retur, akan mengakses *class* retur_m dan melakukan perubahan pada *database*
retur_tambah()	Prosedur yang menangani tambah retur baru, akan mengakses *class* retur_m, dan melakukan penambahan data pada *database*

23. *Class* Regirstrasi

Class Registrasi merupakan *class* yang mengatur tentang registrasi *member*, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Registrasi, yaitu:

Deskripsi operasi *class* Registrasi dapat dilihat pada tabel 3.85.

Tabel 3. 85 Tabel Atribut Class Registrasi

No	Nama	Tipe
1	rule_forget	array
2	rule_login	array
3	rules_registrasi	array

Deskripsi operasi *class* Registrasi dapat dilihat pada tabel 3.86.

Tabel 3. 86 Tabel Operasi Class Registrasi

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
aktivasi()	Prosedur yang menangani proses aktivasi member, setelah melakukan pendaftaran
email_check()	Prosedur yang menangani proses validasi email, melakukan akses pada class member_m, untuk melihat apakah terdapat *email* yang sama di *database*
forget()	Prosedur yang menangani proses lupa *password*, akan mengakses member_m
login()	Prosedur yang menangani proses *login*
logout()	Prosedur yang menangani proses *logout*
notif_email()	Prosedur yang menangani pengiriman pemberitahuan *email* setelah proses daftar
register()	Prosedur yang menangani proses pendaftaran
username_check()	Prosedur yang menangani proses validasi *username*, melakukan akses pada *class* member_m, untuk melihat apakah terdapat *username* yang sama di *database*

24. *Class* Transaksi

Class Transaksi merupakan *class* yang mengatur tentang transaksi *member*, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Transaksi, yaitu:

Deskripsi operasi *class* Transaksi dapat dilihat pada tabel 3.87.

Tabel 3. 87 Tabel Atribut *Class* Transaksi

No	Nama	Tipe
1	rule_forget	array
2	rule_login	array
3	rules_registrasi	array

Deskripsi operasi *class* Transaksi dapat dilihat pada tabel 3.88.

Tabel 3. 88 Tabel Operasi *Class* Transaksi

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	akan diarahkan ke prosedur *cart*
cart()	Prosedur yang akan menampilkan keranjang belanja yang akan dibeli oleh pengunjung atau *member*
checkout()	Prosedur yang mengatur proses *checkout*, mengakses *class* order_m, dan akan menambahkan data order pada *database*
konf_terima()	Prosedur yang mengatur mengenai proses konfirmasi penerimaan barang oleh *member*

25. *Class* Barang

Class Barang merupakan *class* yang mengatur tentang halaman barang, dan bertugas memanggil *class* – *class* yang berkaitan. Atribut dan operasi yang ada pada *class* Barang yaitu:

Deskripsi operasi *class* Barang dapat dilihat pada tabel 3.89.

Tabel 3. 89 Tabel Atribut *Class* Barang

No	Nama	Tipe
1	link	string
2	select_barang	string

Deskripsi operasi *class* Barang dapat dilihat pada tabel 3.90.

Tabel 3. 90 Tabel Operasi *Class* Barang

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
_detail()	Prosedur menampilkan detail barang, akan mengakses *class* barang_m
_kategori()	Prosedure menampilkan barang dengan kategori tertentu, akan mengakses *class* barang_m

Nama	Keterangan
_warna()	Prosedur menampilkan barang dengan warna tertentu, akan mengakses *class* barang_m
_pria()	Prosedur menampilkan barang dengan jenis pria, akan mengakses *class* barang_m
_wanita()	Prosedur menampilkan barang dengan jenis wanita, akan mengakses *class* barang_m
index()	Merupakan Prosedur yang dijalankan ketika *class* pertama kali dipanggil

26. Class Faq

Class Faq merupakan *class* yang mengatur tentang halaman faq, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Faq yaitu:

Deskripsi operasi *class* Faq dapat dilihat pada tabel 3.91.

Tabel 3. 91 Tabel Atribut Class Faq

No	Nama	Tipe
1	rules_konfirmasi	array

Deskripsi operasi *class* Faq dapat dilihat pada tabel 3.92.

Tabel 3. 92 Tabel Operasi *Class* Faq

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Prosedur mengarahkan pada halaman beranda
about()	Prosedur menampilkan halaman *about*
kofirmasi()	Prosedur menangani proses konfirmasi pembayaran, akan mengakses *class* order_konfirmasi_m dan menambahkan data pada *database*
kontak()	Prosedur menampilkan halaman kontak
tracking()	Prosedur menangani proses *tracking* belanja bagi *member*, akan mengakses *class* order_m, dan menampilkan status pengiriman

27. *Class* Home

Class Home merupakan *class* yang mengatur tentang halaman beranda, dan bertugas memanggil *class–class* yang berkaitan. Atribut dan operasi yang ada pada *class* Home, yaitu:

Class home tidak memiliki attribut.

Deskripsi operasi *class* Home dapat dilihat pada tabel 3.93.

Tabel 3. 93 Tabel Operasi *Class* Home

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Prosedur menampilkan halaman beranda

28. *Class* Ajax

Class Ajax merupakan *class* yang mengatur tentang pengolahan respon ajax, dan bertugas memanggil *class – class* yang berkaitan. Atribut dan operasi yang ada pada *class* Ajax, yaitu:

Class ajax tidak memiliki attribut.

Deskripsi operasi *class* Ajax dapat dilihat pada tabel 3.94.

Tabel 3. 94 Tabel Operasi *Class* Ajax

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Prosedur menampilkan halaman beranda
add_cart()	Prosedur menangani proses tambah keranjang belanja member dan pengunjung yang berbelanja
check_email()	Prosedur yang menangani proses validasi *email*, melakukan akses pada *class* member_m, untuk melihat apakah terdapat *email* yang sama di *database*
check_username()	Prosedur yang menangani proses validasi *username*, melakukan akses pada *class* member_m, untuk melihat apakah terdapat *username* yang sama di *database*
form_retur()	Prosedur menampilkan halaman *form* retur

29. *Class* Login

Class Login merupakan *class* yang mengatur tentang *login admin*, dan bertugas memanggil *class – class* yang berkaitan. Atribut dan operasi yang ada pada *class* Login, yaitu:

Deskripsi operasi *class* Login dapat dilihat pada tabel 3.95.

Tabel 3. 95 Tabel Atribut *Class* Login

No	Nama	Tipe
1	rule	array

Deskripsi operasi *class* Login dapat dilihat pada tabel 3.96.

Tabel 3. 96 Tabel Operasi *Class* Login

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
index()	Prosedur mengarahkan pada halaman *login*
_view()	Prosedur render *view login*

30. *Class* Barang_detail_m

Class Barang_detail_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data detail barang. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Barang_detail_m, yaitu:

Class barang_detail_m tidak memiliki atribut.

Deskripsi operasi *class* Barang_detail_m dapat dilihat pada tabel 3.97.

Tabel 3. 97 Tabel Operasi *Class* barang_detail_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
insert()	Prosedur menambahlan data pada *database*
delete()	Prosedur menangani hapus data di *database*

31. *Class* Barang_gambar_m

Class Barang_gambar_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data gambar barang. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Barang_gambar_m, yaitu:

Class barang_gambar_m tidak memiliki atribut.

Deskripsi operasi *class* Barang_gambar_m dapat dilihat pada tabel 3.98.

Tabel 3. 98 Tabel Operasi *Class* barang_gambar_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
set_gambar()	Prosedur menangani perubahan status data gambar
delete()	Prosedur menangani hapus data di *database*

32. *Class* Barang_m

Class Barang_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data barang. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Barang_m, yaitu:

Class barang_detail_m tidak memiliki atribut.

Deskripsi operasi *class* Barang_m dapat dilihat pada tabel 3.99.

Tabel 3. 99 Tabel Operasi *Class* barang_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di *database*

33. *Class* Barang_warna_m

Class Barang_warna_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data warna barang. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Barang_warna_m, yaitu:

Class barang_warna_m tidak memiliki atribut.

Deskripsi operasi *class* Kurir dapat dilihat pada tabel 3.100.

Tabel 3. 100 Tabel Operasi *Class* barang_warna_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
insert()	Prosedur menambahlan data pada *database*
delete()	Prosedur menangani hapus data di *database*

34. *Class* Jabatan_m

Class Jabatan_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data jabatan. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Jabatan_m, yaitu:

Class Jabatan_m tidak memiliki attribut.

Deskripsi operasi *class* Jabatan_m dapat dilihat pada tabel 3.101.

Tabel 3. 101 Tabel Operasi *Class* Jabatan_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
insert()	Prosedur menambahlan data pada *database*
delete()	Prosedur menangani hapus data di *database*

35. *Class* Kabkota_m

Class Kabkota_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data kabupaten kota. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Kabkota_m, yaitu:

Class Kabkota_m tidak memiliki attribut.

Deskripsi operasi *class* Kabkota_m dapat dilihat pada tabel 3.102.

Tabel 3. 102 Tabel Operasi *Class* kabkota_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di *database*

36. *Class* Kategori_m

Class Kategori_m 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 ada pada *class* Kategori_m, yaitu:

Class Kategori_m tidak memiliki attribut.

Deskripsi operasi *class* Kategori_m dapat dilihat pada tabel 3.103.

Tabel 3. 103 Tabel Operasi *Class* kategori_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di *database*

37. *Class* Kurir_jenis_m

Class Kurir_jenis_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data jenis kurir. *Class* ini berhubungan langsung dengan

database untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Kurir_jenis_m, yaitu:

Class Kurir_jenis_m tidak memiliki attribut.

Deskripsi operasi *class* Kurir_jenis_m dapat dilihat pada tabel 3.104.

Tabel 3. 104 Tabel Operasi *Class* kurir_jenis_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di *database*

38. *Class* Kurir_m

Class Kurir_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data kurir. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Kurir_m, yaitu:

Class Kurir_m tidak memiliki attribut.

Deskripsi operasi *class* Kurir_m dapat dilihat pada tabel 3.105.

Tabel 3. 105 Tabel Operasi *Class* Kurir_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di *database*

39. *Class* Kurir_ongkir_m

Class Kurir_ongkir_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data ongkir kurir. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Kurir_ongkir_m, yaitu:

Class Kurir_ongkir_m tidak memiliki attribut.

Deskripsi operasi *class* Kurir_ongkir_m dapat dilihat pada tabel 3.106.

Tabel 3. 106 Tabel Operasi *Class* kurir_ongkir_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di database

40. *Class* Log_m

Class Log_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data log. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Log_m, yaitu:

Class Log_m tidak memiliki attribut.

Deskripsi operasi *class* Log_m dapat dilihat pada tabel 3.107.

Tabel 3. 107 Tabel Operasi *Class* Log_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat awal class diakses

41. *Class* Member_m

Class Member_m 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 ada pada *class* Member_m, yaitu:

Class Member_m tidak memiliki attribut .

Deskripsi operasi *class* Member_m dapat dilihat pada tabel 3.108.

Tabel 3. 108 Tabel Operasi *Class* Member_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
konfirmasi()	Prosedur menangani proses konfirmasi dan melakukan perubahan status ke *database*

42. *Class* Merk_m

Class Merk_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data merk. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Merk_m yaitu:

Class Merk_m tidak memiliki attribut.

Deskripsi operasi *class* Merk_m dapat dilihat pada tabel 3.109.

Tabel 3. 109 Tabel Operasi *Class* Merk_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di database

43. *Class* Order_detail_m

Class Order_detail_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data detail order. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Order_detail_m yaitu:

Class Order_detail_m tidak memiliki attribut.

Deskripsi operasi *class* Order_detail_m dapat dilihat pada tabel 3.110.

Tabel 3. 110 Tabel Operasi *Class* Order_detail_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat awal class diakses
insert()	Prosedur menambahlan data pada database

44. *Class* Order_konfirmasi_m

Class Order_konfirmasi_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data konfirmasi order. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Order_konfirmasi_m, yaitu:

Class Order_konfirmasi_m tidak memiliki attribut.

Deskripsi operasi *class* Order_konfirmasi_m dapat dilihat pada tabel 3.111.

Tabel 3. 111 Tabel Operasi *Class* order_konfirmasi_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat awal class diakses
insert()	Prosedur menambahkan data pada database
delete()	Prosedur menangani hapus data di database

45. *Class* Order_m

Class Order_m 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 ada pada *class* Order_m, yaitu:

Class Order_m tidak memiliki attribut.

Deskripsi operasi *class* Order_m dapat dilihat pada tabel 3.112.

Tabel 3. 112 Tabel Operasi *Class* Order_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
checkout()	Prosedur menambahlan data pada *database*
resume_stock()	Proses mengambil data stok dari *database*
resume_stok_detail()	Proses mengambil data stok detail dari *database*
resume_stok_out()	Proses mengambil data stok keluar dari *database*
resume_transaksi()	Proses mengambil data transaksi dari *database*
update_konfirmasi()	Proses konfirmasi pembayaran merubah status order pada *database*

46. *Class* Order_pengiriman_m

Class Order_pengiriman_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data pengiriman order. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Order_pengiriman_m yaitu:

Class Order_pengiriman_m tidak memiliki attribut.

Deskripsi operasi *class* Order_pengiriman_m dapat dilihat pada tabel 3.113.

Tabel 3. 113 Tabel Operasi Class Order_pengiriman_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses

47. *Class* Pegawai_m

Class Pegawai_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data pegawai. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Pegawai_m, yaitu:

Class Pegawai_m tidak memiliki attribut.

Deskripsi operasi *class* Pegawai_m dapat dilihat pada tabel 3.114.

Tabel 3. 114 Tabel Operasi *Class* pegawai_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
cek_username()	prosedur menangani pengecekan username pada *database*
get_by_username()	procedure mengambil data berdasarkan *username*

48. *Class* Provinsi_m

Class Provinsi_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data provinsi. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Provinsi_m, yaitu:

Class Provinsi_m tidak memiliki attribut.

Deskripsi operasi *class* Provinsi_m dapat dilihat pada tabel 3.115.

Tabel 3. 115 Tabel Operasi *Class* Provinsi_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat awal class diakses
delete()	Prosedur menangani hapus data di database

49. *Class* Rekening_m

Class Rekening_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data rekening. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Rekening_m, yaitu:

Class Rekening_m tidak memiliki attribut.

Deskripsi operasi *class* Rekening_m dapat dilihat pada tabel 3.116.

Tabel 3. 116 Tabel Operasi *Class* rekening_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di *database*

50. *Class* Retur_detail_m

Class Retur_detail_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data detail retur. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Retur_detail_m, yaitu:

Class Retur_detail_m tidak memiliki attribut.

Deskripsi operasi *class* Retur_detail_m dapat dilihat pada tabel 3.117.

Tabel 3. 117 Tabel Operasi *Class* retur_detail_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses

51. *Class* Retur_m

Class Retur_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data retur. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Retur_m, yaitu:

Class Retur_m tidak memiliki attribut.

Deskripsi operasi *class* Retur_m dapat dilihat pada tabel 3.118.

Tabel 3. 118 Tabel Operasi *Class* retur_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
tambah()	Prosedur menambahlan data pada *database*

52. *Class* Suplier_m

Class Supplier_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data suplier. *Class* ini berhubungan langsung dengan *database*

untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Supplier_m, yaitu:

Class Suplier_m tidak memiliki attribut.

Deskripsi operasi *class* Suplier_m dapat dilihat pada tabel 3.119.

Tabel 3. 119 Tabel Operasi *Class* suplier_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di *database*

53. *Class* Ukuran_m

Class Ukuran_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data ukuran. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data. Atribut dan operasi yang ada pada *class* Ukuran_m, yaitu:

Class Ukuran_m tidak memiliki attribut.

Deskripsi operasi *class* Ukuran_m dapat dilihat pada tabel 3.120.

Tabel 3. 120 Tabel Operasi *Class* Ukuran_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di *database*

54. *Class* Warna_m

Class Warna_m 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 ada pada *class* Warna_m yaitu:

Class Warna_m tidak memiliki attribut.

Deskripsi operasi *class* Warna_m dapat dilihat pada tabel 3.121.

Tabel 3. 121 Tabel Operasi *Class* warna_m

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses
delete()	Prosedur menangani hapus data di database

55. *Class* Web_analytic_m

Class Web_analytic_m merupakan *class* yang berisi prosedur-prosedur untuk mengolah data log. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil data. Atribut dan operasi yang ada pada *class* Web_analytic_m, yaitu:

Class Web_analytic_m tidak memiliki attribut.

Deskripsi operasi *class* Web_analytic_m dapat dilihat pada tabel 3.122.

Tabel 3. 122 Tabel Operasi *Class* web_analytic

Nama	Keterangan
__construct()	prosedur yang otomatis diakses pada saat *class* diakses

56. *Class* View_kabkota

Class View_kabkota merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class* View_kabkota, yaitu:

Class View_kabkota tidak memiliki attribut.

Deskripsi operasi *class* View_kabkota dapat dilihat pada tabel 3.123.

Tabel 3. 123 Tabel Operasi *Class* view_kabkota

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

57. *Class* View_kurir

Class View_kurir merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class* View_kurir, yaitu:

Class View_kurir tidak memiliki attribut.

Deskripsi operasi *class* View_kurir dapat dilihat pada tabel 3.124.

Tabel 3. 124 Tabel Operasi *Class* view_kurir

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

58. *Class* View_kurir_jenis

Class View_kurir_jenis merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class,* yaitu:

Class View_kurir_jenis tidak memiliki attribut.

Deskripsi operasi *class* View_kurir_jenis dapat dilihat pada tabel 3.125.

Tabel 3. 125 Tabel Operasi *Class* view_kurir_jenis

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

59. *Class* View_kurir_ongkir

Class View_kurir_ongkir merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_kurir_ongkir tidak memiliki attribut.

Deskripsi operasi *class* View_kurir_ongkir dapat dilihat pada tabel 3.126.

Tabel 3. 126 Tabel Operasi *Class* view_kurir_ongkir

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

60. *Class* View_provinsi

Class View_provinsi merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_provinsi tidak memiliki attribut.

Deskripsi operasi *class* View_provinsi dapat dilihat pada tabel 3.127.

Tabel 3. 127 Tabel Operasi *Class* view_provinsi

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

61. *Class* View_rekening

Class View_rekening merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_rekening tidak memiliki attribut.

Deskripsi operasi *class* View_rekening dapat dilihat pada tabel 3.128.

Tabel 3. 128 Tabel Operasi Class view_rekening

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

62. *Class* View_barang

Class View_barang merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_barang tidak memiliki attribut.

Deskripsi operasi *class* View_barang dapat dilihat pada tabel 3.129.

Tabel 3. 129 Tabel Operasi *Class* view_barang

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah
detail()	Prosedur menampilkan detail data yang didapatkan dari *control* dan *model*

63. *Class* View_gambar

Class View_barang merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_gambar tidak memiliki attribut.

Deskripsi operasi *class* View_gambar dapat dilihat pada tabel 3.130.

Tabel 3. 130 Tabel Operasi Class view_gambar

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.

64. *Class* View_kategori

Class View_kategori merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang ada pada *class*, yaitu:

a. Deskripsi Atribut

Class View_kategori tidak memiliki attribut.

b. Deskripsi Operasi

Deskripsi operasi *class* View_kategori dapat dilihat pada tabel 3.131.

Tabel 3. 131 Tabel Operasi Class view_kategori

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

65. *Class* View_merk

Class View_merk merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_merk tidak memiliki attribut.

Deskripsi operasi *class* View_merk dapat dilihat pada tabel 3.132.

Tabel 3. 132 Tabel Operasi *Class* view_merk

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

66. *Class* View_suplier

Class View_suplier merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_suplier tidak memiliki attribut.

Deskripsi operasi *class* View_suplier dapat dilihat pada tabel 3.133.

Tabel 3. 133 Tabel Operasi *Class* view_suplier

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

67. *Class* View_ukuran

Class View_ukuran merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_ukuran tidak memiliki attribut.

Deskripsi operasi *class* View_ukuran dapat dilihat pada tabel 3.134.

Tabel 3. 134 Tabel Operasi *Class* view_ukuran

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

68. *Class* View_warna

Class View_warna merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_warna tidak memiliki attribut.

Deskripsi operasi *class* View_warna dapat dilihat pada tabel 3.135.

Tabel 3. 135 Tabel Operasi *Class* view_warna

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

69. *Class* View_jabatan

Class View_jabatan merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_jabatan tidak memiliki attribut.

Deskripsi operasi *class* View_jabatan dapat dilihat pada tabel 3.136.

Tabel 3. 136 Tabel Operasi *Class* view_jabatan

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

70. *Class* View_laporan

Class View_laporan merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_laporan tidak memiliki attribut.

Deskripsi operasi *class* View_laporan dapat dilihat pada tabel 3.137.

Tabel 3. 137 Tabel Operasi *Class* view_laporan

Nama	Keterangan
front()	Prosedur untuk menampilkan data *resume*.
stock()	Prosedur untuk menampilkan data stok
transaksi()	Prosedur untuk menampilkan data transaksi

71. *Class* View_log

Class View_log merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_log tidak memiliki attribut.

Deskripsi operasi *class* View_log dapat dilihat pada tabel 3.138.

Tabel 3. 138 Tabel Operasi *Class* view_log

Nama	Keterangan
list()	Prosedur untuk menampilkan data statistik *log*.

72. *Class* View_pegawai

Class View_pegawai merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_pegawai tidak memiliki attribut.

Deskripsi operasi *class* View_pegawai dapat dilihat pada tabel 3.139.

Tabel 3. 139 Tabel Operasi *Class* view_pegawai

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

73. *Class* View_member

Class View_member merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_member tidak memiliki attribut.

Deskripsi operasi *class* View_member dapat dilihat pada tabel 3.140.

Tabel 3. 140 Tabel Operasi *Class* view_member

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
detail()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

74. *Class* View_penjualan

Class View_penjualan merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_penjualan tidak memiliki attribut.

Deskripsi operasi *class* View_penjualan dapat dilihat pada tabel 3.141.

Tabel 3. 141 Tabel Operasi Class view_penjualan

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah
detail()	Prosedur menampilkan detail data yang didapatkan dari control dan model

75. *Class* View_retur

Class View_retur merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_retur tidak memiliki attribut.

Deskripsi operasi *class* View_retur dapat dilihat pada tabel 3.142.

Tabel 3. 142 Tabel Operasi *Class* view_retur

Nama	Keterangan
list()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
form()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah
detail()	Prosedur menampilkan detail data yang didapatkan dari control dan model

76. *Class* View_belanja

Class View_belanja merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_belanja tidak memiliki attribut.

Deskripsi operasi *class* View_belanja dapat dilihat pada tabel 3.143.

Tabel 3. 143 Tabel Operasi *Class* view_belanja

Nama	Keterangan
home()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
detail()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

77. *Class* View_retur

Class View_retur merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_retur tidak memiliki attribut.

Deskripsi operasi *class* View_retur dapat dilihat pada tabel 3.144.

Tabel 3. 144 Tabel Operasi *Class* view_retur

Nama	Keterangan
home()	Prosedur untuk menampilkan data hasil prosedur dari *class model* dan *control*.
detail()	Prosedur untuk menampilkan *form* ketika memanggil prosedur tambah data dan ubah

78. *Class* View_toko

Class View_toko merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*. Atribut dan operasi yang ada pada *class*, yaitu:

Class View_toko tidak memiliki attribut.

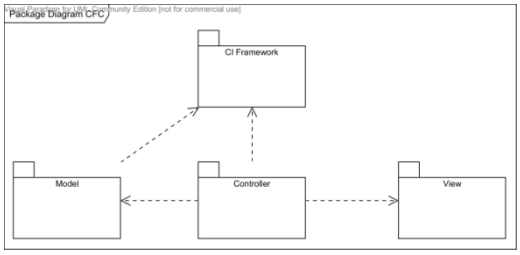
Deskripsi operasi *class* View_toko dapat dilihat pada tabel 3.145.

Tabel 3. 145 Tabel Operasi *Class* view_toko

Nama	Keterangan
about()	Menampilkan halaman *about*
konfirmasi()	Menampilkan halaman konfirmasi pembayaran
kontak()	Menampilkan halaman kontak
tracking()	Menampilkan halaman *tracking*
cart()	Menampilkan halaman keranjang belanja
detail()	Menampilkan halaman detail barang yang didapat dari *control* dan *model*
login()	Menampilkan halaman *login*
produk()	Menampilkan halaman data barang yang didapat dari *control* dan *model*

3.1.5.6 Diagram *Package*

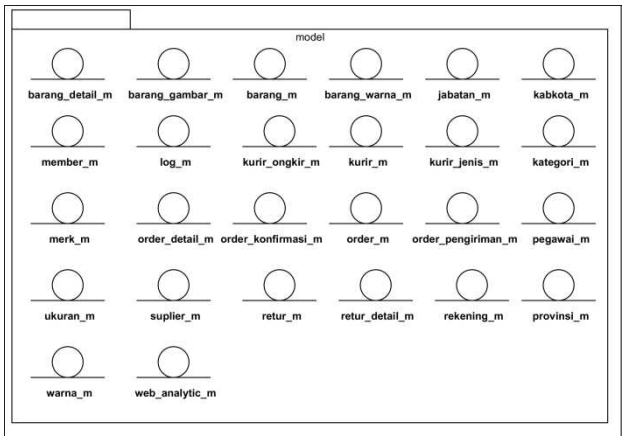
Package diagram merupakan pengelompokan kesamaan jenis *class* untuk memudahkan dalam pengorganisasian file. Berikut adalah *package* diagram untuk *E-Commerce* Campus Fashion Centre.



Gambar 3. 53. Diagram *Package*

1. *Model Package*

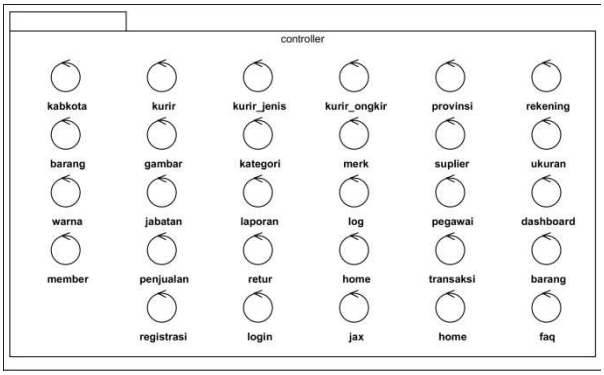
   Package model atau entities berisi *class-class* model yang mengakses langsung pada *database* untuk dapat mengolah data. Isi dari *package* model dapat dilihat pada gambar 3.54.



Gambar 3. 54. ***Model Package***

2. *Controller Package*

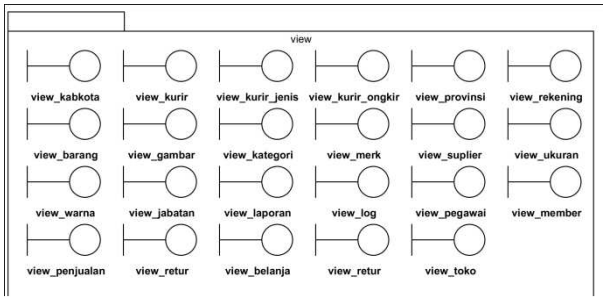
   Package controllers atau control berisi *class-class* control yang merupakan pengatur dalam setiap proses yang ada. Isi dari *package controllers* dapat dilihat pada gambar 3.55.



Gambar 3. 55. ***Controller Package***

3. *View Package*

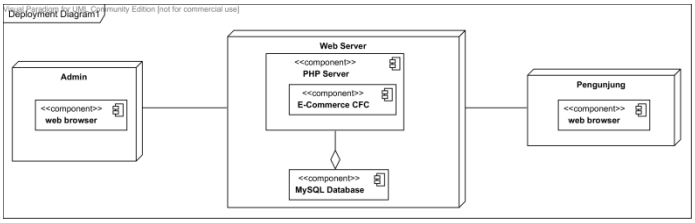
   Package view atau boundary berisi *class-class* view yang digunakan untuk menampilkan *interface* yang dapat berupa tampilan tabel atau form. Isi dari *package* view dapat dilihat pada gambar 3.56.



Gambar 3. 56. *View Package*

3.1.5.7 Diagram *Deployment*

Deployment diagram menunjukkan susunan fisik sebuah aplikasi, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras. deployment diagram *E-Commerce* Campus Fashion Centre dapat dilihat pada gambar 3.57.



Gambar 3. 57. Diagram *Deployment*

3.2 Perancangan Sistem

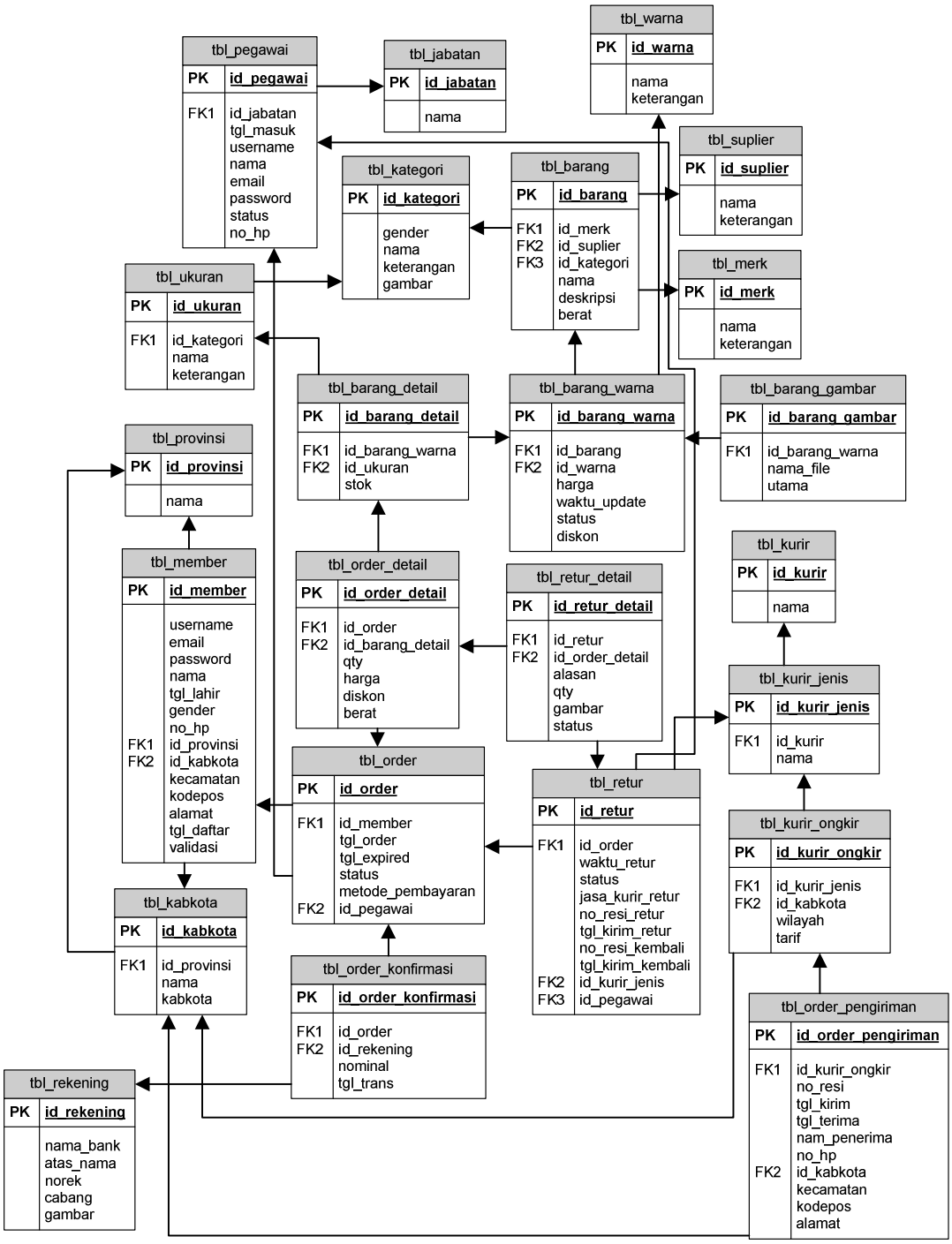
Perancangan merupakan tahapan setelah dilakukan analisis, Pada bagian ini akan dijelaskan perancangan aplikasi yang akan dikembangkan untuk menggambarkan perbedaan antara sistem yang sedang berjalan dengan sistem yang akan dikembangkan.

3.2.1 Perancangan Data

Perancangan data merupakan tahapan membangun keterkaitan antar data sehingga tercipta basis data. Untuk menggambarkan digunakan *tools* skema relasi dan struktur tabel.

3.2.1.1 Skema Relasi

Berikut skema relasi data Campus Fashion Centre, pada gambar 3.58.



Gambar 3. 58. Skema Relasi

3.2.1.2 Struktur Tabel

Tabel – tabel yang terdapat dalam basis data yang digunakan dalam pembangunan aplikasi *e-commerce* di Campus Fashion Centre ini adalah sebagai berikut.

1. Table tbl_member

   Tbl_member berfungsi untuk menyimpan data *member*. Berikut struktur tbl_member.

Tabel 3. 146 Struktur Tbl_member

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_member	Int	10	PK
username	Varchar	12	
email	Varchar	50	
password	Varchar	256	
nama	Varchar	60	
tgl_lahir	Date		
gender	Enum		(‘L’, ‘P’)
no_hp	Varchar	12	
id_provinsi	Int	10	FK *reference* tabel tbl_provinsi (id_ provinsi)
id_kabkota	Int	10	FK *reference* tabel tbl_kabkota (id_ kabkota)
kecamatan	Varchar	30	
kodepos	char	5	
alamat	Varchar	250	
tgl_daftar	dateTime		
validasi	enum		(‘Y’, ‘N’)

2. Tabel tbl_order

   Tbl_order digunakan untuk menyimpan data-data pesanan. Berikut struktur tbl_order.

Tabel 3. 147 Struktur Tbl_order

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_order	Integer	10	PK
id_member	Integer	10	FK *reference* tabel tbl_member (id_member)

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_pegawai	Integer	10	FK *reference* tabel tbl_pegawai (id_pegawai)
status	Enum		(‘new’, ‘konf’, ‘bayar’, ‘kirim’, ‘terima’, ‘batal’)
tgl_order	datetime		
expired_date	datetime		
metode_pembayaran	Enum		(‘trans’, ‘paypal’)

3. Tabel Tbl_order_detail

   Tbl_order_detail digunakan untuk menyimpan data-data detail pemesanan. Berikut struktur tabel tbl_order_detail.

Tabel 3. 148 Struktur tbl_order_detail

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_order _detail	Integer	10	PK
id_barang_detail	Integer	10	FK *reference* tabel tbl_barang_detail (id_barang_detail)
id_order	Integer	10	FK *reference* tabel tbl_order (id_order)
qty	Integer	10	
harga	Integer	10	
berat	Integer	10	
diskon	Integer	10	

4. Tabel Tbl_kabkota

   Tbl_kabkota digunakan untuk menyimpan data kota. Berikut struktur tabel tbl_kabkota.

Tabel 3. 149 Struktur tbl_kabkota

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_kabkota	Integer	10	PK
id_provinsi	Integer	10	FK *reference* tabel tbl_provinsi (id_provinsi)
nama	Varchar	50	
kabkota	Enum		(‘KABUPATEN’, ‘KOTA’)

5. Tabel Tbl_provinsi

   Tbl_provinsi digunakan untuk menyimpan data provinsi. Berikut struktur tabel tbl_provinsi.

Tabel 3. 150 Struktur tbl_provinsi

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_provinsi	Integer	10	PK
nama	Varchar	100	

6. Tabel Tbl_kurir

   Tbl_kurir digunakan untuk menyimpan data jasa pengiriman. Berikut struktur tabel tbl_kurir.

Tabel 3. 151 Struktur Tbl_Kurir

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_kurir	Integer	10	PK
nama	Varchar	30	

7. Tabel Tbl_kurir_jenis

   Tbl_kurir_jenis digunakan untuk menyimpan data kategori pengiriman. Berikut struktur tabel tbl_kurir_jenis.

Tabel 3. 152 Struktur Tbl_Kurir_Jenis

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_kurir_jenis	Integer	10	PK
id_kurir	Integer	10	FK *reference* tabel tbl_kurir (id_kurir)
nama	Varchar	30	

8. Tabel Tbl_kategori

   Tabel tbl_kategori digunakan untuk menyimpan data kategori barang. Berikut struktur tabel tbl_kategori.

Tabel 3. 153 Struktur Tbl_kategori

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_kategori	Integer	10	PK

Nama *Field*	Tipe Data	Ukuran	Keterangan
gender	Enum		(‘L’,’P’)
nama	Varchar	50	
keterangan	Varchar	256	
gambar	Varchar	256	

9. Tabel Tbl_warna

   Tabel tbl_warna digunakan untuk menyimpan data warna barang. Berikut struktur tabel tbl_warna.

Tabel 3. 154 Struktur Tbl_Warna

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_warna	Integer	10	PK
nama	Varchar	30	
keterangan	Varchar	256	

10. Tabel Tbl_ukuran

    Tabel tbl_ukuran digunakan untuk menyimpan data ukuran barang. Berikut struktur tabel tbl_ukuran.

Tabel 3. 155 Struktur Tbl_Ukuran

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_ukuran	Integer	10	PK
Id_kategori	Integer	10	FK *reference* tabel tbl_kategori (id_kategori)
nama	Varchar	30	
keterangan	Varchar	256	

11. Tabel Tbl_barang_warna

    Tabel tbl_barang_warna digunakan untuk menyimpan data warna tertentu. Berikut struktur tabel tbl_barang_warna.

Tabel 3. 156 Struktur Tbl_barang_warna

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_barang_warna	Integer	10	PK

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_barang	Integer	10	FK *reference* tabel tbl_barang (id_barang)
id_warna	Integer	10	FK *reference* tabel tbl_warna (id_ warna)
harga	decimal	(10,0)	
waktu_update	datetime		
status	enum		(‘Y’, ‘N’)
diskon	Integer	10	

12. Tabel Tbl_barang_gambar

    Tabel tbl_barang_gambar berfungsi untuk menyimpan data gambar. Berikut struktur tabel tbl_barang_gambar.

Tabel 3. 157 Struktur Tbl_barang_gambar

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_barang_ gambar	Integer	10	PK
id_barang_warna	Integer	10	FK *references* tbl_barang_warna (id_barang_warna)
nama_file	Varchar	60	
utama	Enum		(‘1’,’0’)

13. Tabel Tbl_barang

    Tabel tbl_barang digunakan untuk menyimpan data barang. Berikut struktur tabel tbl_barang.

Tabel 3. 158 Struktur Tbl_Barang

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_barang	Integer	10	PK
id_merk	Integer	10	FK *reference* tabel tbl_merk (id_merk)
Id_suplier	Integer	10	FK *reference* tabel tbl_suplier (id_suplier)
id_kategori	Integer	10	FK *reference* tabel tbl_kategori (id_kategori)
nama	Varchar	50	
deskripsi	Text		
berat	Integer	10	

14. Tabel tbl_barang_detail

Tabel tbl_barang_detail berfungsi untuk menyimpan data barang detail. Berikut struktur tabel tbl_barang_detail.

Tabel 3. 159 Struktur Tbl_Barang_Detail

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_barang_detail	Integer	10	PK
id_barang_warna	Integer	10	FK *references* tbl_barang_warna (id_barang_warna)
Id_ukuran	Integer	10	FK *reference* tabel tbl_ukuran (id_ukuran)
stok	Integer	10	

15. Tabel Tbl_merk

Tabel tbl_merk digunakan untuk menyimpan data merk barang. Berikut struktur tabel tbl_merk.

Tabel 3. 160 Struktur Tbl_Merk

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_merk	Integer	10	PK
Nama	Varchar	30	
Keterangan	Varchar	256	

16. Tabel Tbl_suplier

Tabel tbl_suplier digunakan untuk menyimpan data suplier. Berikut struktur tabel tbl_suplier.

Tabel 3. 161 Struktur Tbl_suplier

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_ suplier	Integer	10	PK
nama	Varchar	30	
keterangan	Varchar	256	

17. Tabel Tbl_pegawai

Tabel tbl_pegawai digunakan untuk menyimpan data pegawai. Berikut struktur tabel tbl_pegawai.

Tabel 3. 162 Struktur Tbl_Pegawai

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_pegawai	Integer	11	PK
Id_jabatan	Char	3	FK *reference* tabel tbl_jabatan (id_jabatan)
tgl_masuk	date		
username	Varchar	30	
password	Varchar	256	
email	Varchar	60	
nama	Varchar	60	
status	char	1	
no_hp	Varchar	15	

18. Tabel Tbl_jabatan

    Tabel tbl_jabatan digunakan untuk menyimpan data jabatan Berikut struktur tabel tbl_jabatan.

Tabel 3. 163 Struktur Tbl_jabatan

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_jabatan	char	3	PK
nama	Varchar	20	

19. Tabel Tbl_rekening

    Tabel tbl_rekening digunakan untuk menyimpan data bank. Berikut struktur tabel tbl_rekening.

Tabel 3. 164Struktur Tbl_rekening

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_rekening	Integer	11	PK
nama_ bank	Varchar	20	
atas_nama	Varchar	60	
norek	Varchar	15	
cabang	Varchar	120	
gambar	Varchar	60	

20. Tabel Tbl_order_konfirmasi

Tabel tbl_order_konfirmasi digunakan untuk menyimpan data konfirmasi pesanan. Berikut struktur tabel tbl_order_konfirmasi.

Tabel 3. 165 Struktur Tbl_order_konfirmasi

Nama *Field*	Tipe Data	Ukuran	Keterangan
id_ order_konfirmasi	Integer	10	PK
id_order	Integer	10	FK *reference* tabel tbl_order (id_ order)
id_rekening	Integer	10	FK *reference* tabel tbl_rekening (id_rekening)
nominal	Integer	10	
Tgl_Trans	Date		

21. Tabel Tbl_kurir_ongkir

Tabel tbl_kurir_ongkir digunakan untuk menyimpan data ongkos kirim. Berikut struktur tabel tbl_kurir_ongkir.

Tabel 3. 166 Struktur Tbl_kurir_ongkir

Nama Field	Tipe Data	Ukuran	Keterangan
id_kurir_ongkir	Integer	10	PK
id_ kurir_jenis	Integer	10	FK *reference* tabel tbl_kurir_jenis (id_kurir_jenis)
id_kabkota	Integer	10	FK *reference* tabel tbl_kabkota (id_kabkota)
wilayah	Varchar	30	
tarif	Integer	10	

22. Tabel Tbl_order_pengiriman

Tabel tbl_order_pengiriman digunakan untuk menyimpan data pengiriman. Berikut struktur tabel tbl_order_pengiriman.

Tabel 3. 167 Struktur Tbl_order_pengiriman

Nama Field	Tipe Data	Ukuran	Keterangan
id_order_pengiriman	Integer	10	PK
Id_kurir_ongkir	Integer	10	FK *reference* tabel tbl_kurir_ongkir (id_kurir_ongkir)

Nama Field	Tipe Data	Ukuran	Keterangan
no_resi	Varchar	20	
tgl_kirim	Date		
tgl_terima	Date		
nama_penerima	Varchar	60	
no_hp	Varchar	12	
id_kabkota	Integer	10	FK *reference* tabel tbl_kabkota (id_kabkota)
kecamatan	Varchar	30	
kodepos	Char	5	
alamat	Varchar	250	

23. Tabel Tbl_retur

Tabel tbl_retur digunakan untuk menyimpan data retur. Berikut struktur tabel tbl_retur.

Tabel 3. 168 Struktur Tbl_retur

Nama Field	Tipe Data	Ukuran	Keterangan
id_retur	Integer	10	PK
id_order	Integer	10	FK *reference* tabel tbl_order (id_order)
id_pegawai	Integer	10	FK *reference* tabel tbl_pegawai (id_pegawai)
waktu_retur	Datetime		
status	Enum		('new','konf','kirim', 'terima', 'kembali', 'batal')
jasa_kurir_retur	Varchar		
no_resi_retur	Varchar	40	
tanggal_kirim_retur	Date		
Id_kurir_jenis	Integer	30	FK *reference* tabel tbl_kurir_jenis (id_kurir_jenis)
no_resi_kembali	Varchar	30	
tgl_kirim_kembali	Date		

24. Tabel Tbl_retur_detail

Tabel tbl_retur_detail digunakan untuk menyimpan data retur detail. Berikut struktur tabel tbl_retur_detail.

Tabel 3. 169 Struktur Tbl_retur_detail

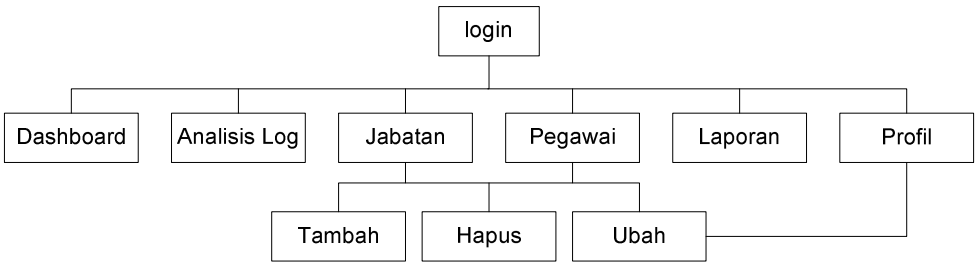
Nama Field	Tipe Data	Ukuran	Keterangan
id_retur_detail	Integer	10	PK
id_retur	Integer	10	FK *reference* tabel retur (id_retur)
id_order_detail	Integer	10	FK *reference* tabel detail order (id_detail_order)
alasan	Enum		(‘rusak’,’salah’)
qty	Integer		
gambar	varchar	20	
status	Enum		(‘New’,'Diterima', 'Ditolak')

3.2.2 Perancangan Arsitektur Menu

Struktur menu dirancang sesuai dengan hak akses pengguna aplikasi. Pengguna aplikasi ini yaitu pimpinan, *database* administrator, gudang,akunting, member, dan pengunjung. Adapun struktur menunya adalah sebagai berikut.

1. Struktur Menu Pimpinan

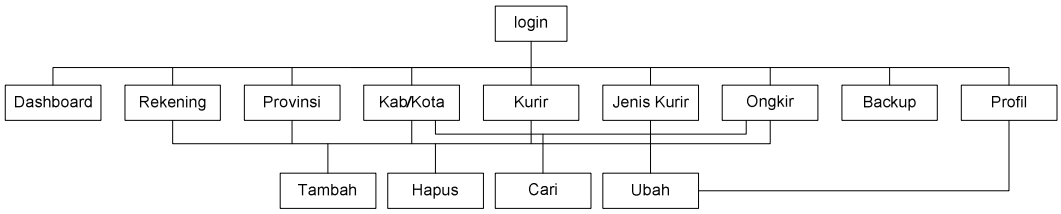
   Perancangan struktur menu untuk Pimpinan dapat dilihat pada gambar 3.59.



Gambar 3. 59 Struktur Menu Pimpinan

2. Struktur Menu Database Admin

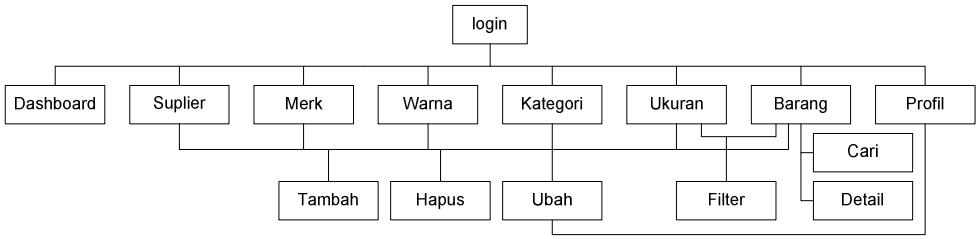
   Perancangan struktur menu untuk Database Admin dapat dilihat pada gambar 3.60.



Gambar 3. 60 Struktur Menu *Database* Admin

3. Struktur Menu Gudang

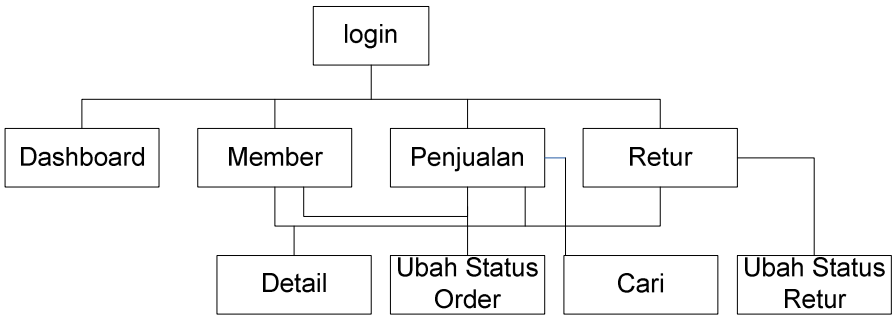
   Perancangan struktur menu untuk Gudang dapat dilihat pada gambar 3.61.



Gambar 3. 61 Struktur Menu Gudang

4. Struktur Menu Akunting

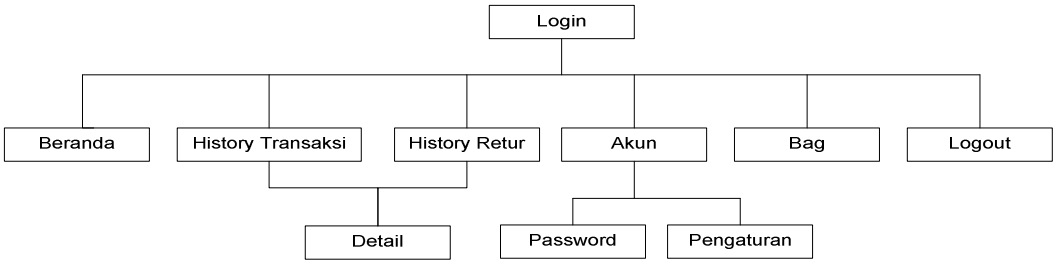
   Perancangan struktur menu untuk Akunting dapat dilihat pada gambar 3.62.



Gambar 3. 62 Struktur Menu Akunting

5. Struktur Menu Member

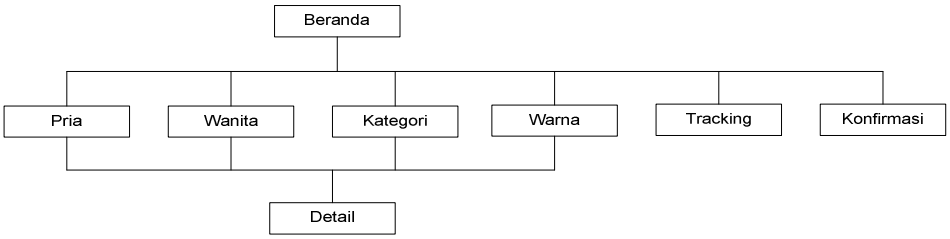
   Perancangan struktur menu untuk member dapat dilihat pada gambar 3.63.



Gambar 3. 63 Struktur Menu Member

6. Struktur Menu Pengunjung

Perancangan struktur menu untuk dba dapat dilihat pada gambar 3.64.



Gambar 3. 64 Struktur Menu Pengunjung

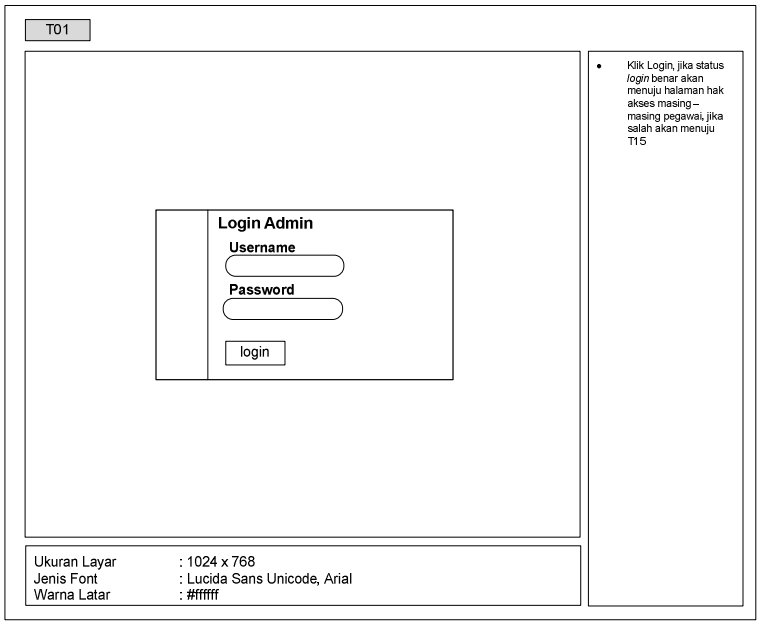
3.2.3 Perancangan Antarmuka

Interface atau antar muka merupakan tampilan dari suau aplikasi yang berperan sebagai media komunikasi antara aplikasi dan pengguna.

3.2.3.1 Perancangan Tampilan

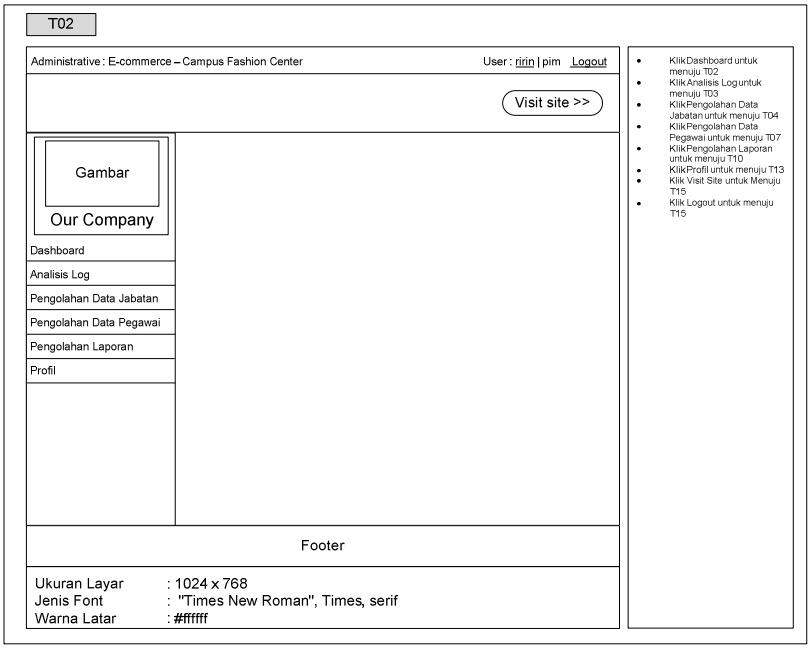
Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh pengguna, berikut perancangan *interface* bagi tiap pengguna aplikasi:

1. Perancangan Antarmuka Pimpinan
   a. Perancangan Antarmuka *Login* Pegawai



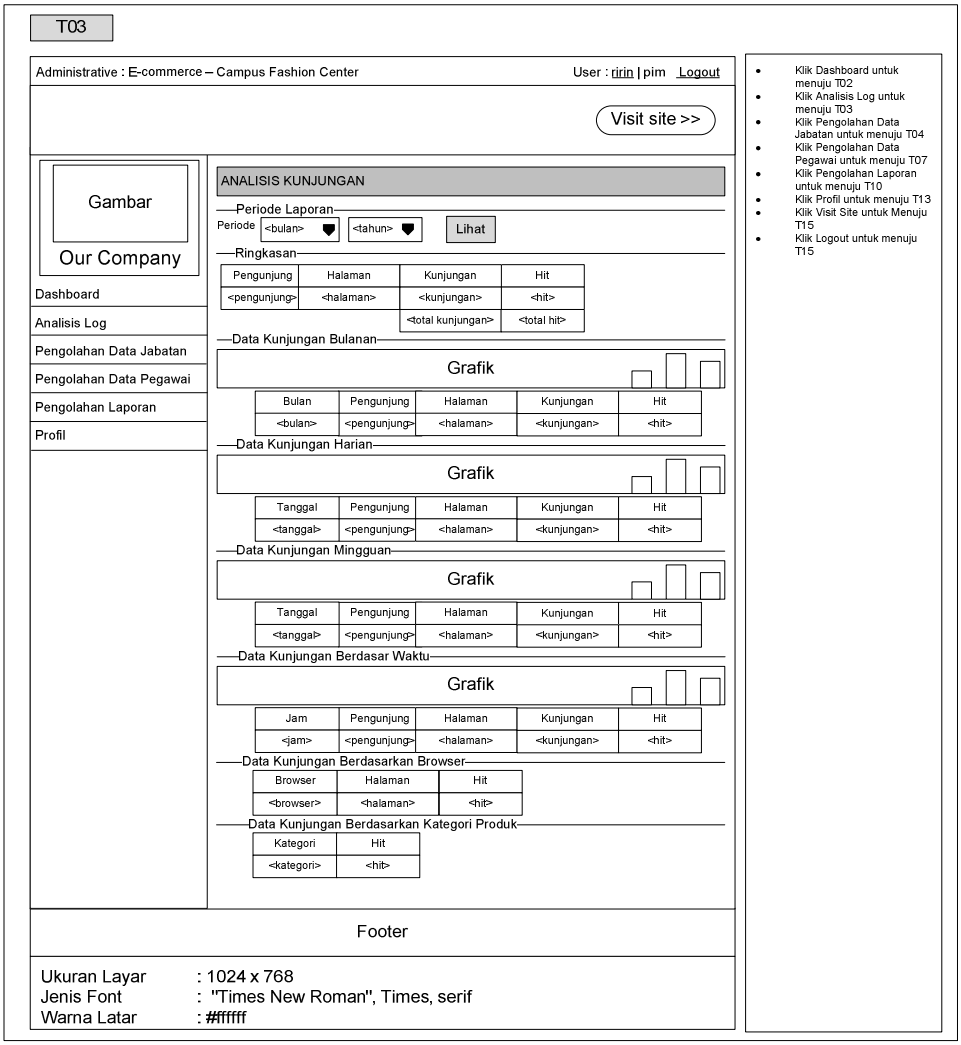
Gambar 3. 65. Perancangan Tampilan *Login* Administrator

   b. Perancangan Antarmuka Halaman Utama Pimpinan



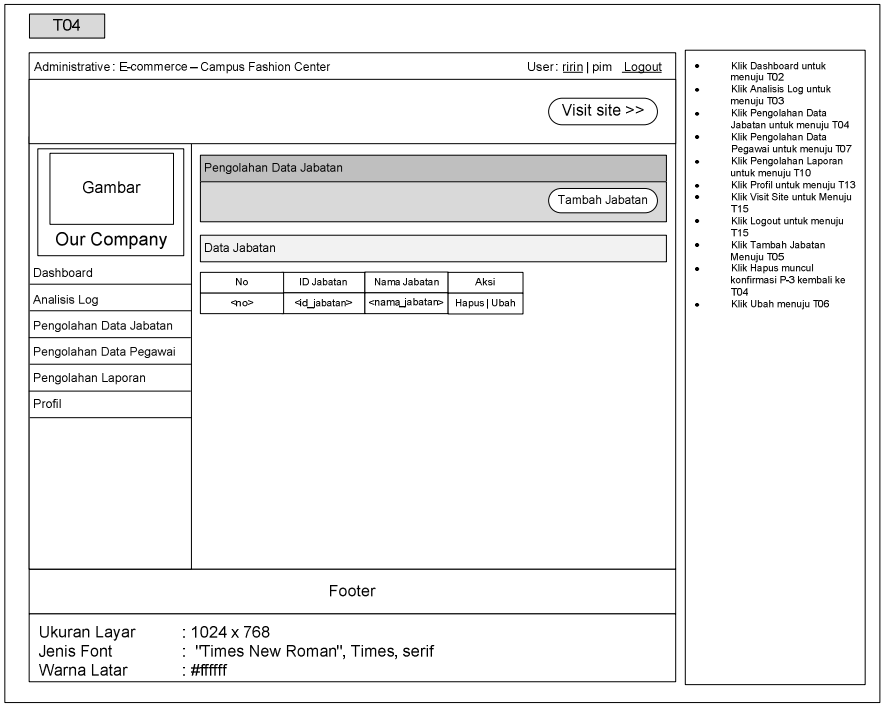
Gambar 3. 66. Perancangan Tampilan Halaman Utama Pimpinan

c. Perancangan Antarmuka Analisis *Log*



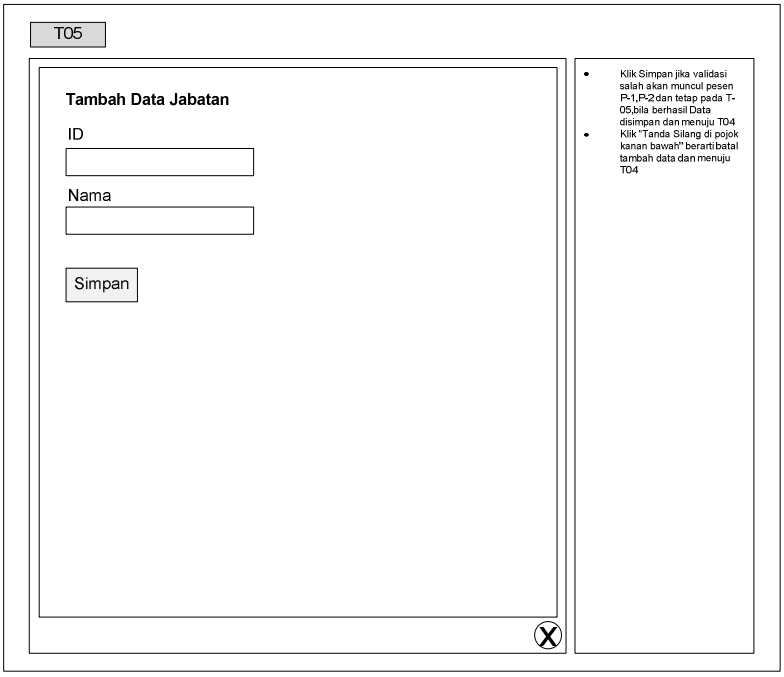
Gambar 3. 67. Perancangan Tampilan Analisis *Log*

d. Perancangan Antarmuka Pengolahan Data Jabatan



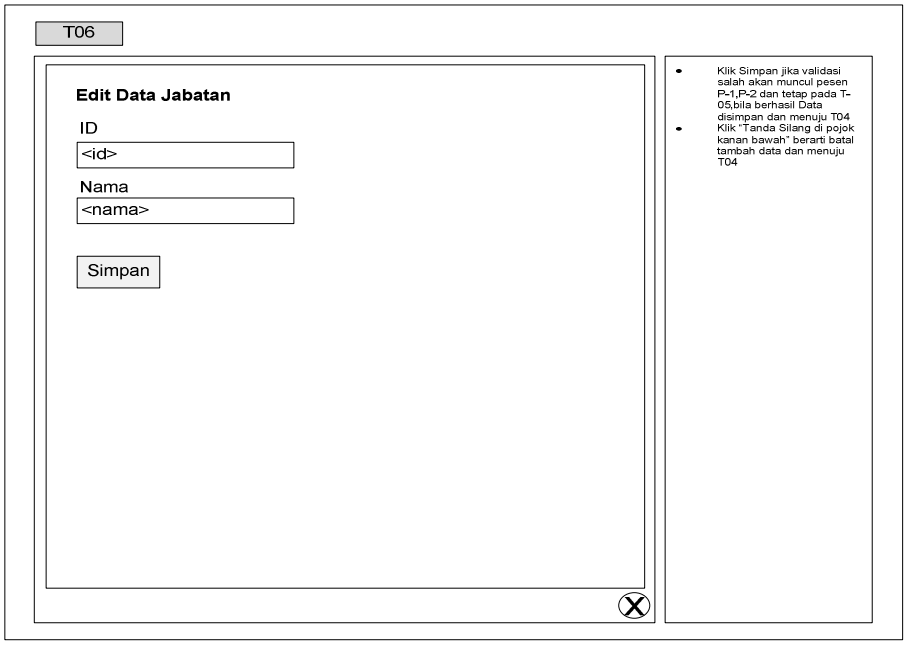
Gambar 3. 68. Perancangan Tampilan Pengolahan Data Jabatan

e. Perancangan Antarmuka Tambah Data Jabatan



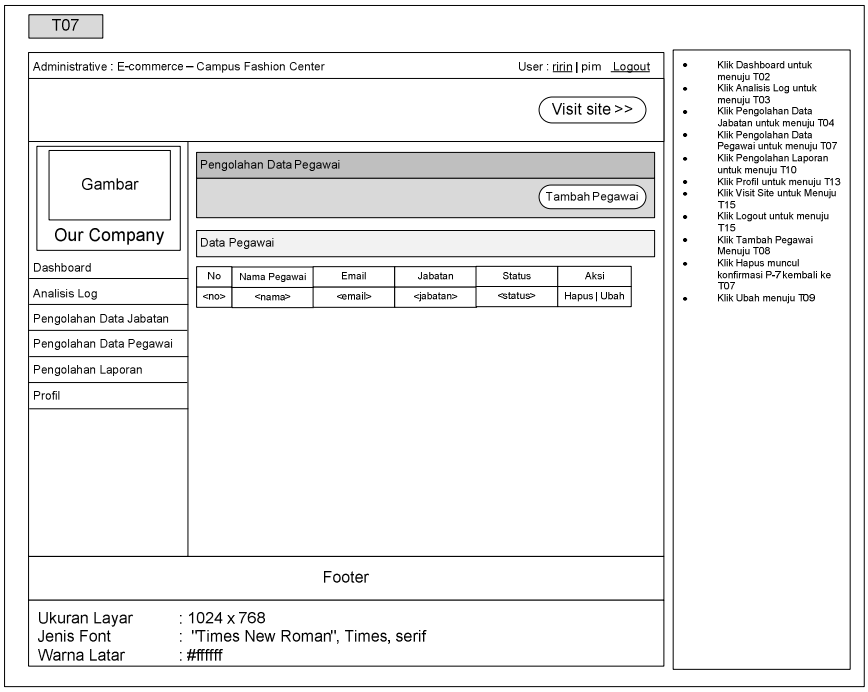
Gambar 3. 69. Perancangan Tampilan Tambah Data Jabatan

f. Perancangan Antarmuka Ubah Data Jabatan



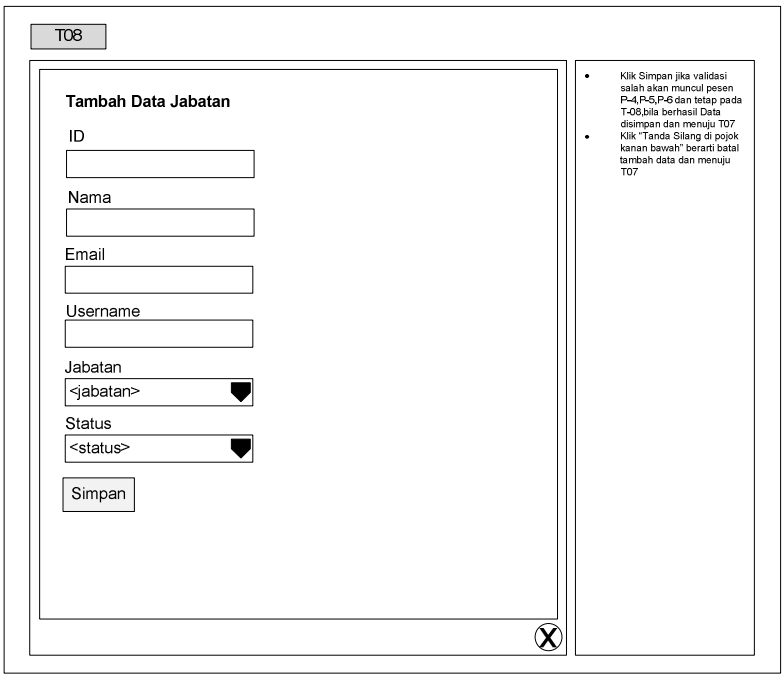
Gambar 3. 70. Perancangan Tampilan Ubah Data Jabatan

g. Perancangan Antarmuka Pengolahan Data Pegawai



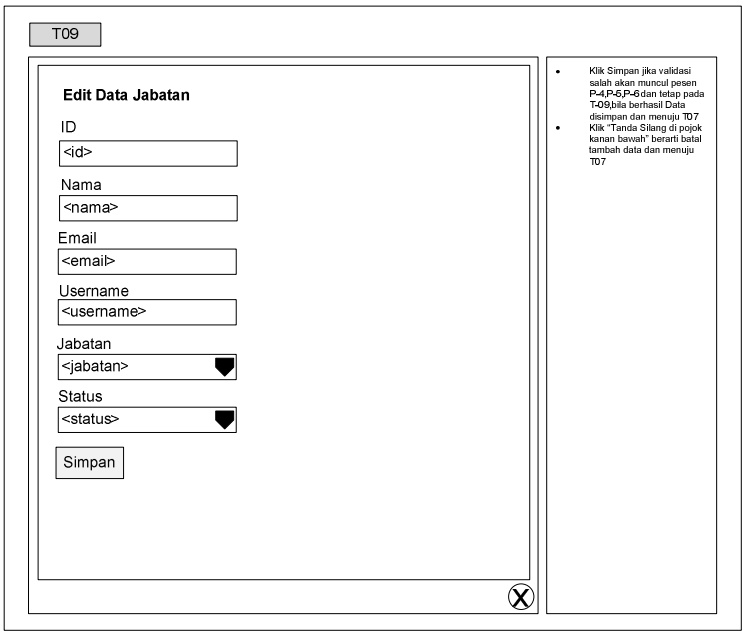
Gambar 3. 71. Perancangan Tampilan Pengolahan Data Pegawai

h. Perancangan Antarmuka Tambah Data Pegawai



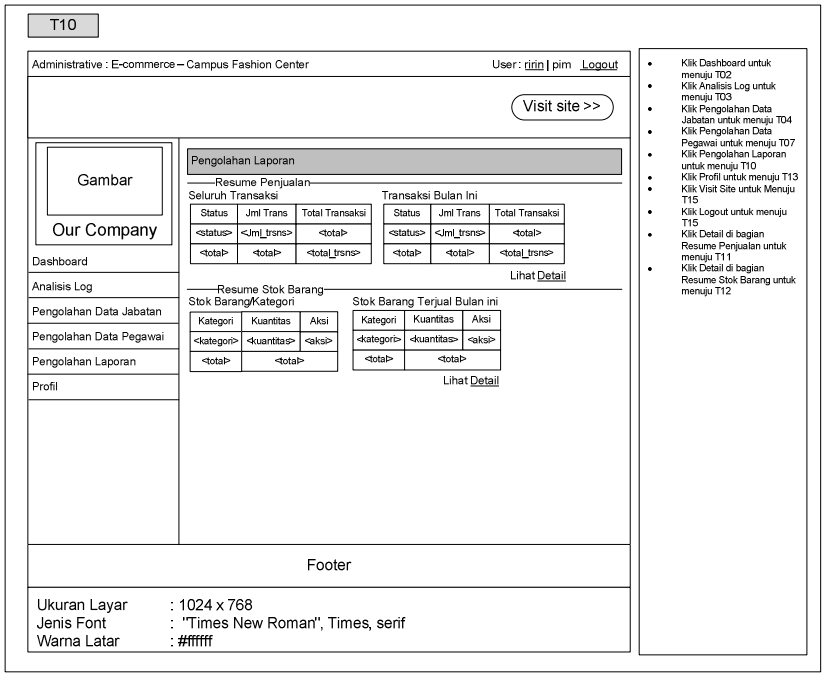
Gambar 3. 72. Perancangan Tampilan Tambah Data Pegawai

i. Perancangan Antarmuka Ubah Data Pegawai



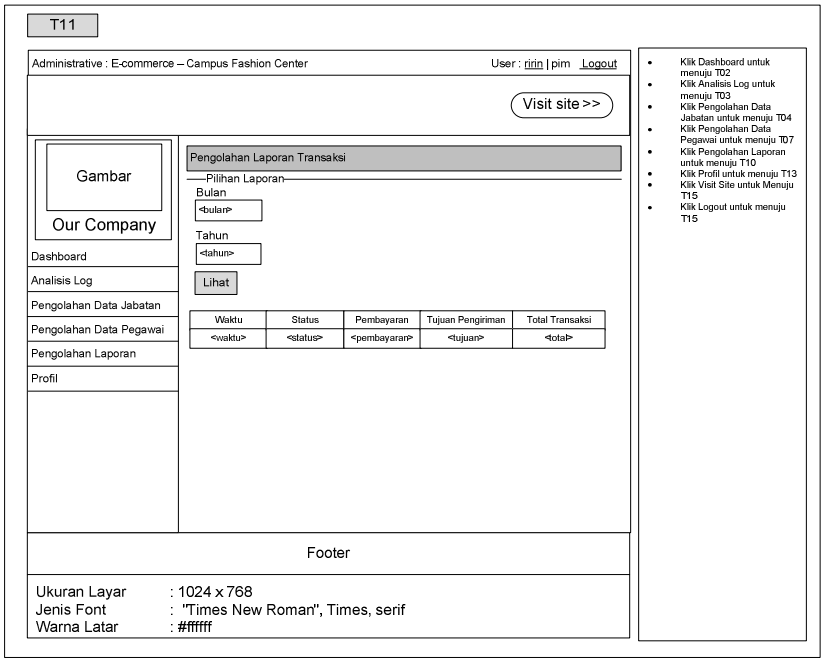
Gambar 3. 73. Perancangan Tampilan Ubah Data Pegawai

j. Perancangan Antarmuka Pengolahan Laporan



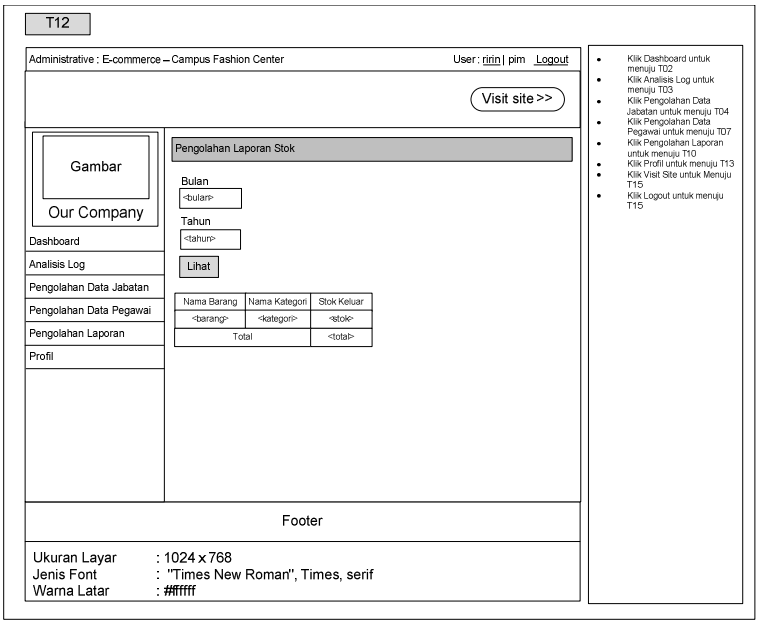
Gambar 3. 74. Perancangan Tampilan Pengolahan Laporan

k. Perancangan Antarmuka Pengolahan Laporan Resume Penjualan



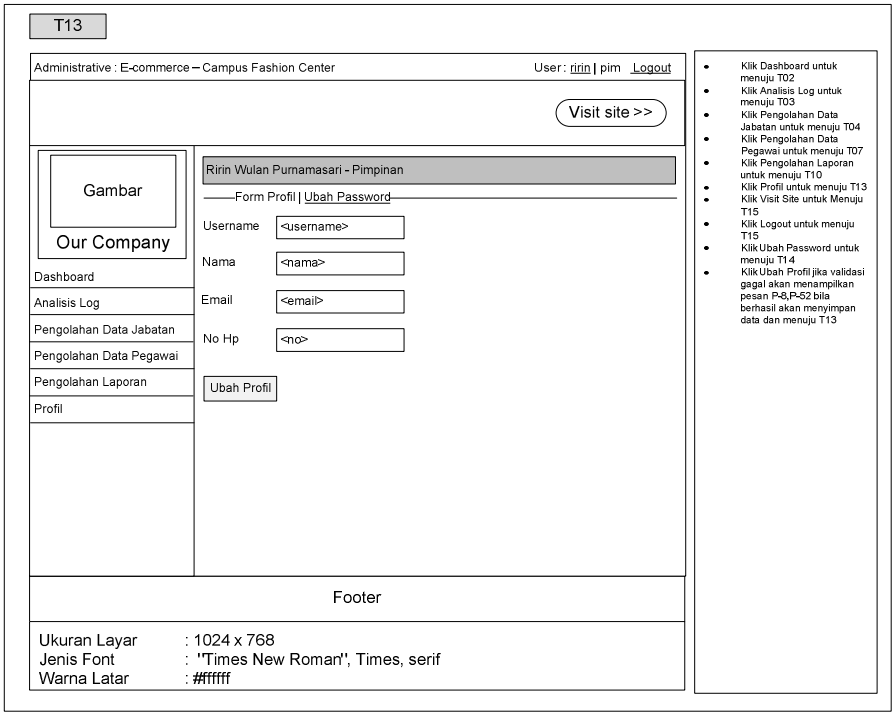
Gambar 3. 75. Perancangan Tampilan Laporan *Resume* Penjualan

l. Perancangan Antarmuka Pengolahan Laporan Resume Stok Barang



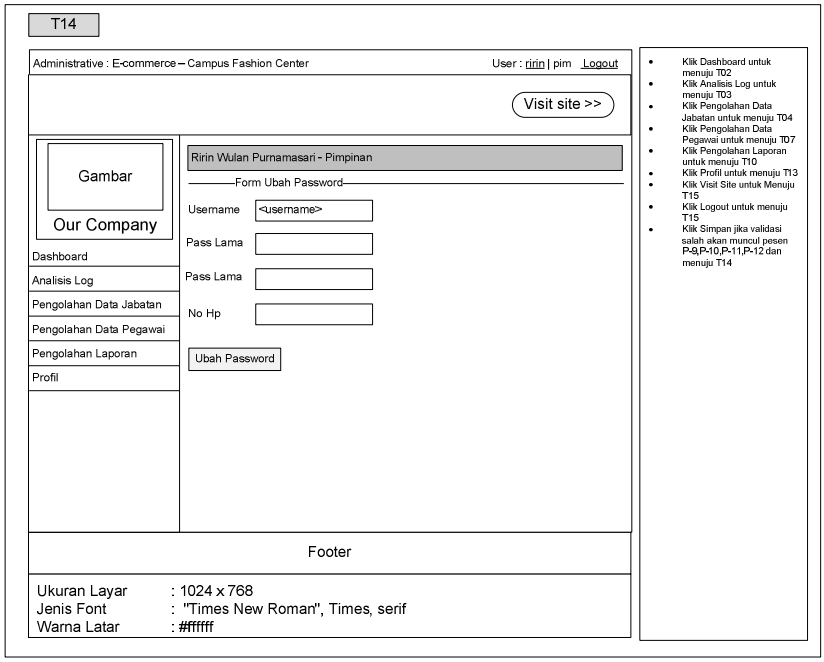
Gambar 3. 76. Perancangan Tampilan Laporan *Resume* Stok Barang

m. Perancangan Antarmuka Profil



Gambar 3. 77. Perancangan Tampilan Profil

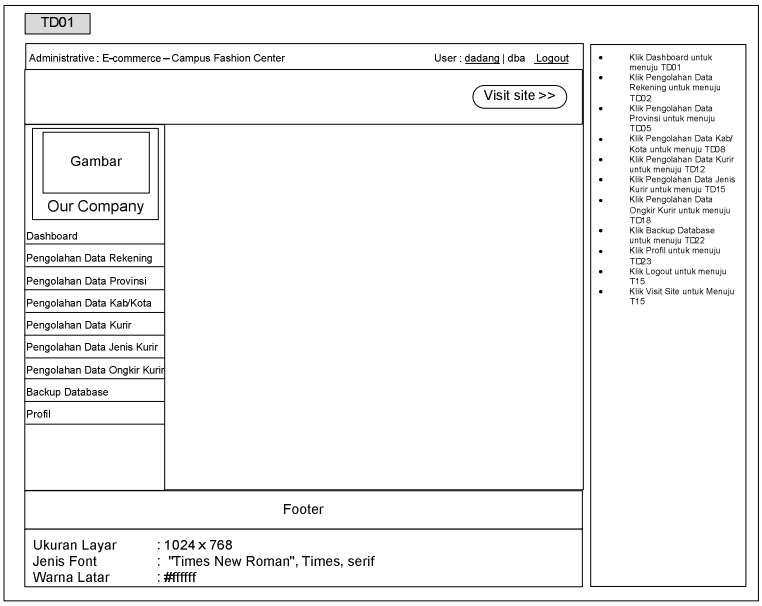
n. Perancangan Antarmuka Ubah *Password*



Gambar 3. 78. Perancangan Tampilan Ubah *Password*

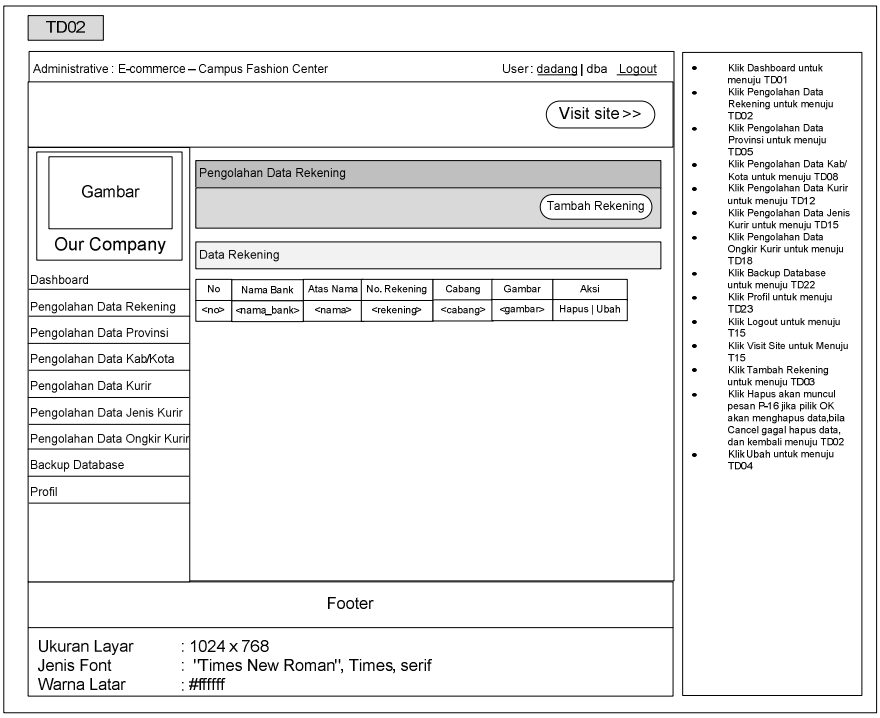
2. Perancangan Antarmuka *Database* Administrator

a. Perancangan Antarmuka Halaman Utama *Database* Administrator



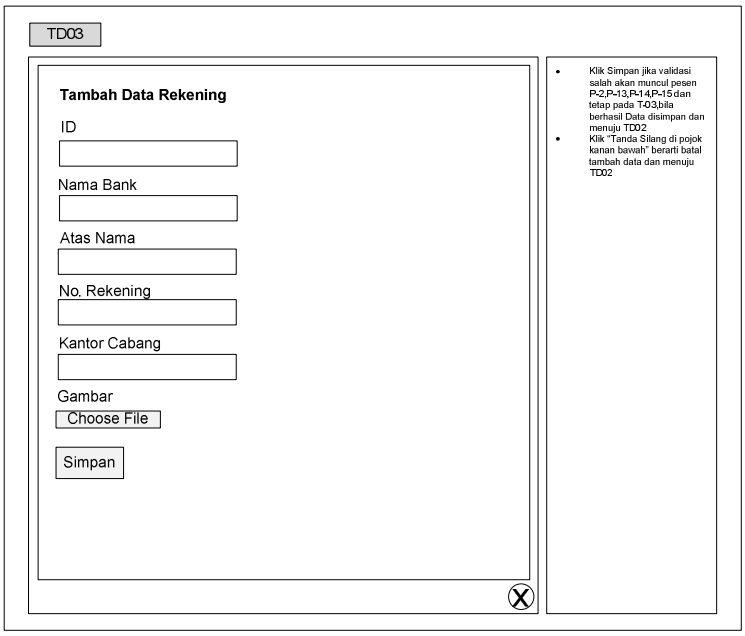
Gambar 3. 79. Perancangan Tampilan Halaman Utama

b. Perancangan Antarmuka Pengolahan Data Rekening



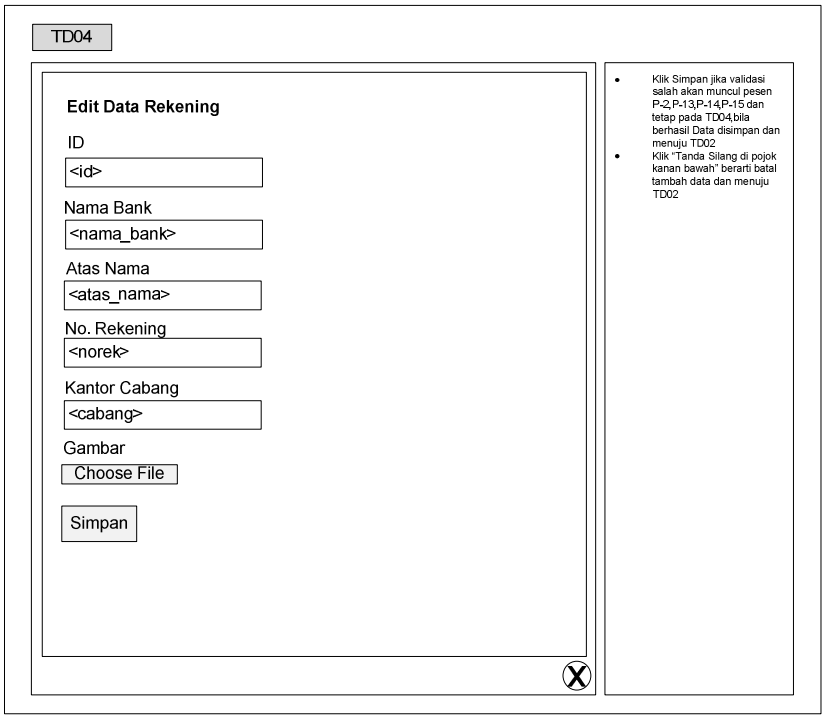
Gambar 3. 80. Perancangan Tampilan Data Rekening

c. Perancangan Antarmuka Tambah Data Rekening



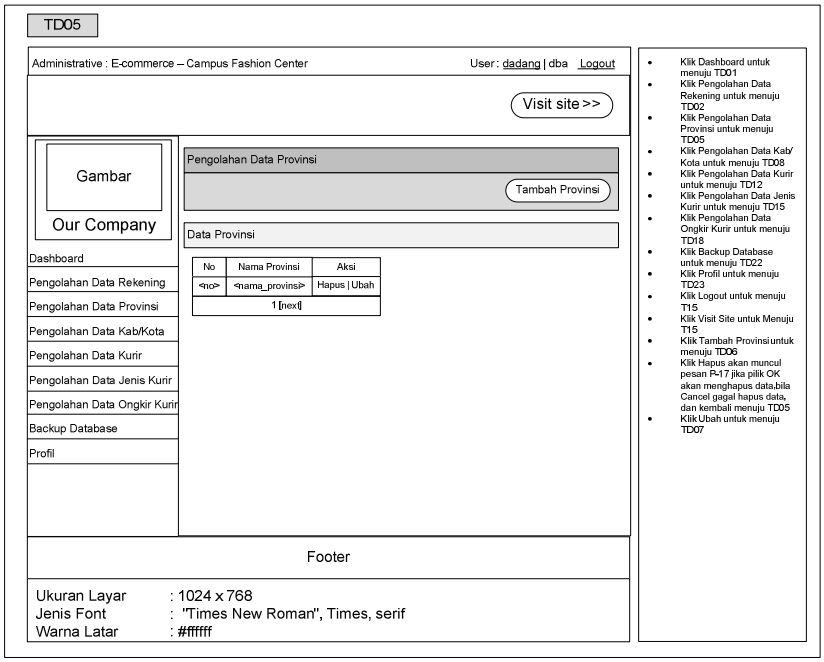
Gambar 3. 81. Perancangan Tampilan Tambah Data Rekening

d. Perancangan Antarmuka Ubah Data Rekening



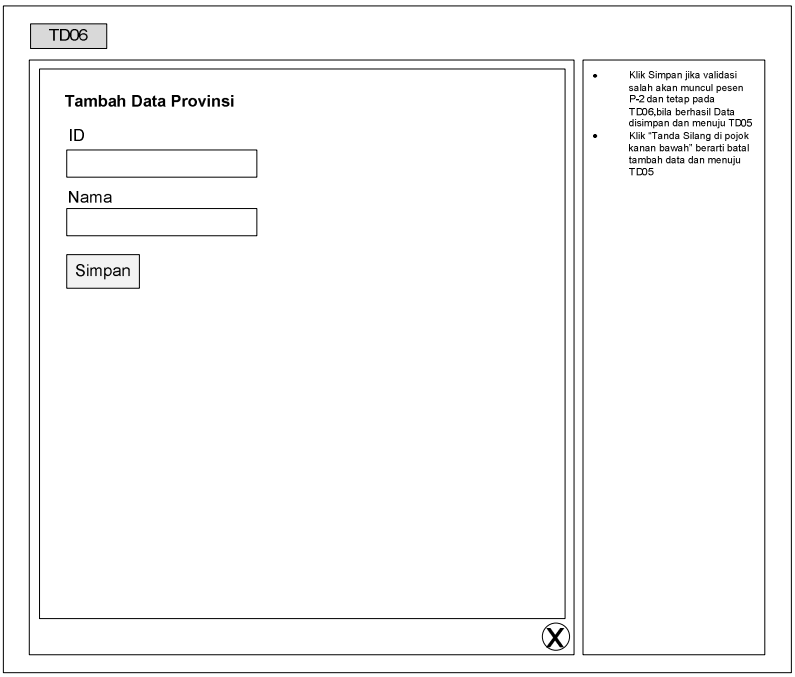
Gambar 3. 82. Perancangan Tampilan Ubah Rekening

e. Perancangan Antarmuka Pengolahan Data Provinsi



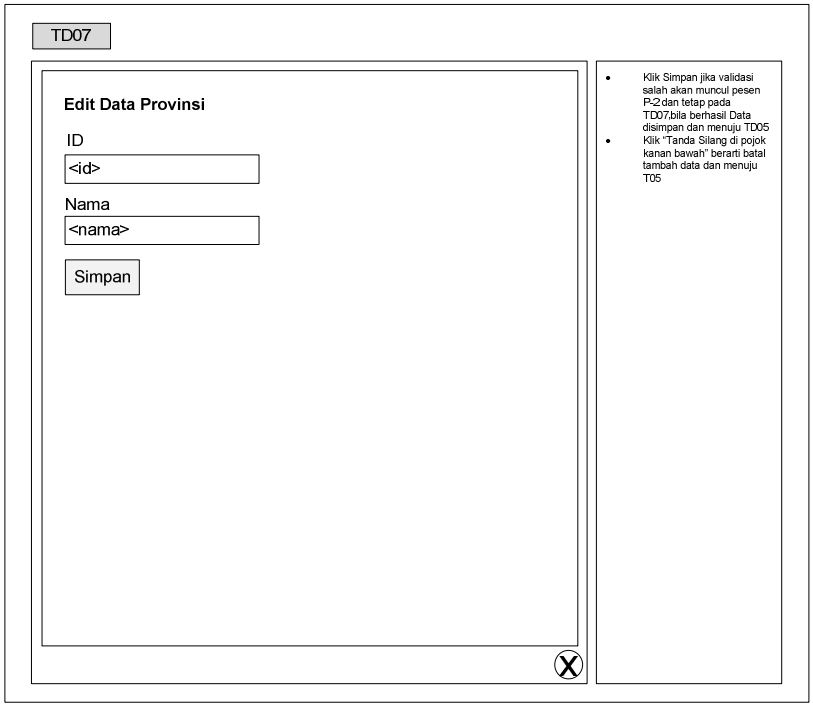
Gambar 3. 83. Perancangan Tampilan Data Provinsi

f. Perancangan Antarmuka Tambah Data Provinsi



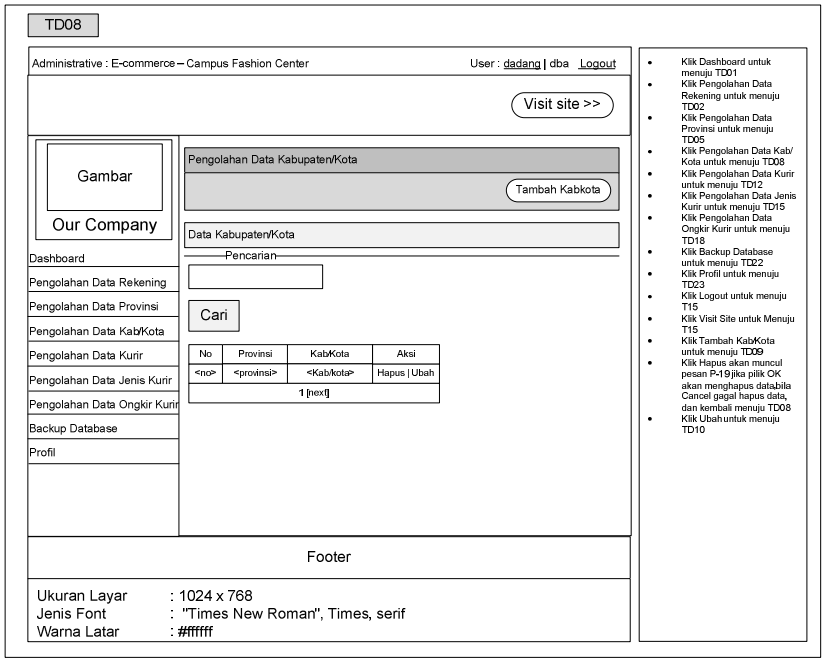
Gambar 3. 84. Perancangan Tampilan Tambah Provinsi

g. Perancangan Antarmuka Ubah Data Provinsi



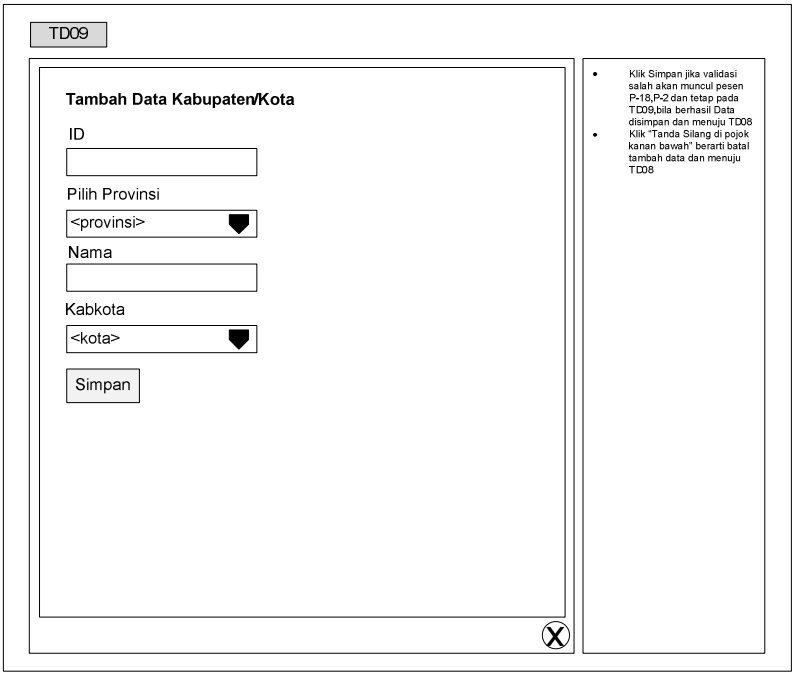
Gambar 3. 85. Perancangan Antarmuka Ubah Provinsi

h. Perancangan Antarmuka Pengolahan Data Kab/Kota



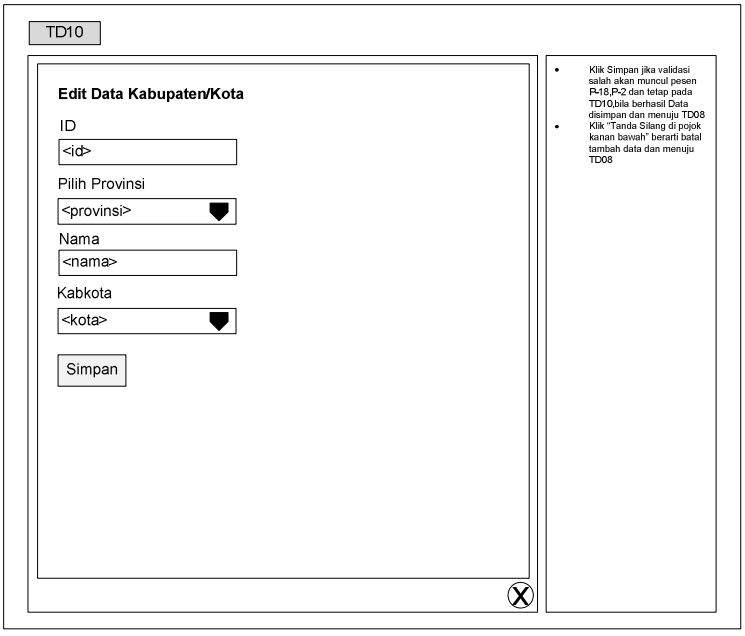
Gambar 3. 86. Perancangan Tampilan Pengolahan Data Kab/Kota

i. Perancangan Antarmuka Tambah Data Kab/Kota



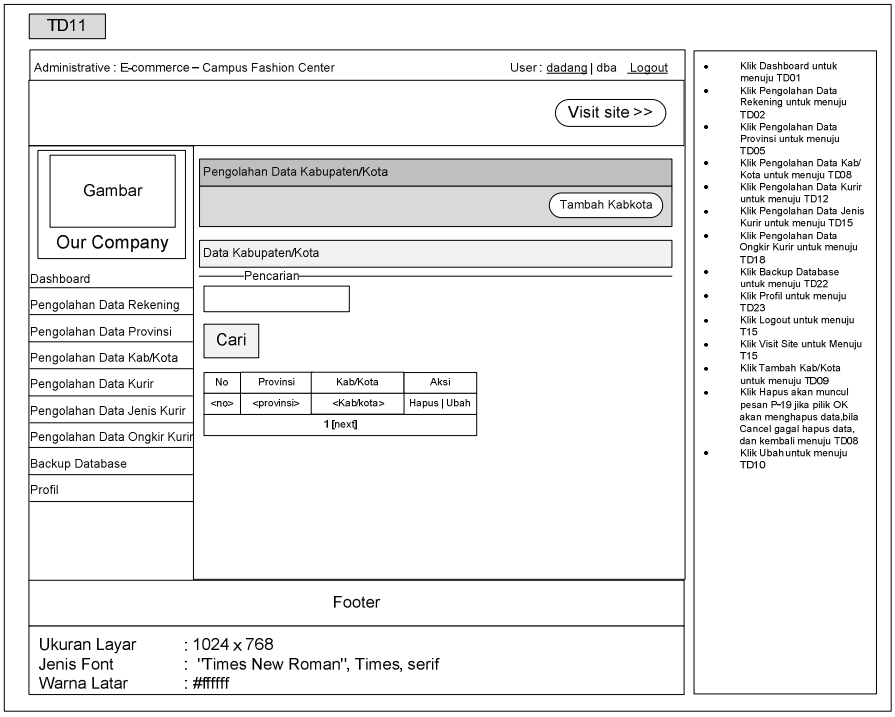
Gambar 3. 87. Perancangan Tampilan Tambah Kab/Kota

j. Perancangan Antarmuka Ubah Data Kab/Kota



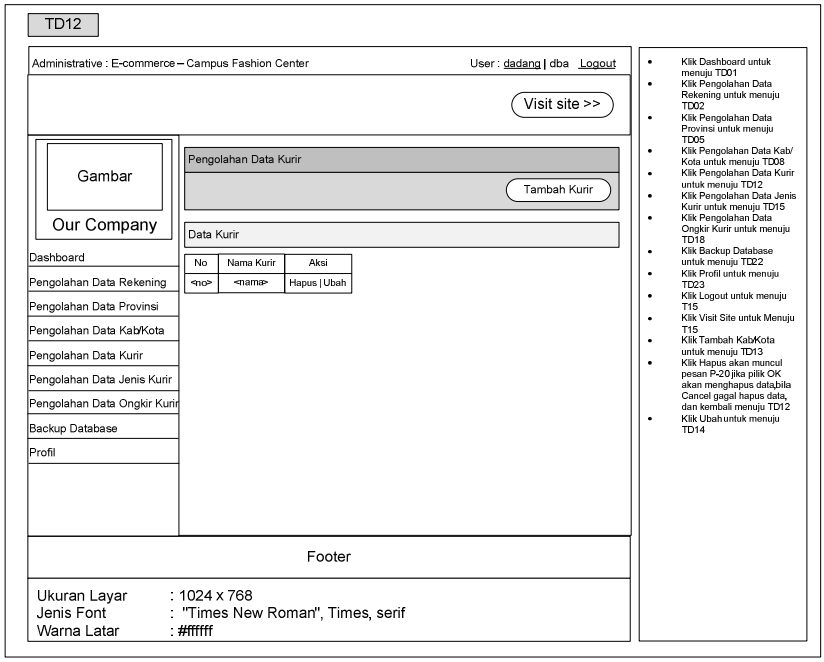
Gambar 3. 88. Perancangan Tampilan Ubah Kab/Kota

k. Perancangan Antarmuka Cari Data Kab/Kota



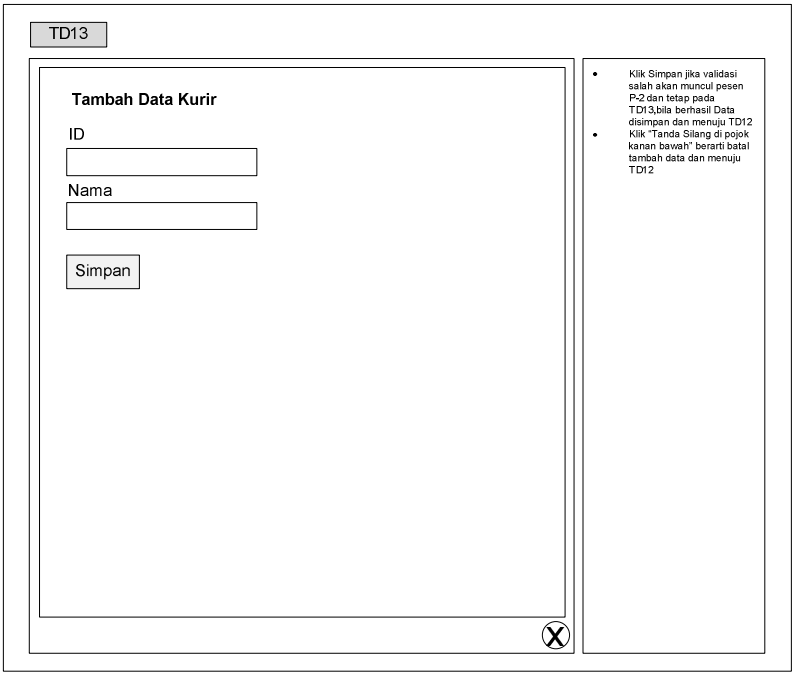
Gambar 3. 89. Perancangan Tampilan Cari Kab/Kota

l. Perancangan Antarmuka Pengolahan Data Kurir



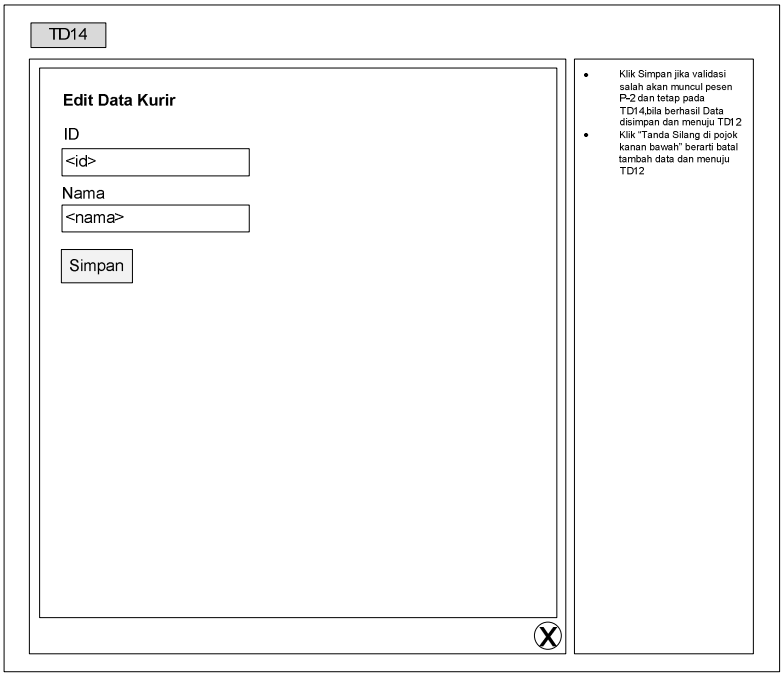
Gambar 3. 90. Perancangan Tampilan Pengolahan Data Kurir

m. Perancangan Antarmuka Tambah Data Kurir



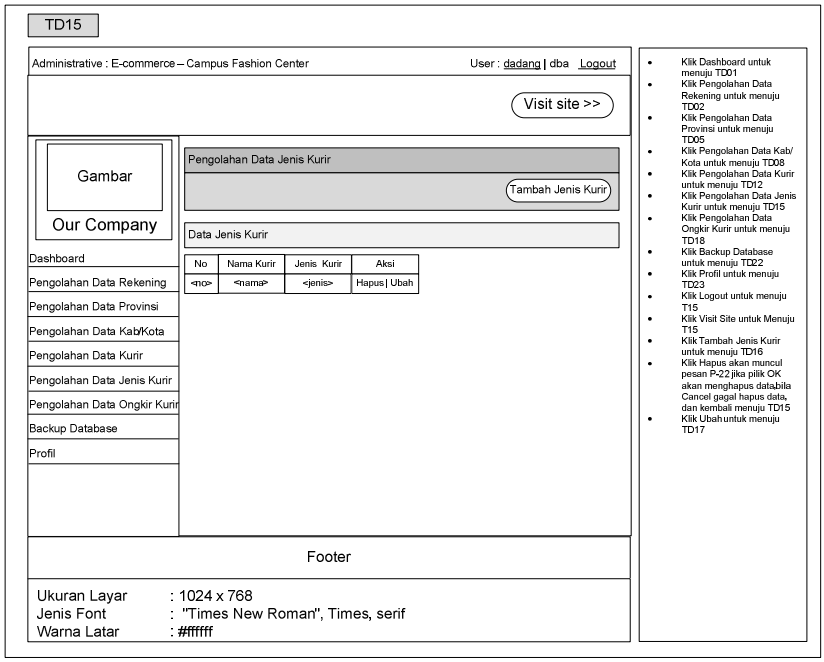
Gambar 3. 91. Perancangan Tampilan Antarmuka Tambah Kurir

n. Perancangan Antarmuka Ubah Data Kurir



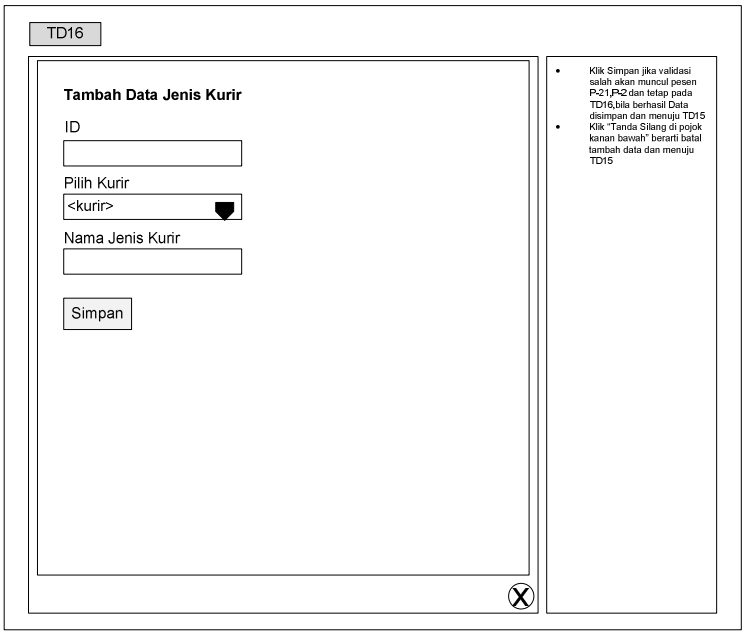
Gambar 3. 92. Perancangan Tampilan Edit Kurir

o. Perancangan Antarmuka Pengolahan Data Jenis Kurir



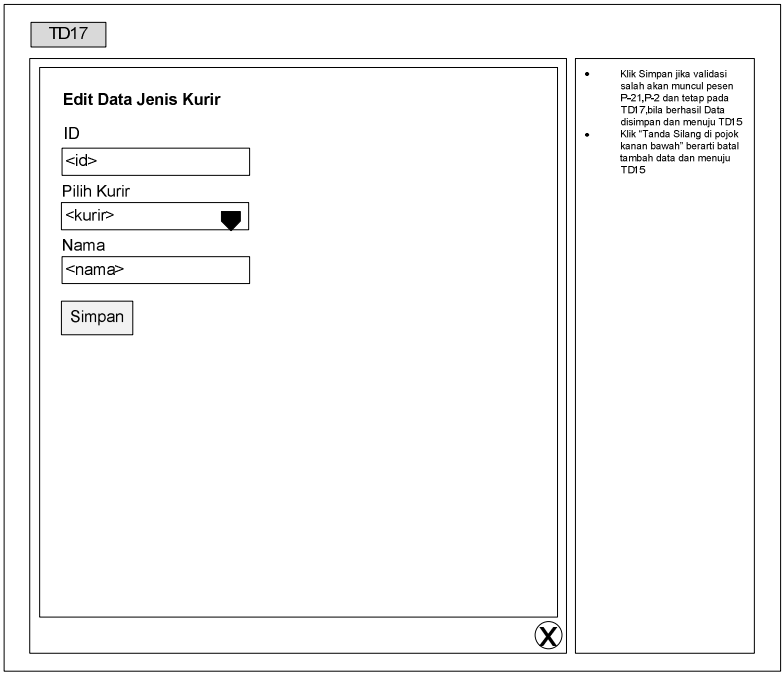
Gambar 3. 93. Perancangan Tampilan Data Jenis Kurir

p. Perancangan Antarmuka Tambah Data Jenis Kurir



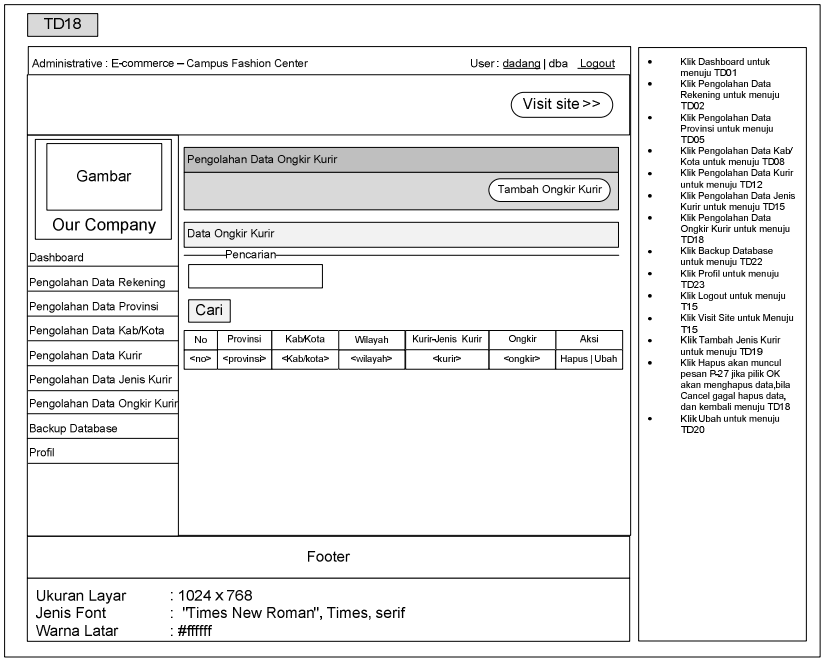
Gambar 3. 94. Perancangan Tampilan Tambah Jenis Kurir

q. Perancangan Antarmuka Ubah Data Jenis Kurir



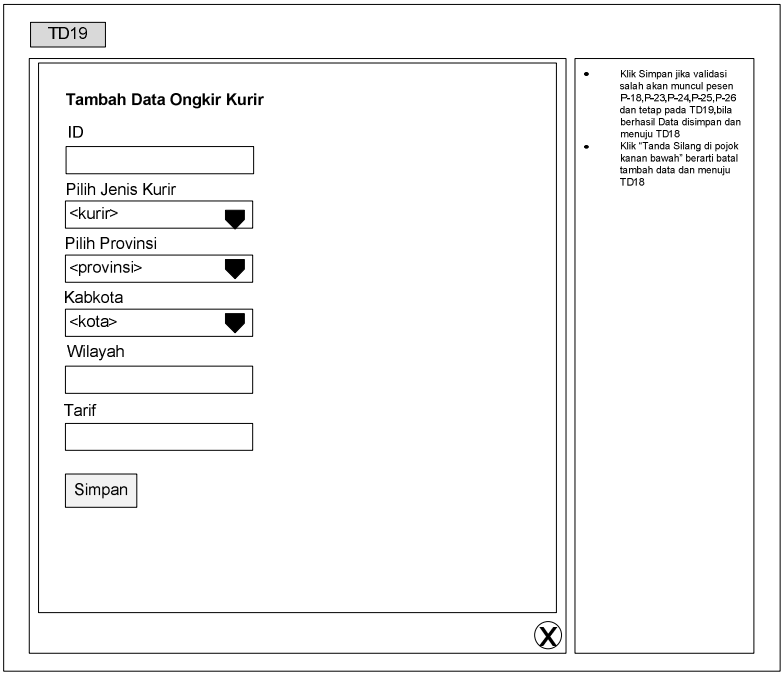
Gambar 3. 95. Perancangan Tampilan Ubah Jenis Kurir

r. Perancangan Antarmuka Pengolahan Data Ongkir Kurir



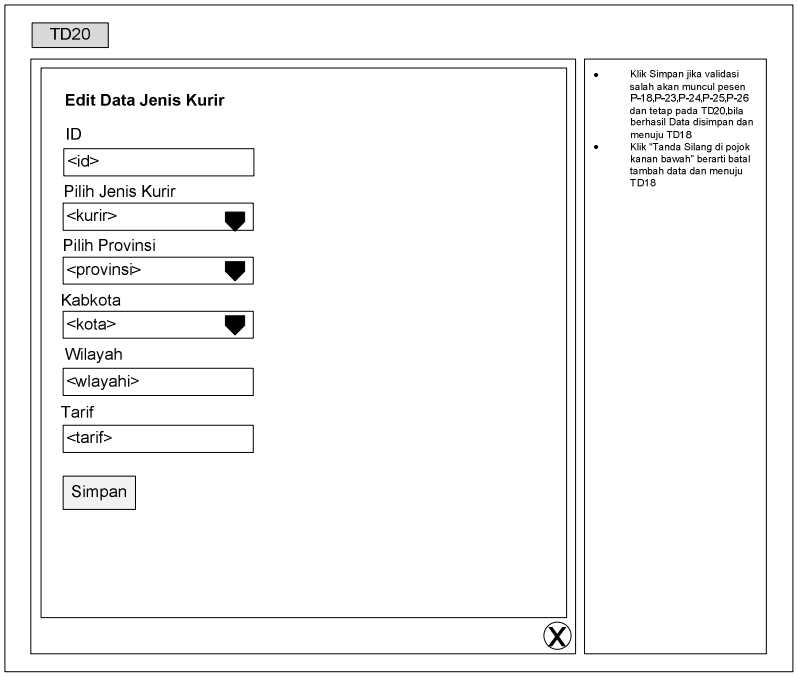
Gambar 3. 96. Penrancangan Tampilan Data Ongkir Kurir

s. Perancangan Antarmuka Tambah Data Ongkir Kurir



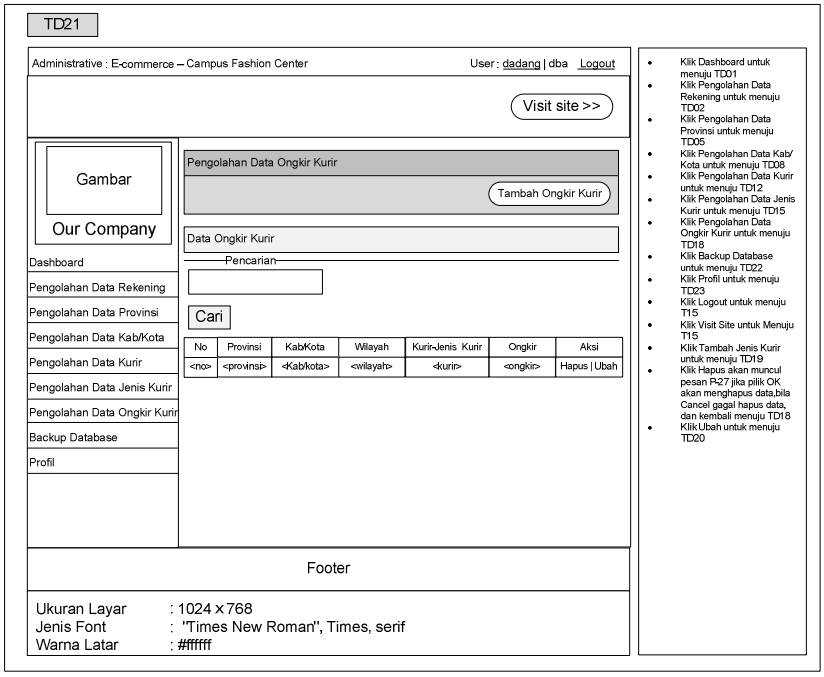
Gambar 3. 97. Perancangan Tampilan Tambah Ongkir Kurir

t. Perancangan Antarmuka Ubah Data Ongkir Kurir



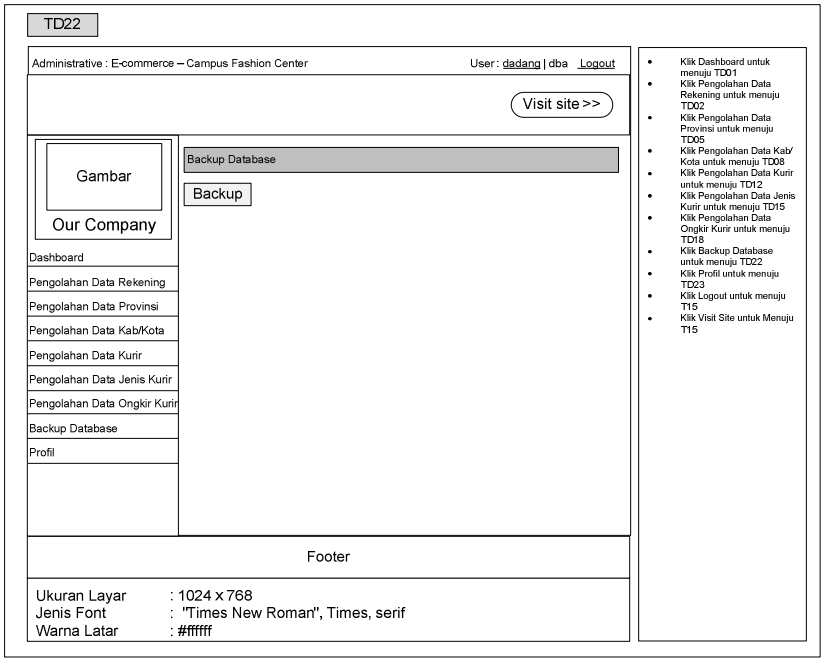
Gambar 3. 98. Perancangan Tampilan Ubah Ongkir Kurir

u. Perancangan Antarmuka Cari Data Ongkir Kurir



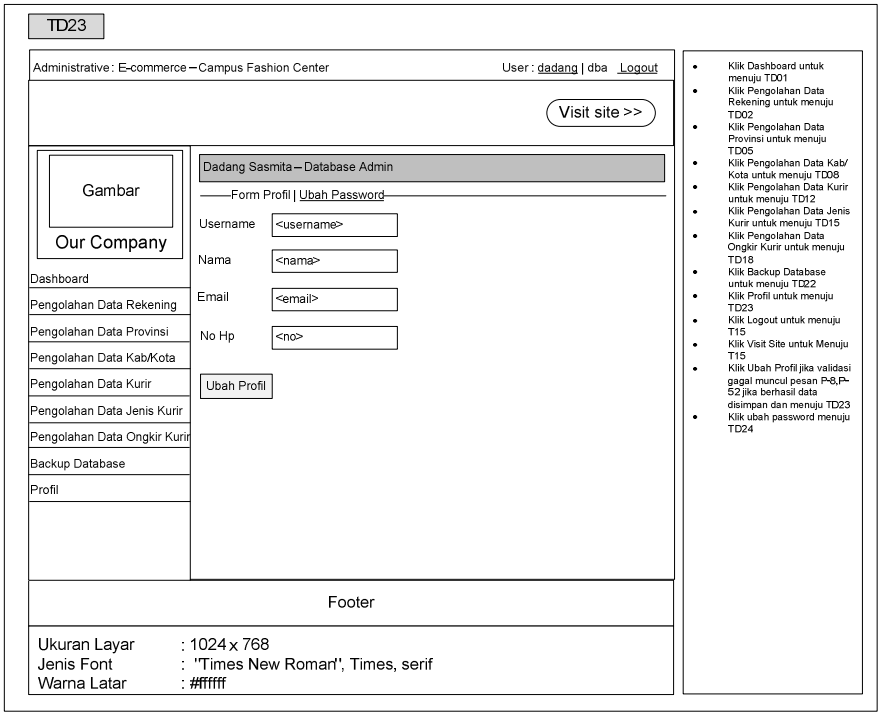
Gambar 3. 99. Perancangan Tampilan Cari Data Ongkir

v. Perancangan Antarmuka *Backup* Data



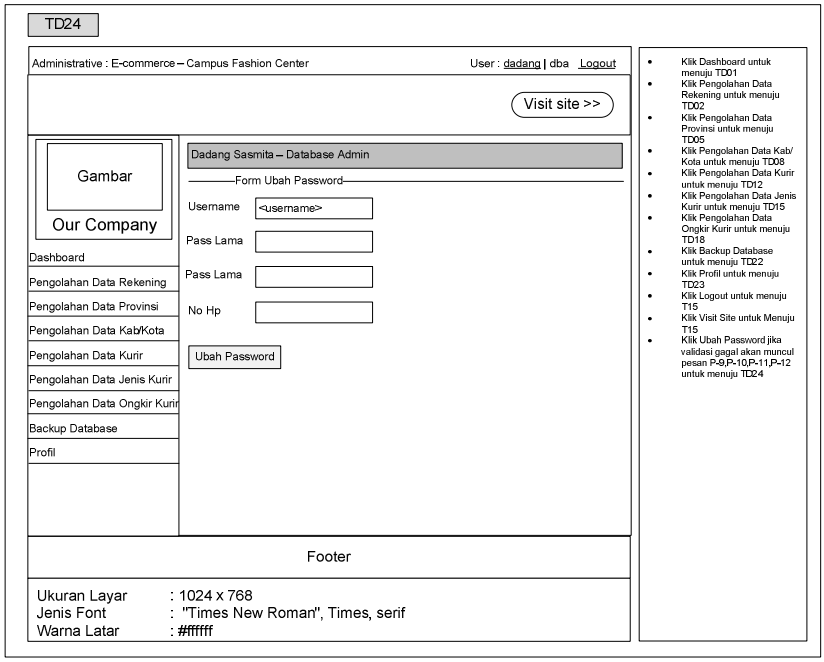
Gambar 3. 100. Perancangan Tampilan *Backup* Data

w. Perancangan Antarmuka Profil



Gambar 3. 101. Perancangan Antarmuka Profil

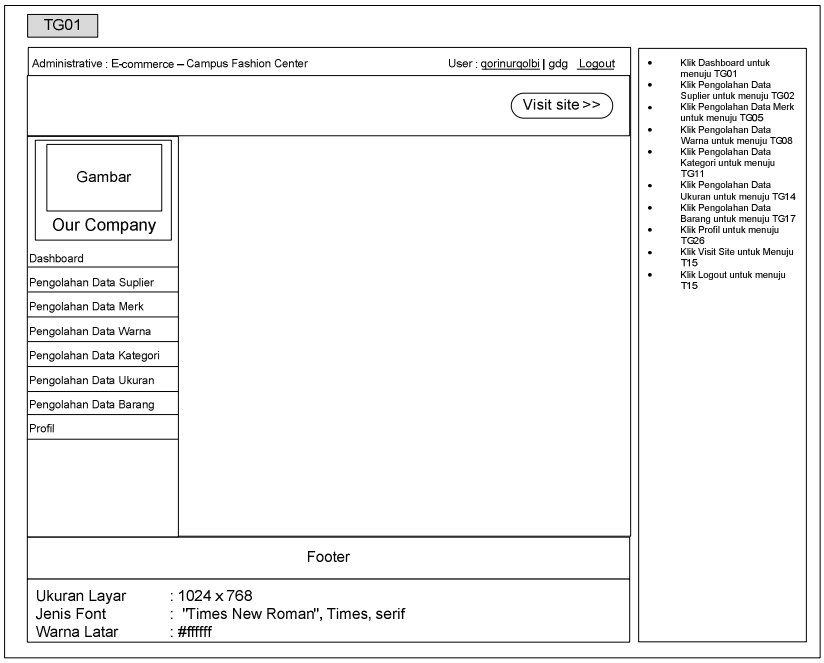
x. Perancangan Antarmuka Ubah *Password*



Gambar 3. 102. Perancangan Tampilan Ubah *Password*

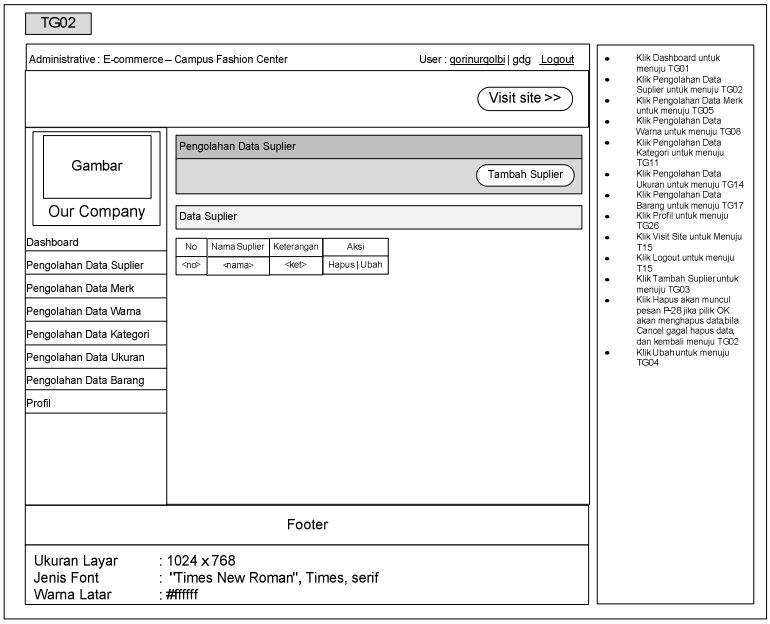
3. Perancangan Antarmuka Gudang

a. Perancangan Antarmuka Halaman Utama Gudang



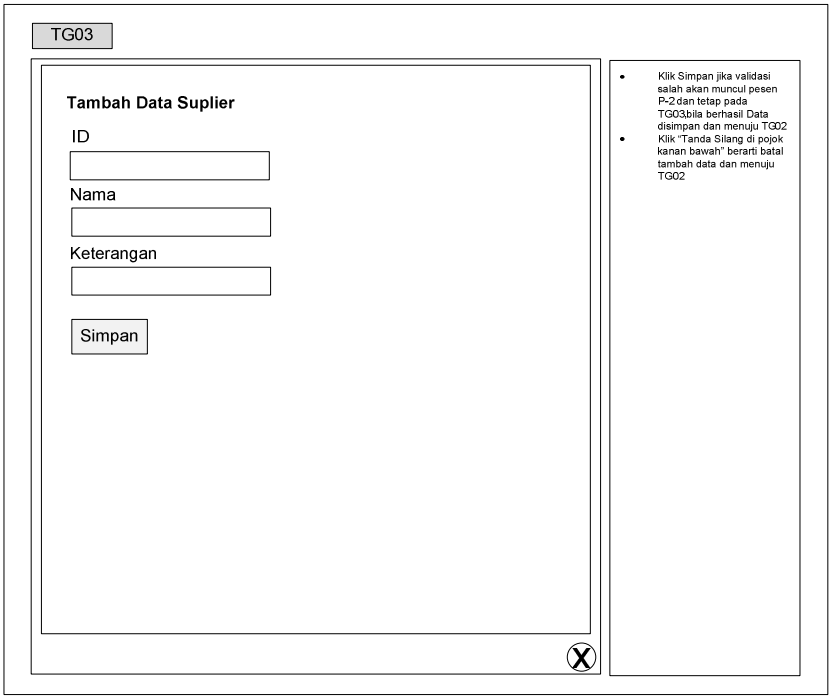
Gambar 3. 103. Perancangan Halaman Utama Gudang

b. Perancangan Antarmuka Pengolahan Data Suplier



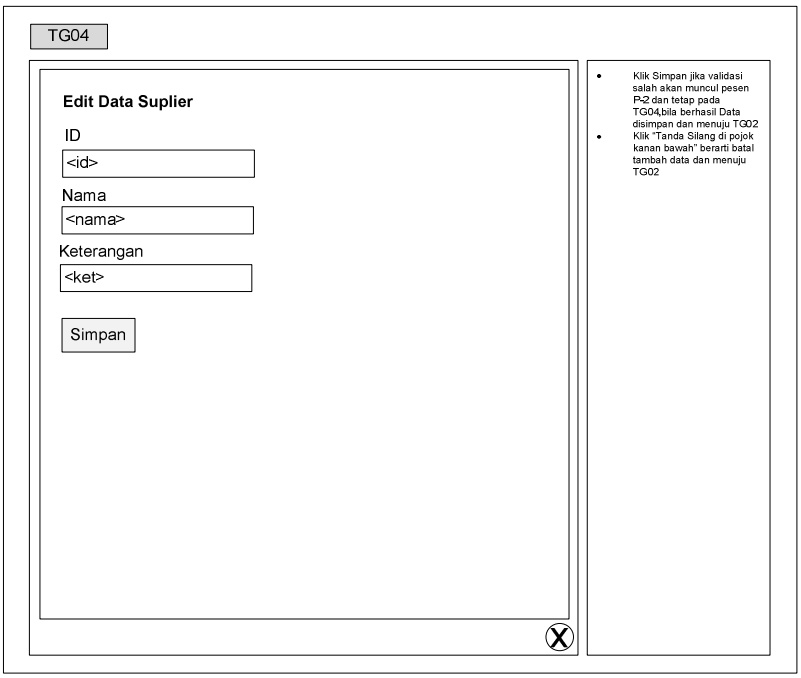
Gambar 3. 104. Perancangan Tampilan Pengolahan Data Suplier

c. Perancangan Antarmuka Tambah Suplier



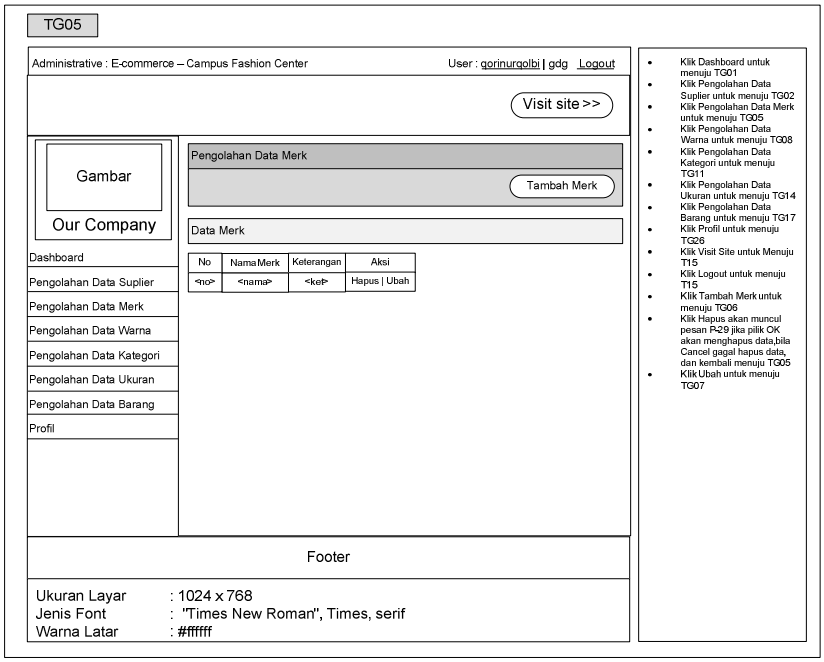
Gambar 3. 105. Perancangan Tampilan Tambah Suplier

d. Perancangan Antarmuka Ubah Suplier



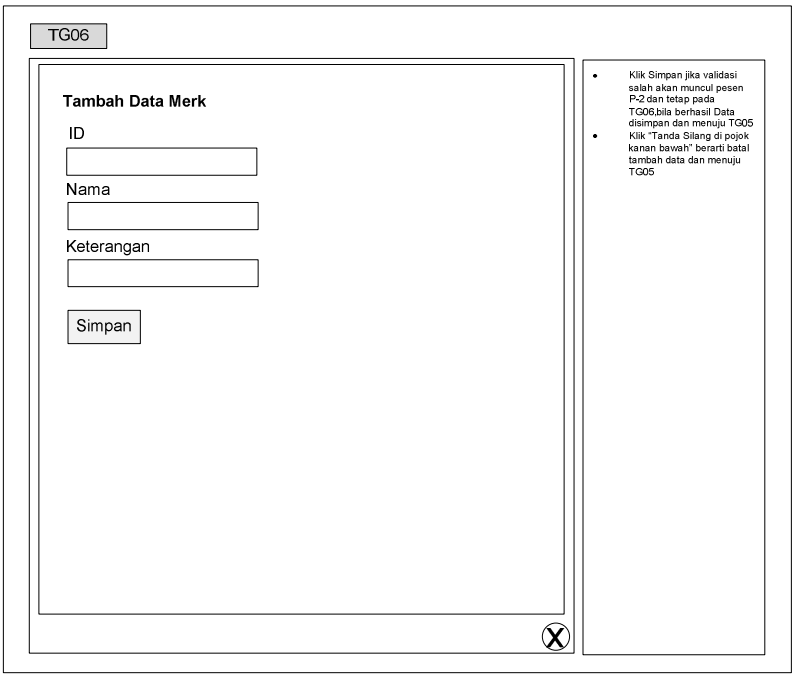
Gambar 3. 106. Perancangan Tampilan Ubah Suplier

e. Perancangan Antarmuka Pengolahan Data Merk



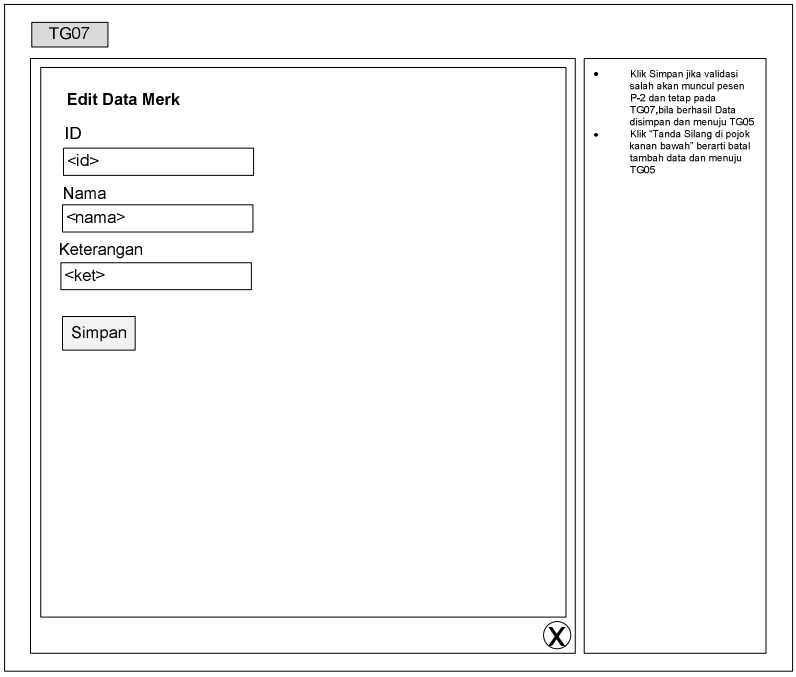
Gambar 3. 107. Perancangan Tampilan Pengolahan Data Merk

f. Perancangan Antarmuka Tambah Merk



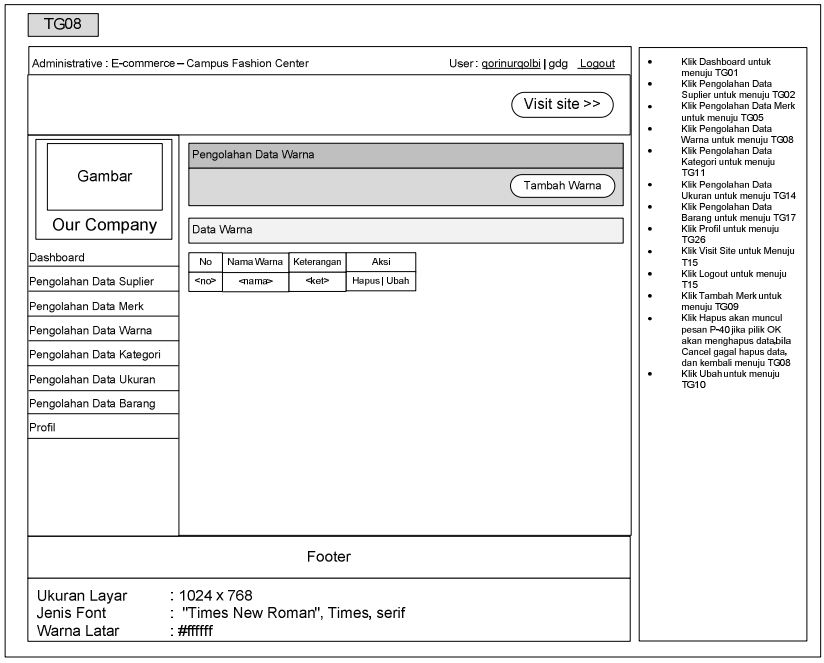
Gambar 3. 108. Perancangan Tampilan Tambah Merk

g. Perancangan Antarmuka Ubah Merk



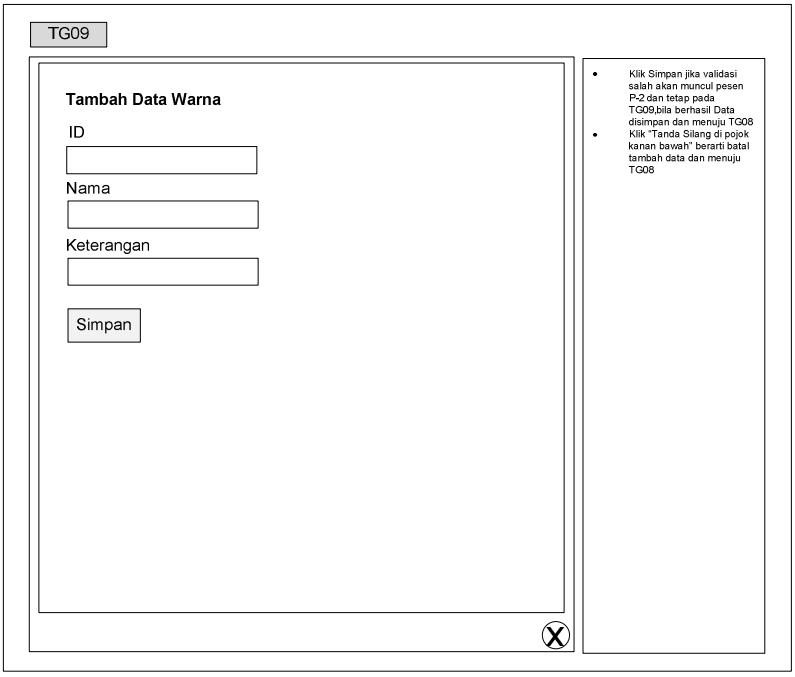
Gambar 3. 109. Perancangan Tampilan Ubah Merk

h. Perancangan Antarmuka Pengolahan Data Warna



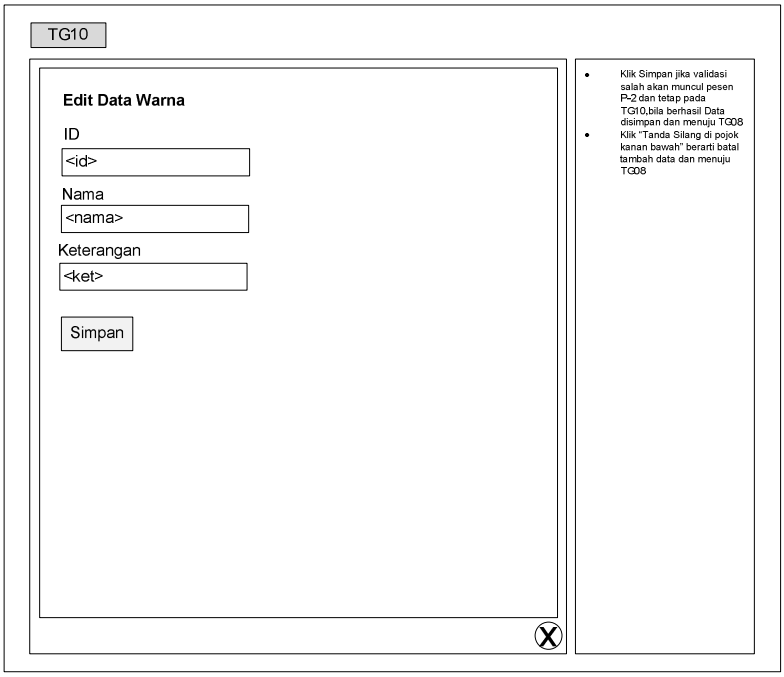
Gambar 3. 110. Perancangan Tampilan Data Warna

i. Perancangan Antarmuka Tambah Warna



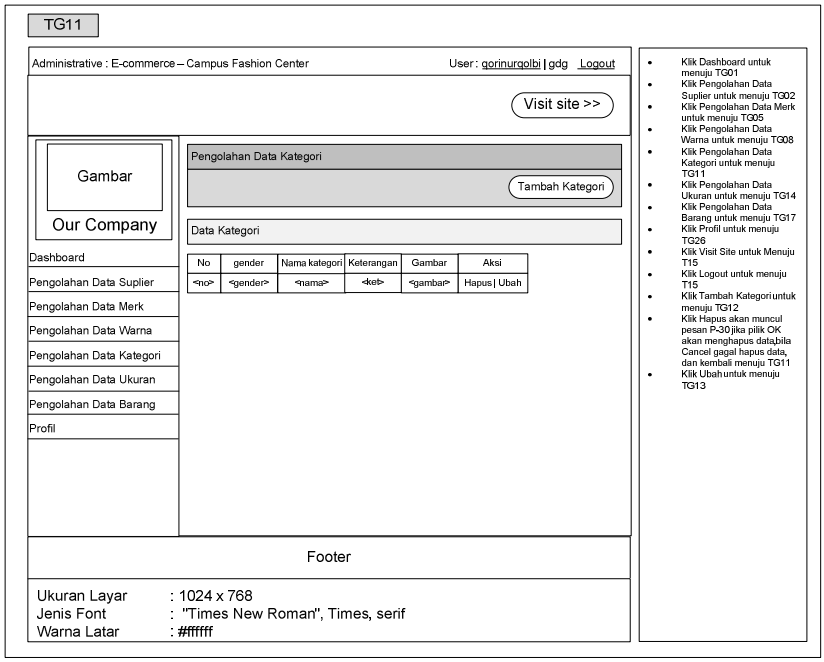
Gambar 3. 111. Perancangan Tampilan Tambah Warna

j. Perancangan Antarmuka Ubah Warna



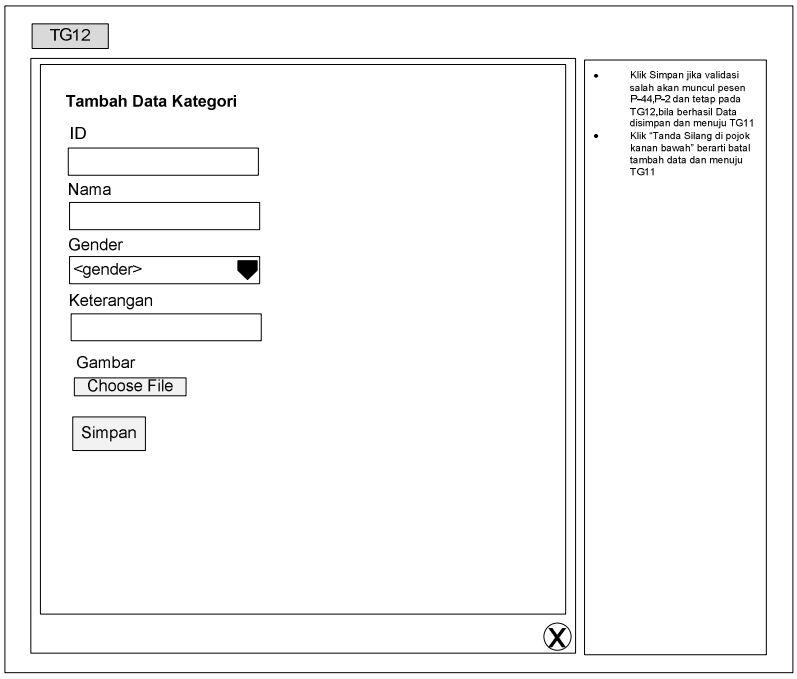
Gambar 3. 112. Perancangan Tampilan Ubah Warna

k. Perancangan Antarmuka Pengolahan Data Kategori



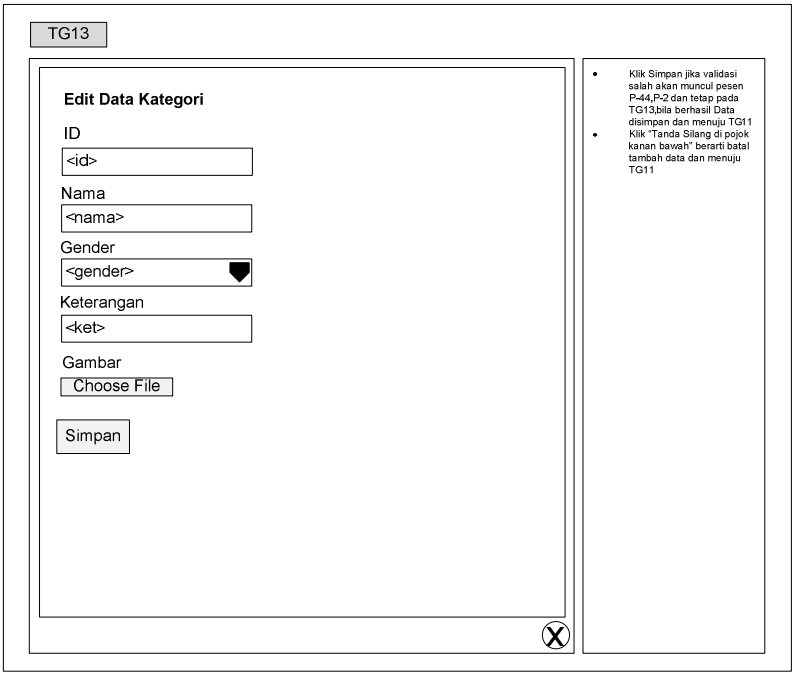
Gambar 3. 113. Perancangan Tampilan Pengolahan Data Kategori

l. Perancangan Antarmuka Tambah Kategori



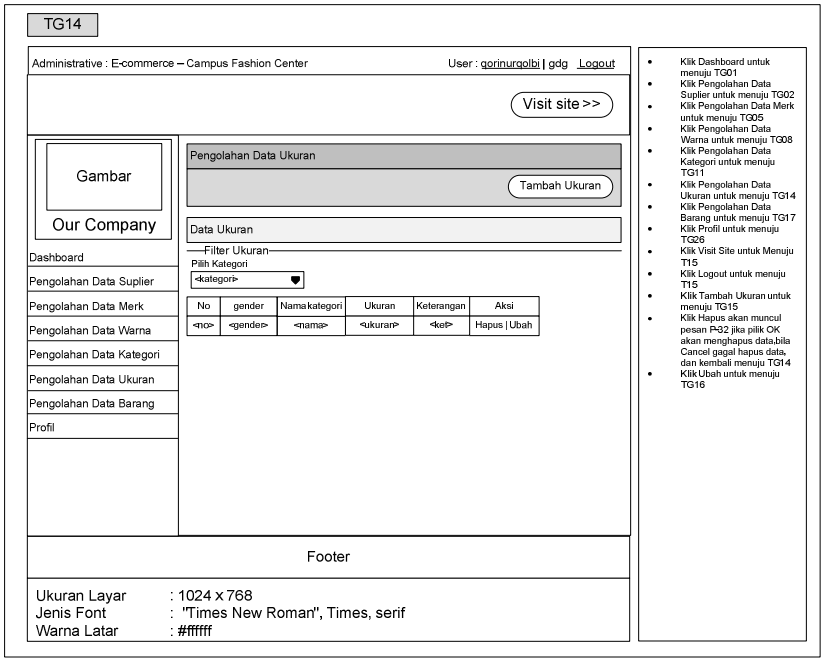
Gambar 3. 114. Perancangan Tampilan Tambah Kategori

m. Perancangan Antarmuka Ubah Kategori



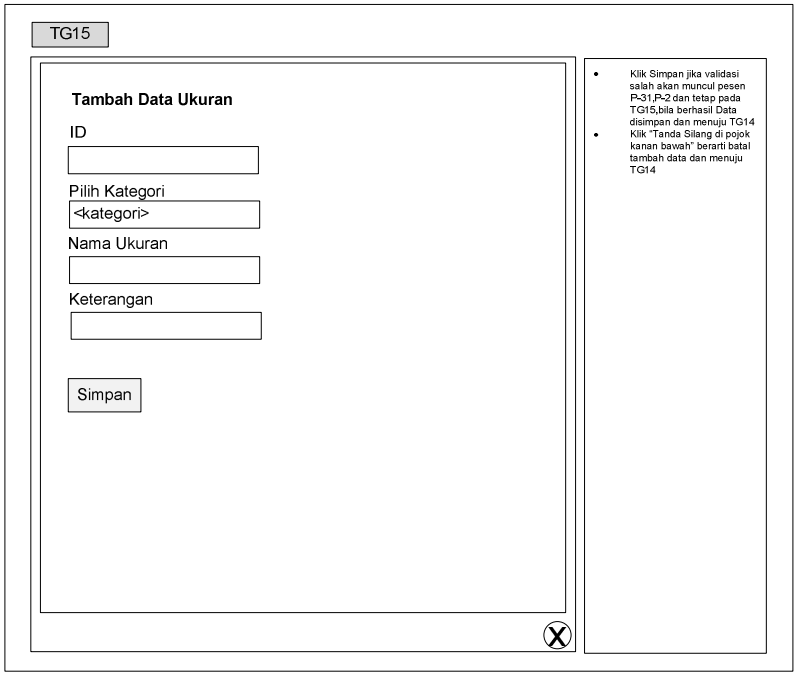
Gambar 3. 115. Perancangan Tampilan Ubah Kategori

n. Perancangan Antarmuka Pengolahan Data Ukuran



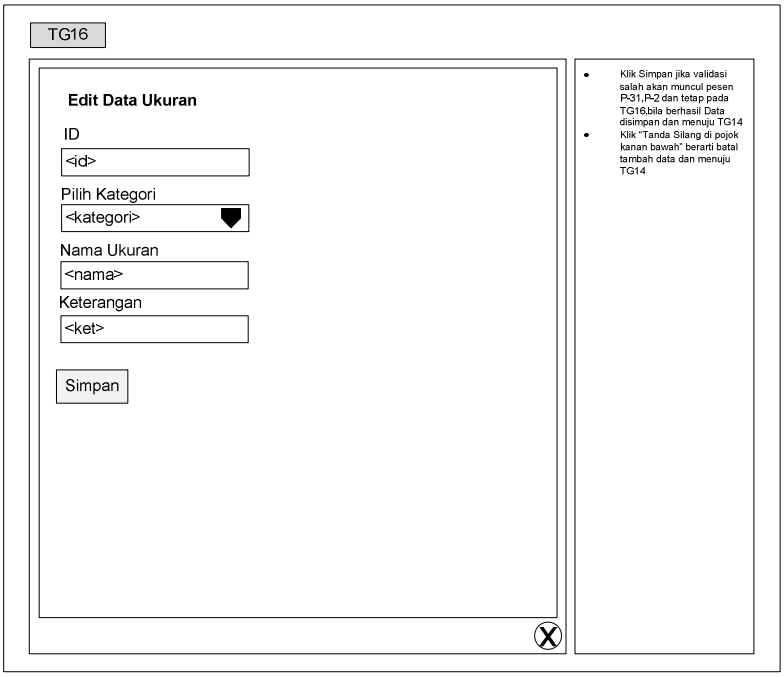
Gambar 3. 116. Perancangan Tampilan Pengolahan Data Ukuran

o. Perancangan Antarmuka Tambah Ukuran



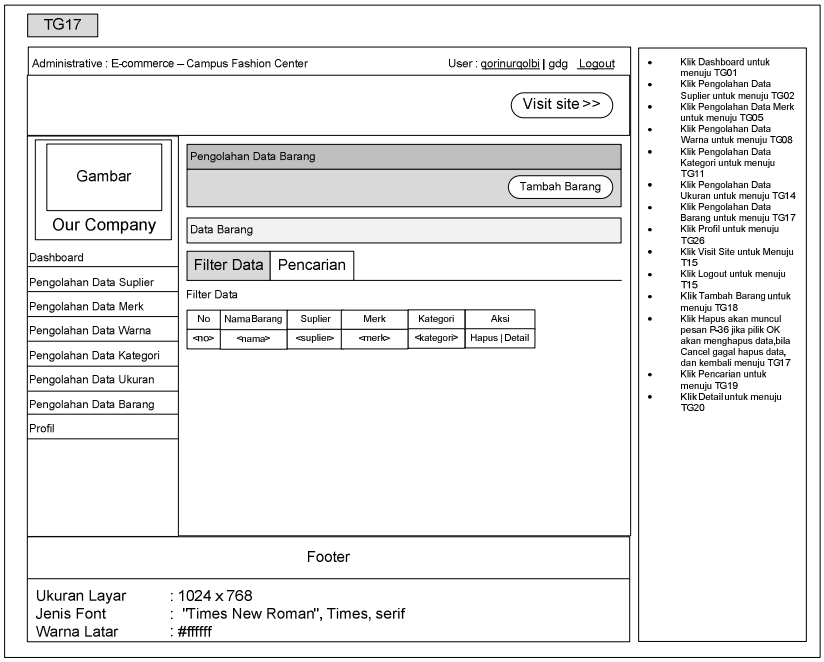
Gambar 3. 117. Perancangan Tampilan Tambah Ukuran

p. Perancangan Antarmuka Ubah Ukuran



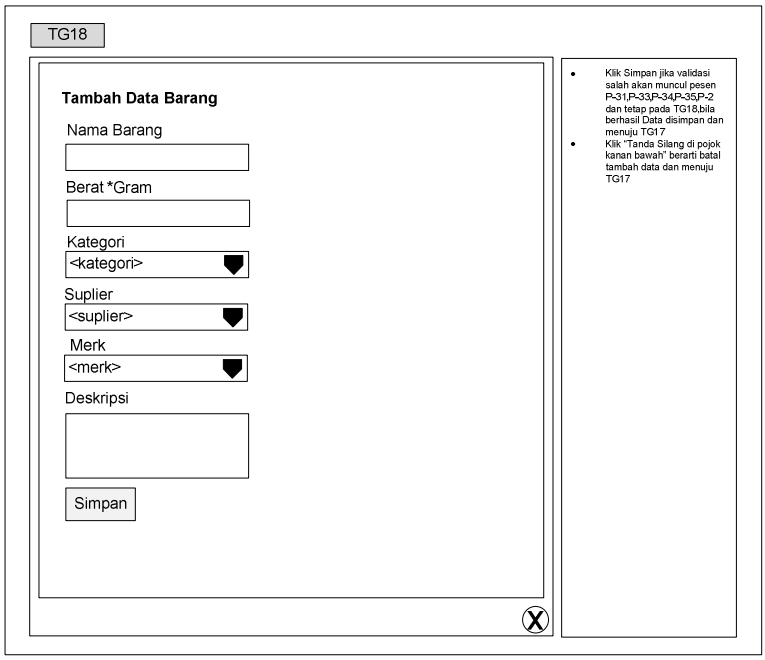
Gambar 3. 118. Perancangan Tampilan Ubah Ukuran

q. Perancangan Antarmuka Pengolahan Data Barang



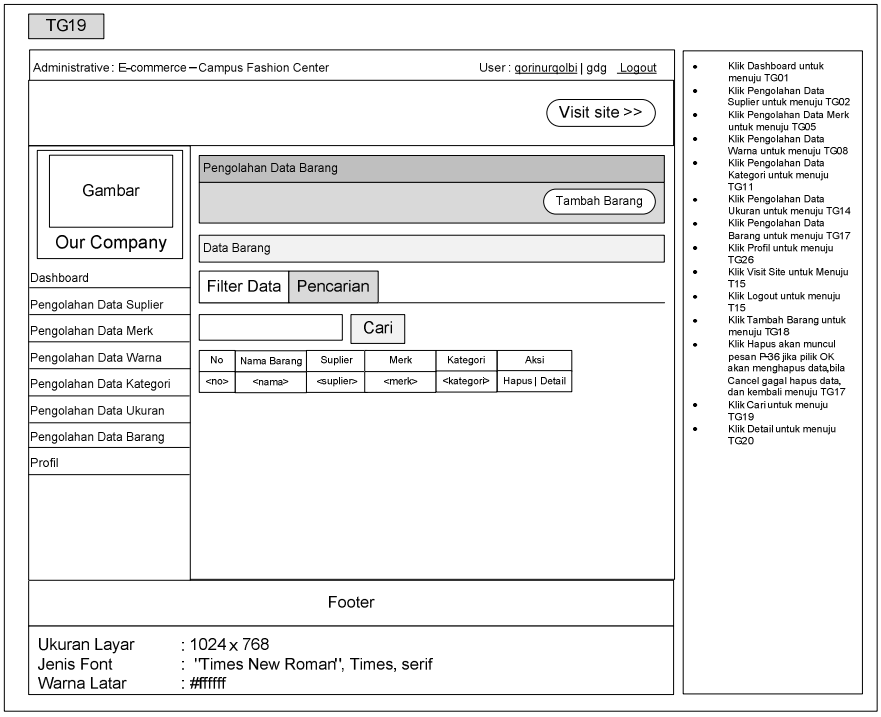
Gambar 3. 119. Perancangan Tampilan Pengolahan Data Barang

r. Perancangan Antarmuka Tambah Barang



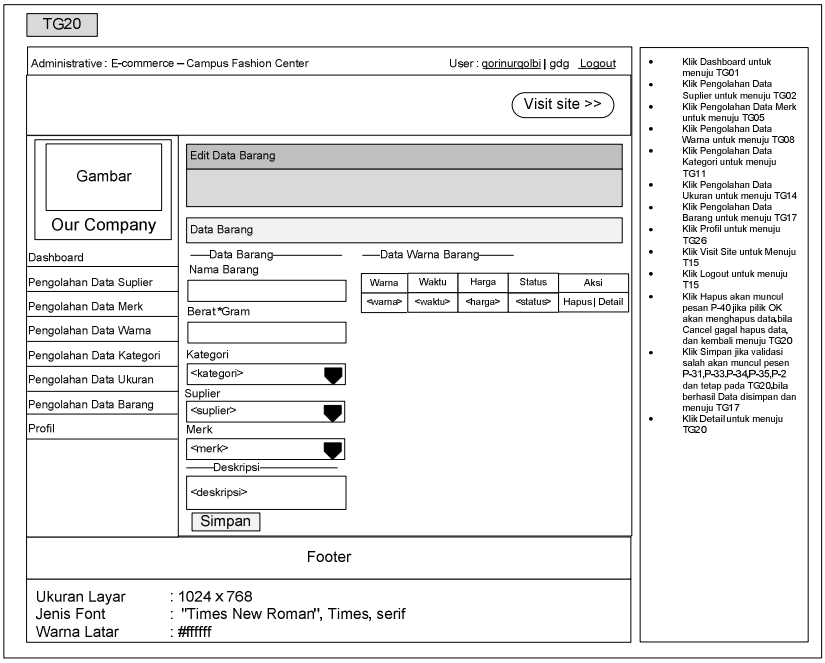
Gambar 3. 120. Perancangan Tampilan Tambah Barang

s. Perancangan Antarmuka Cari Barang



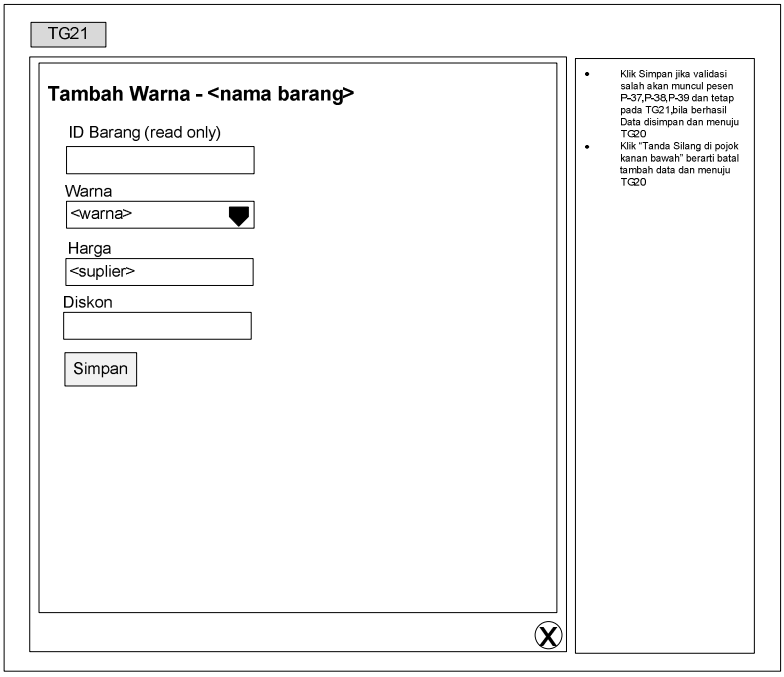
Gambar 3. 121. Perancangan Tampilan Cari Barang

t. Perancangan Antarmuka Barang



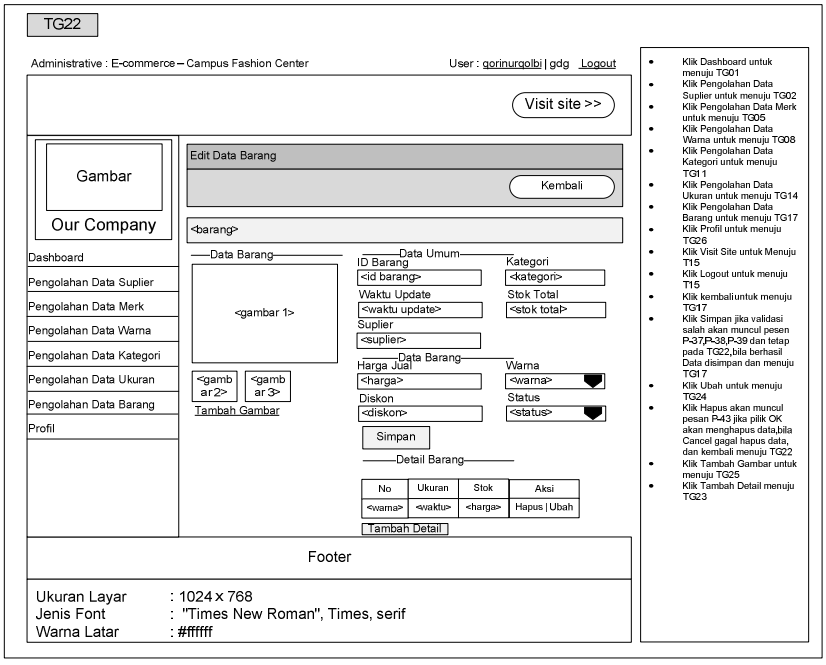
Gambar 3. 122. Perancangan Tampilan Barang Warna

u. Perancangan Antarmuka Tambah Barang Warna



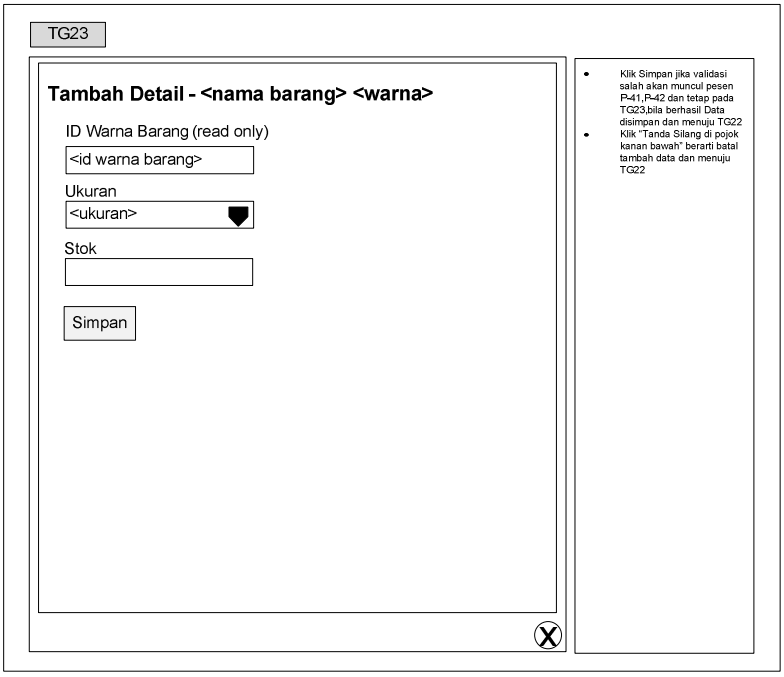
Gambar 3. 123. Perancangan Tampilan Tambah Barang Warna

v. Perancangan Antarmuka Barang Warna



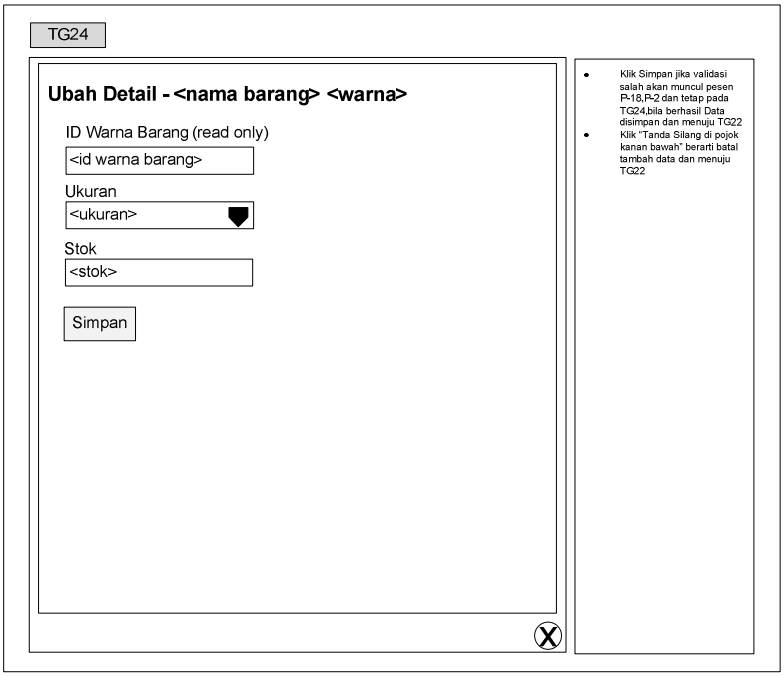
Gambar 3. 124. Perancangan Tampilan Barang Warna

w. Perancangan Antarmuka Tambah Detail Barang



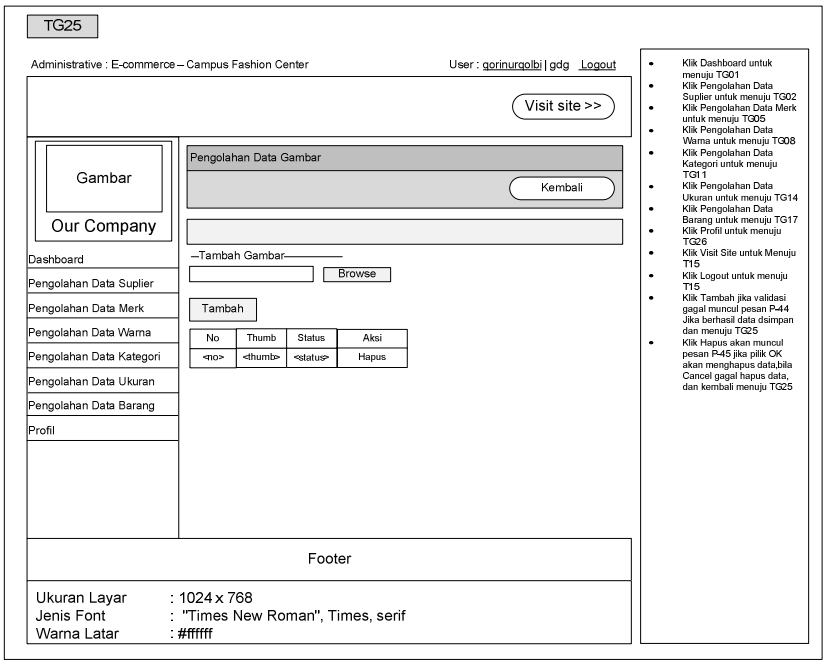
Gambar 3. 125. Perancangan Tampilan Tambah Detail Barang

x. Perancangan Antarmuka Ubah Detail Barang



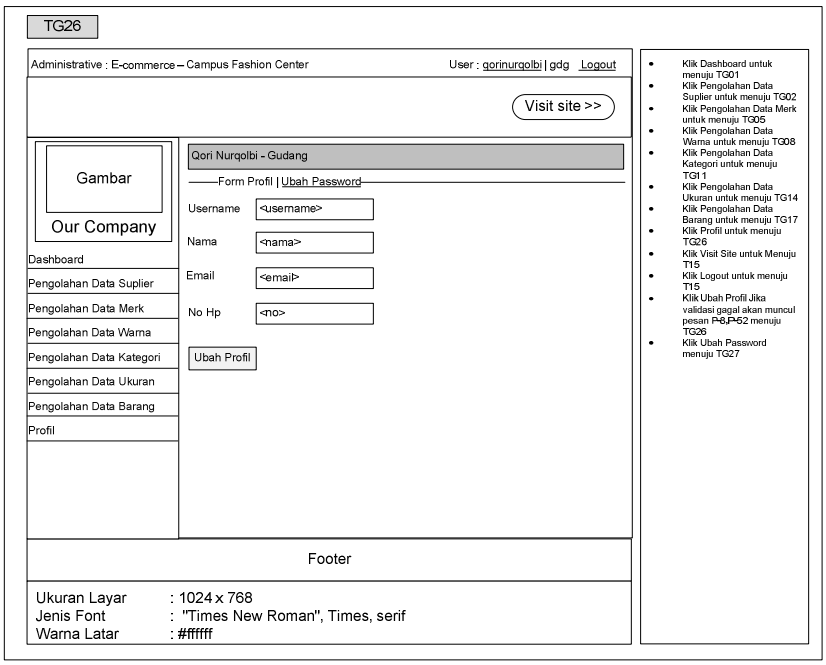
Gambar 3. 126. Perancangan Tampilan Ubah Detail Barang

y. Perancangan Antarmuka Pengolahan Gambar



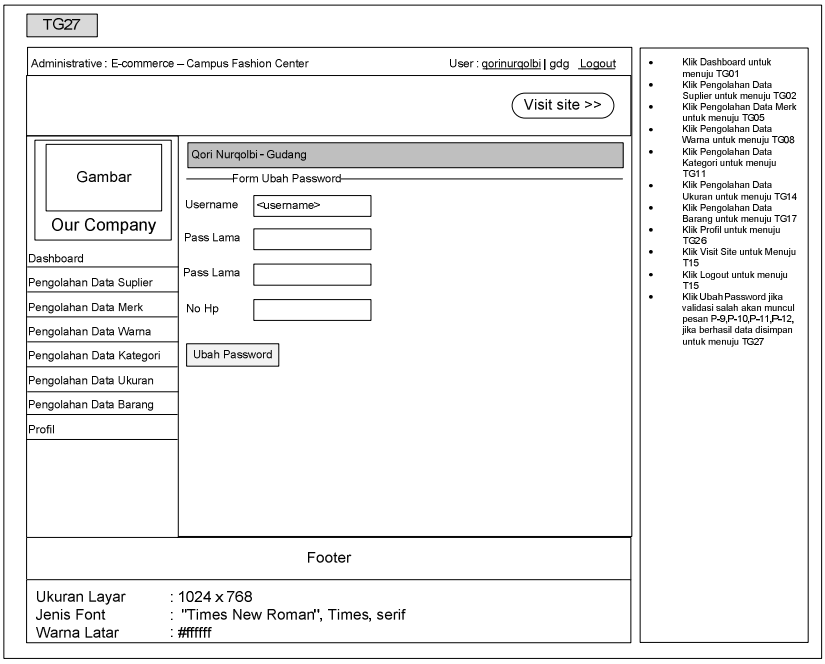
Gambar 3. 127. Perancangan Tampilan Pengolahan Gambar

z. Perancangaa Antarmuka Profil



Gambar 3. 128. Perancangan Ubah Profil

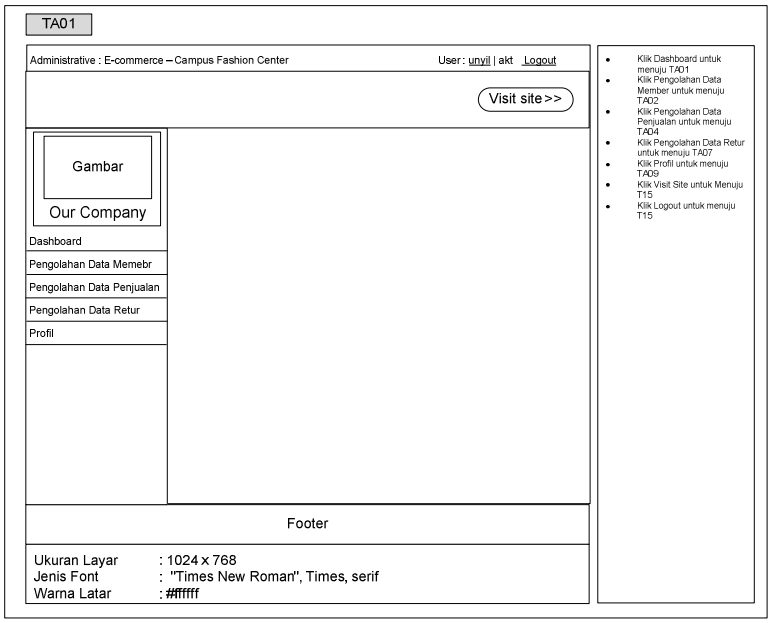
aa. Perancangan Antarmuka Ubah *Password*



Gambar 3. 129. Perancangan Tampilan Ubah *Password*

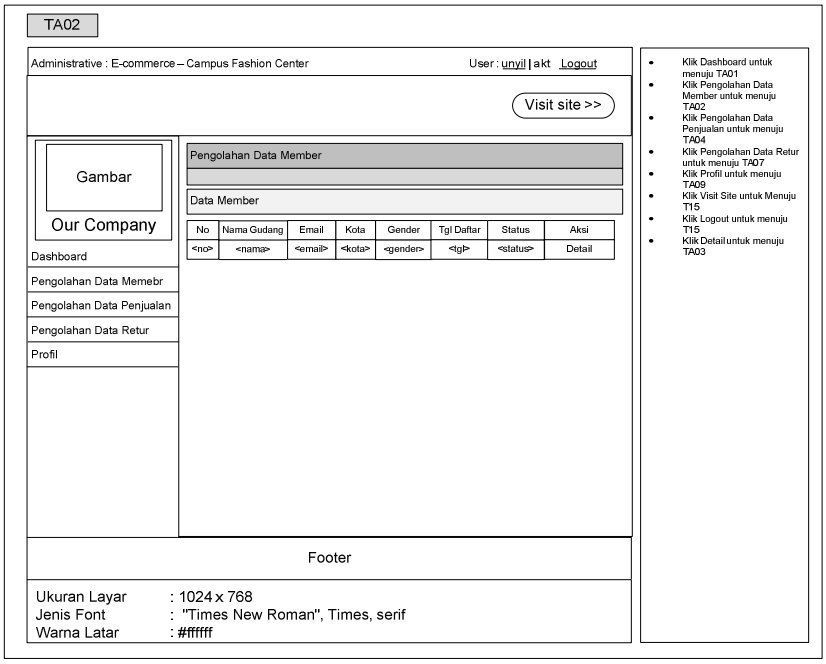
4. Perancangan Antarmuka Akunting

   a. Perancangan Antarmuka Halaman Utama Akunting



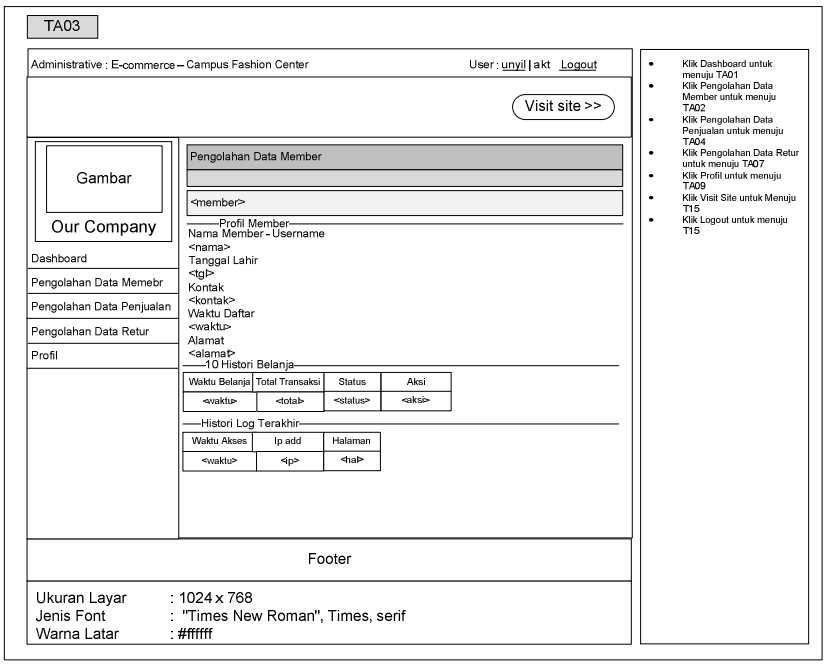
Gambar 3. 130. Perancangan Tampilan Halaman Utama Akunting

b. Perancangan Antarmuka Pengolahan Data Member



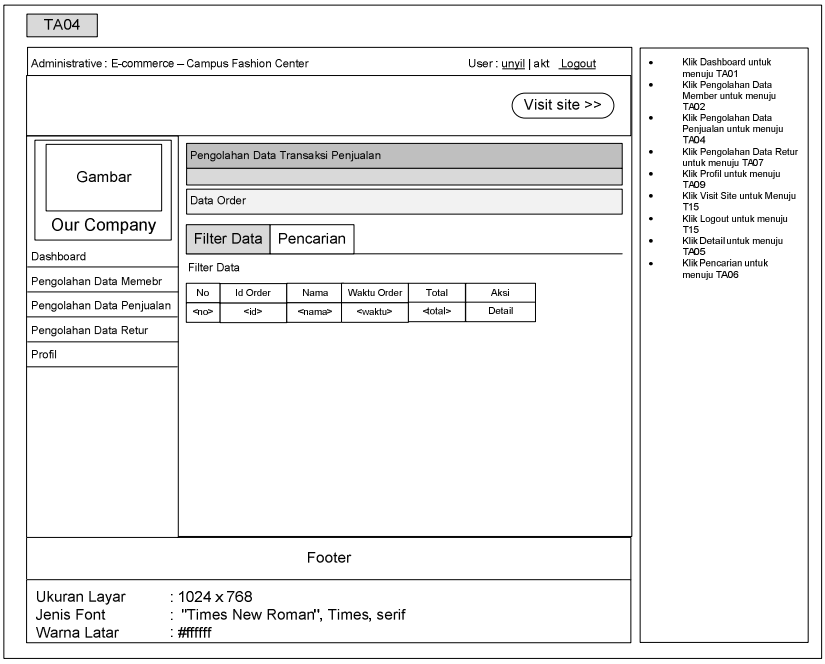
Gambar 3. 131. Perancangan Tampilan Pengolahan Data Member

c. Perancangan Antarmuka Detail Member



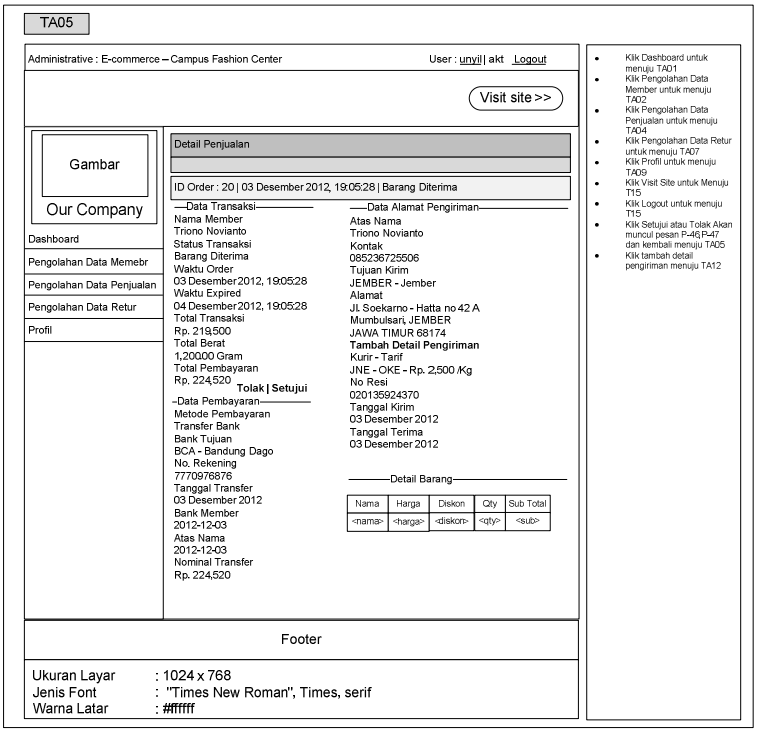
Gambar 3. 132. Perancangan Tampilan Detail Member

d. Perancangan Antarmuka Pengolahan Data Penjualan



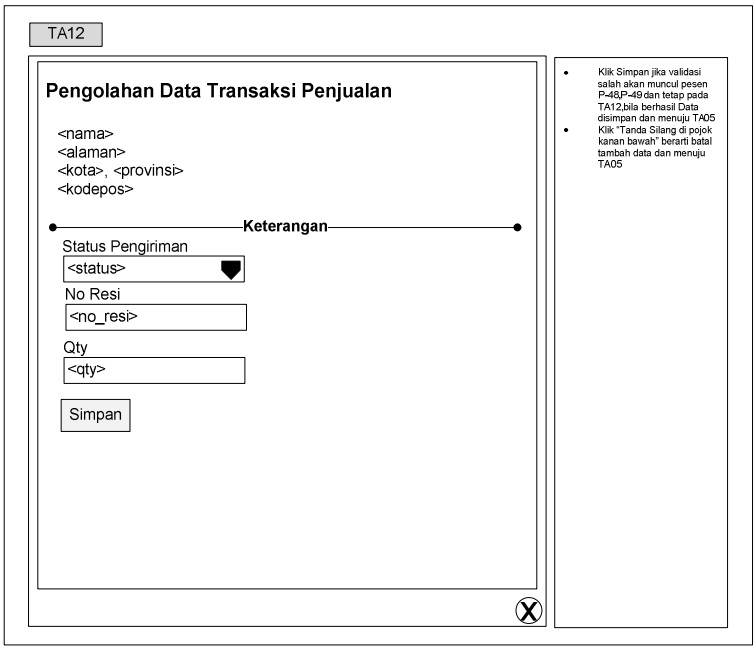
Gambar 3. 133. Perancangan Tampilan Pengolahan Data Penjualan

e. Perancangan Antarmuka Detail Penjualan



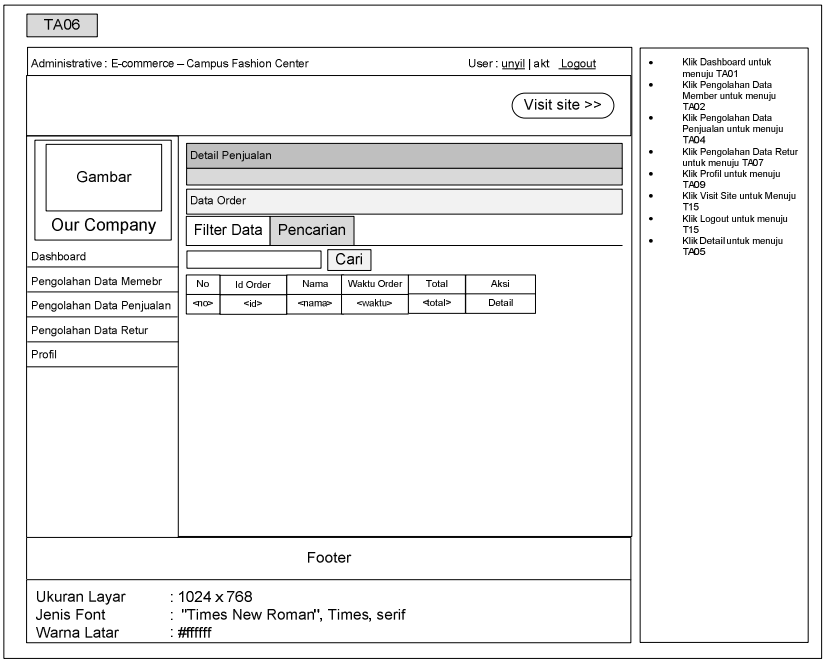
Gambar 3. 134. Perancangan Tampilan Detail Penjualan

f. Perancangan Antarmuka Detail Pengiriman



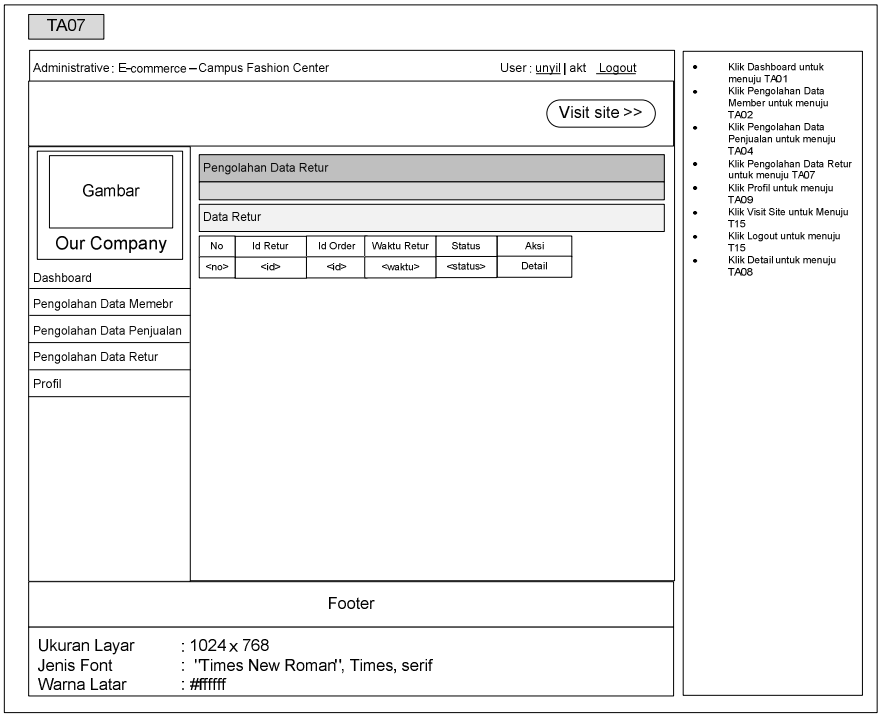
Gambar 3. 135. Perancangan Tampilan Detail Pengiriman

g. Perancangan Antarmuka Cari Penjualan



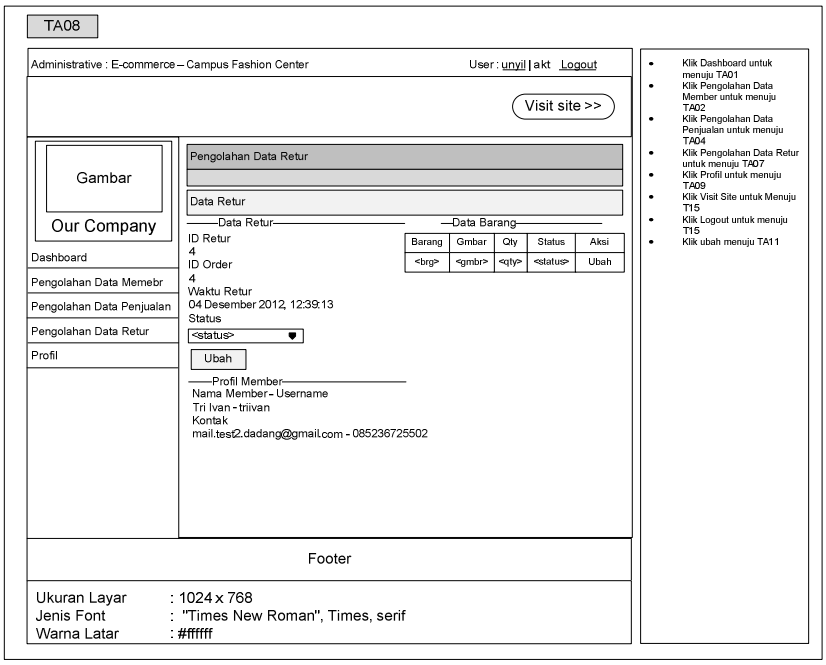
Gambar 3. 136. Perancangan Tampilan Cari Penjualan

h. Perancangan Antarmuka Pengolahan Data Retur



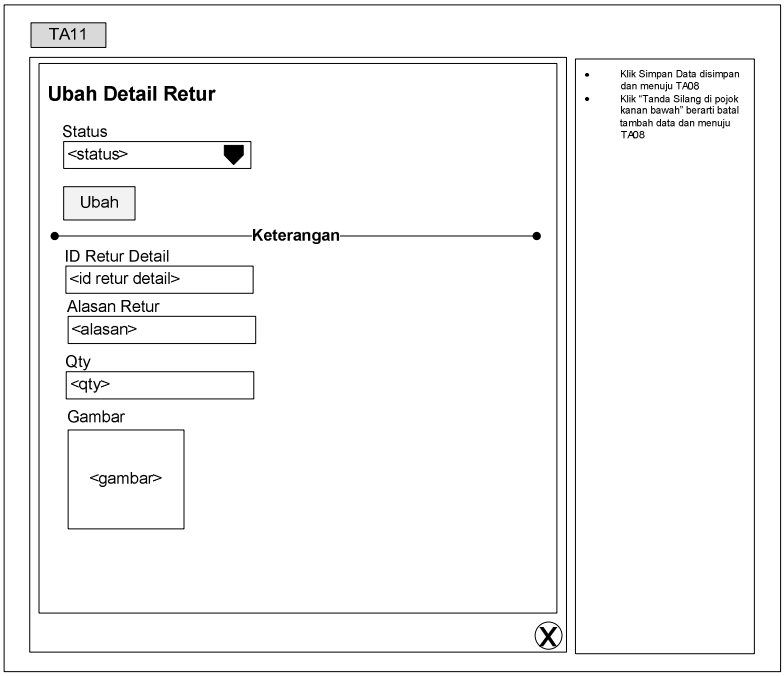
Gambar 3. 137. Perancangan Tampilan Pengolahan Data Retur

i. Perancangan Antarmuka Detail Retur



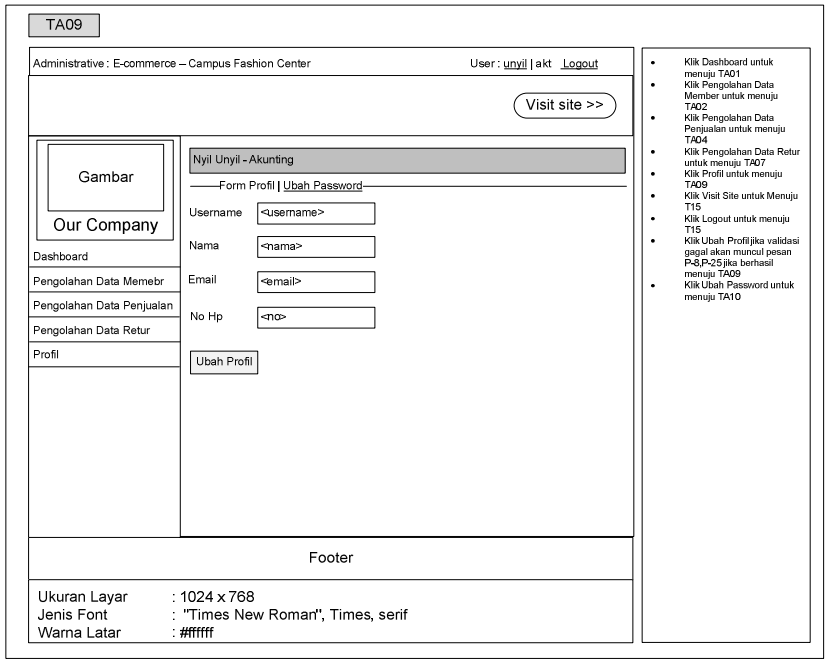
Gambar 3. 138. Perancangan Tampilan Detail Retur

j. Perancangan Antarmuka Ubah Detail Retur



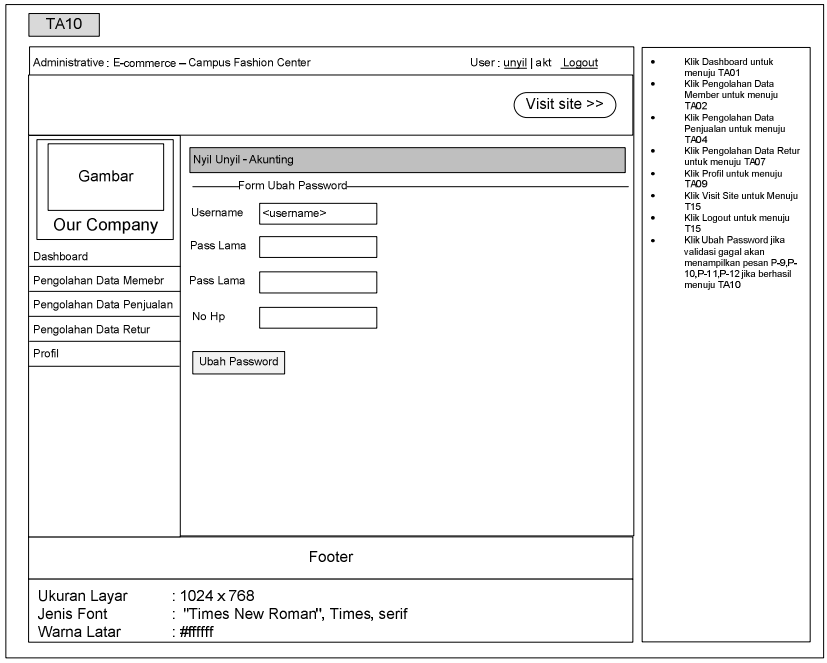
Gambar 3. 139. Perancangan Tampilan Ubah Detail Retur

k. Perancangan Antarmuka Profil



Gambar 3. 140. Perancangan Tampilan Profil

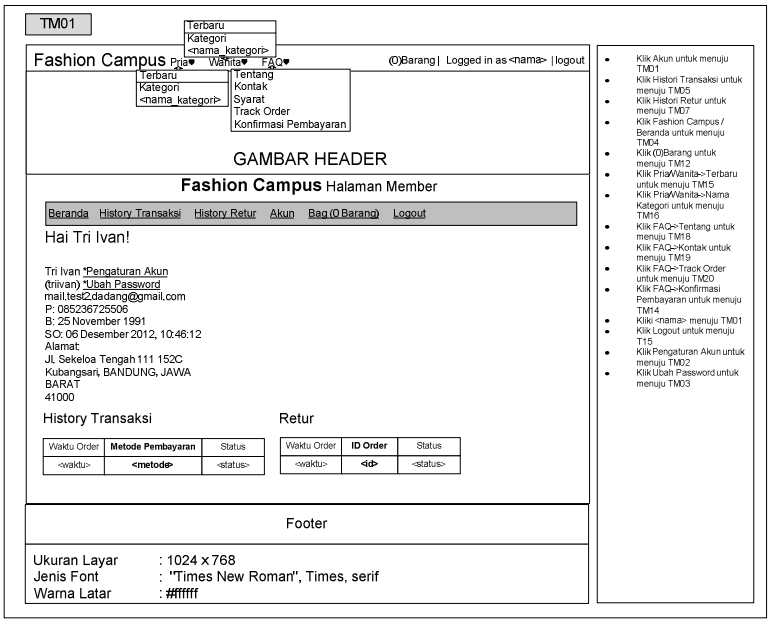
l. Perancangan Antarmuka Ubah *Password*



Gambar 3. 141. Perancangan Tampilan Ubah *Password*

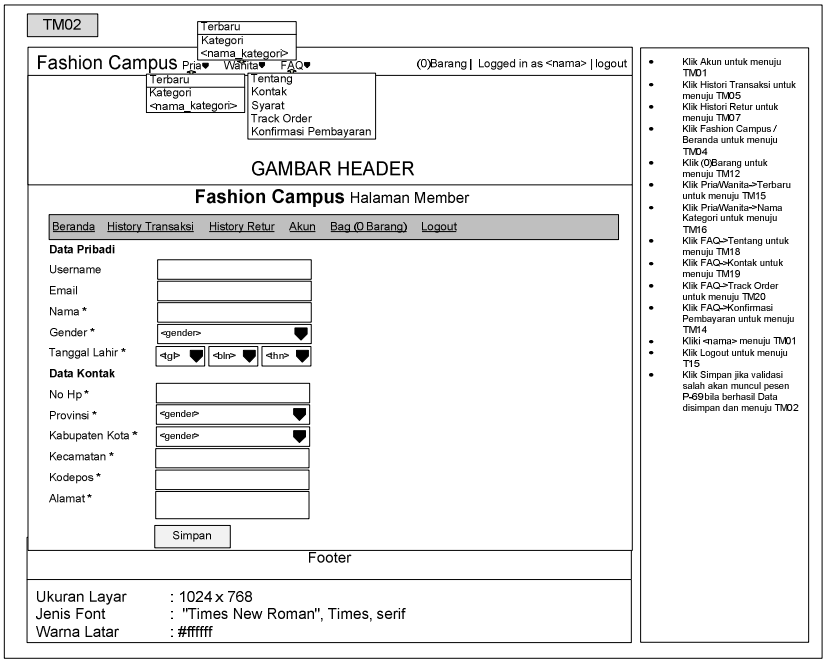
5. Perancangan Antarmuka *Member*

a. Perancangan Antarmuka Halaman *Member*



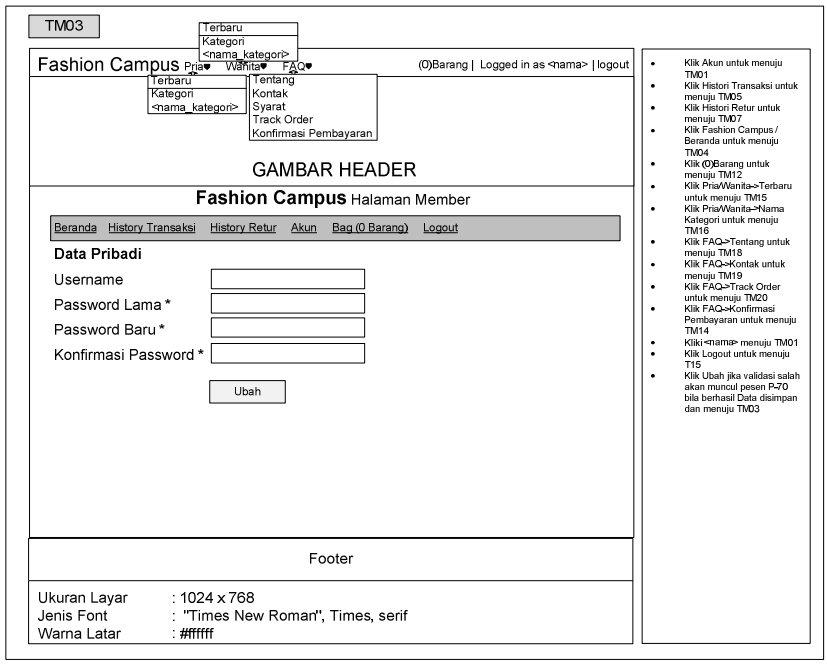
Gambar 3. 142. Perancangan Tampilan Halaman *Member*

b. Perancangan Antarmuka Pengaturan Akun



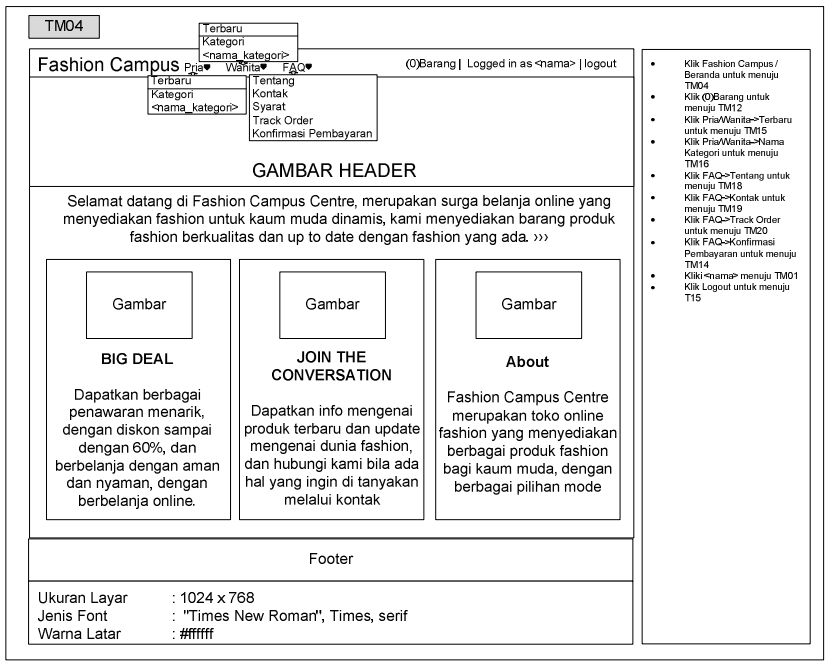
Gambar 3. 143. Perancangan Tampilan Pengaturan Akun

c. Perancangan Antarmuka Ubah *Password*



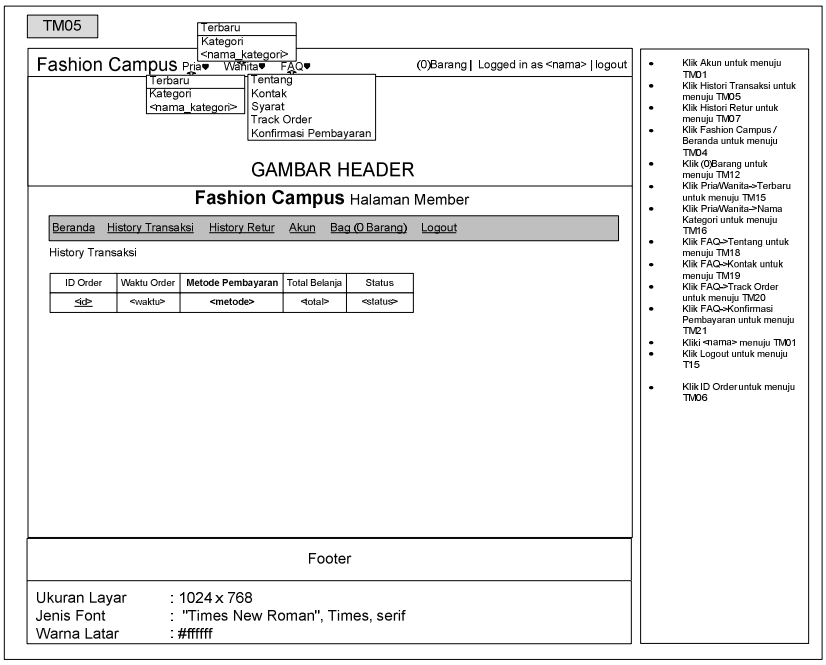
Gambar 3. 144. Perancangan Tampilan Ubah *Password*

d. Perancangan Antarmuka Halaman Beranda



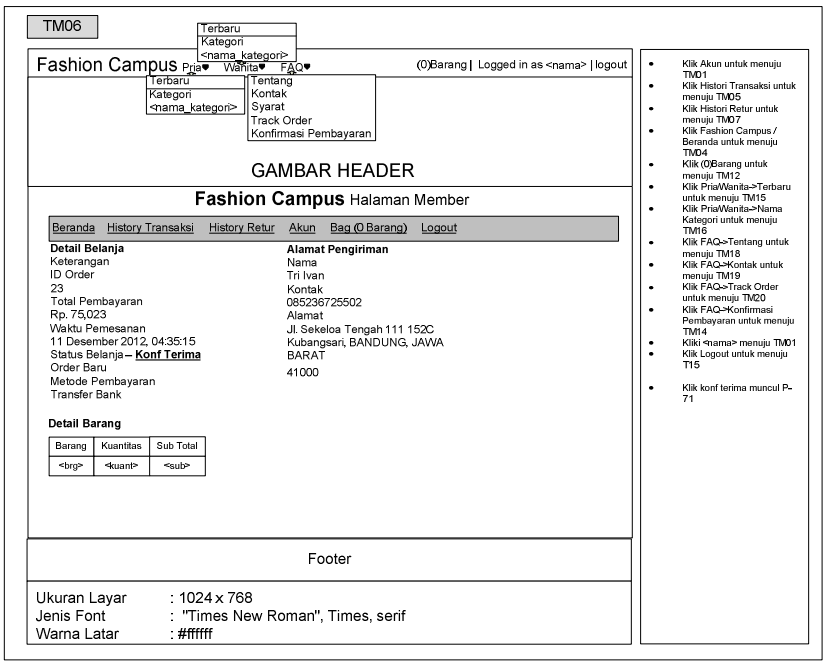
Gambar 3. 145. Perancangan Tampilan Halaman Beranda

e. Perancangan Antarmuka *History* Transaksi



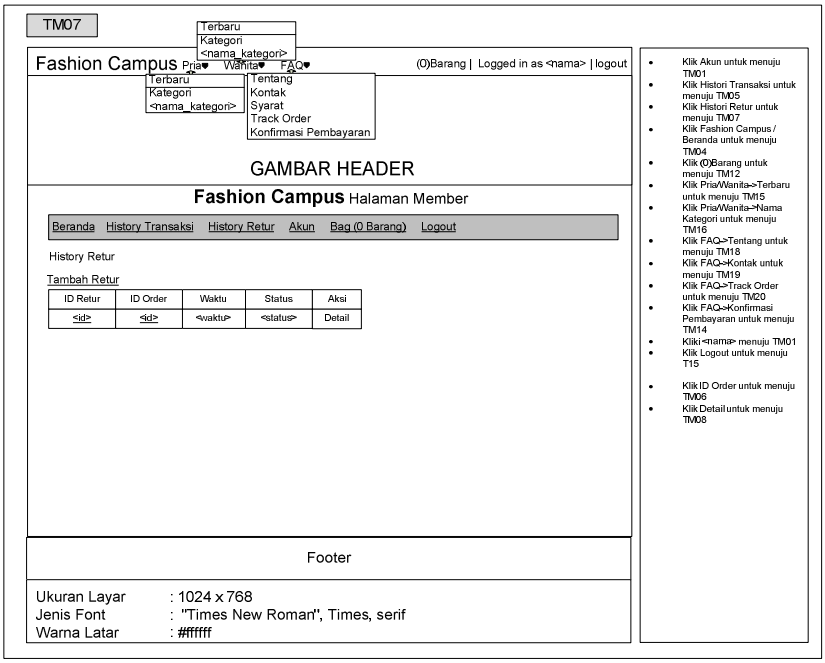
Gambar 3. 146. Perancangan Tampilan *History* Transaksi

f. Perancangan Antarmuka Detail Transaksi



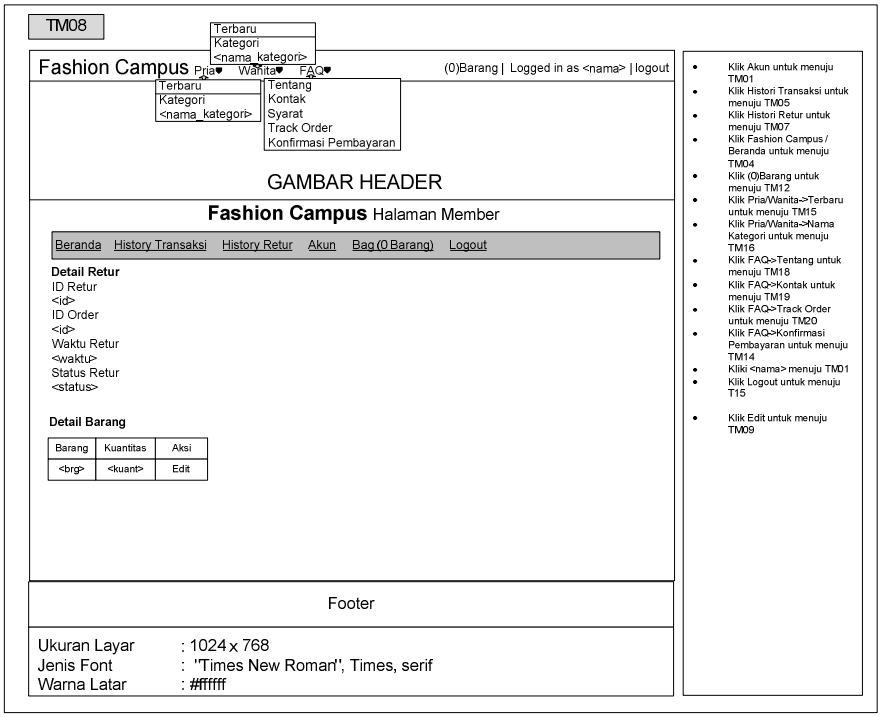
Gambar 3. 147. Perancangan Tampilan Detail Transaksi

g. Perancangan Antarmuka *History* Retur



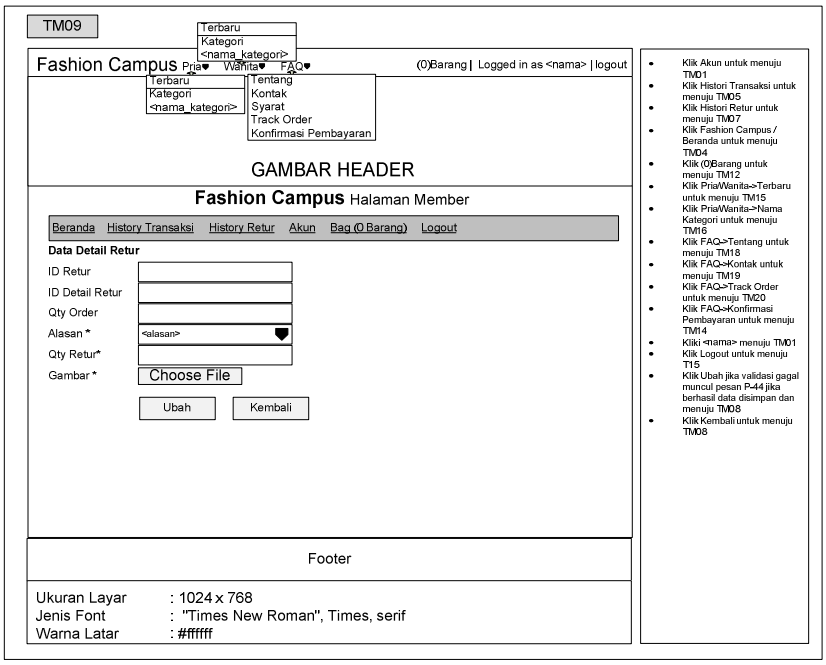
Gambar 3. 148. Perancangan Tampilan *History* Retur

h. Perancangan Antarmuka Detail Retur



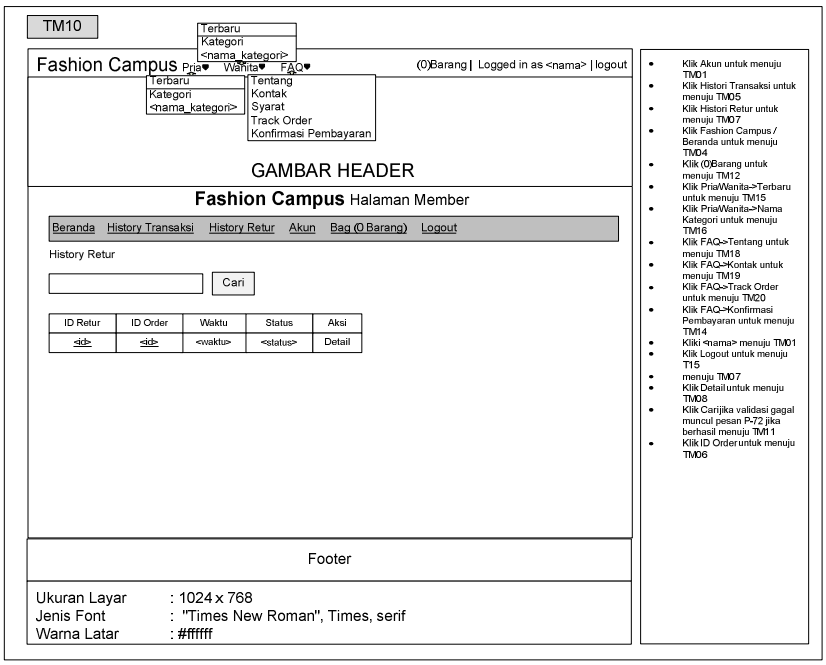
Gambar 3. 149. Perancangan Tampilan Detail Retur

i. Perancangan Antarmuka Edit Retur



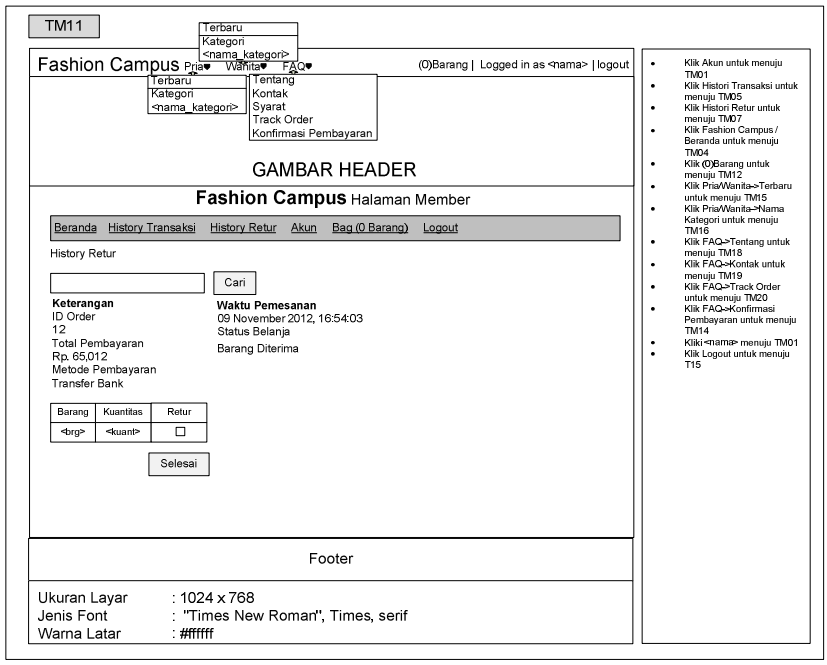
Gambar 3. 150. Perancangan Tampilan Ubah Retur

j. Perancangan Antarmuka Cari Retur



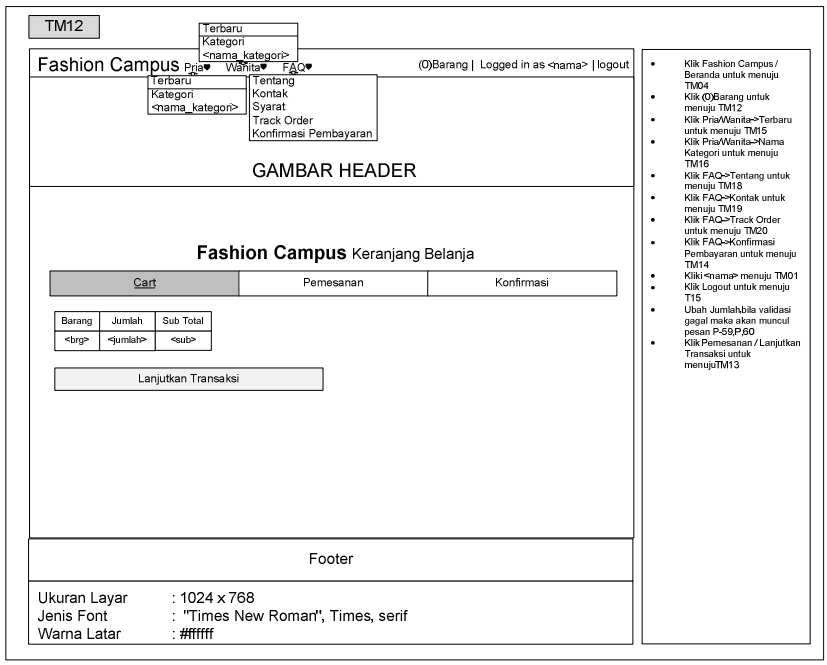
Gambar 3. 151. Perancangan Tampilan Cari Retur

k. Perancangan Antarmuka Tambah Retur



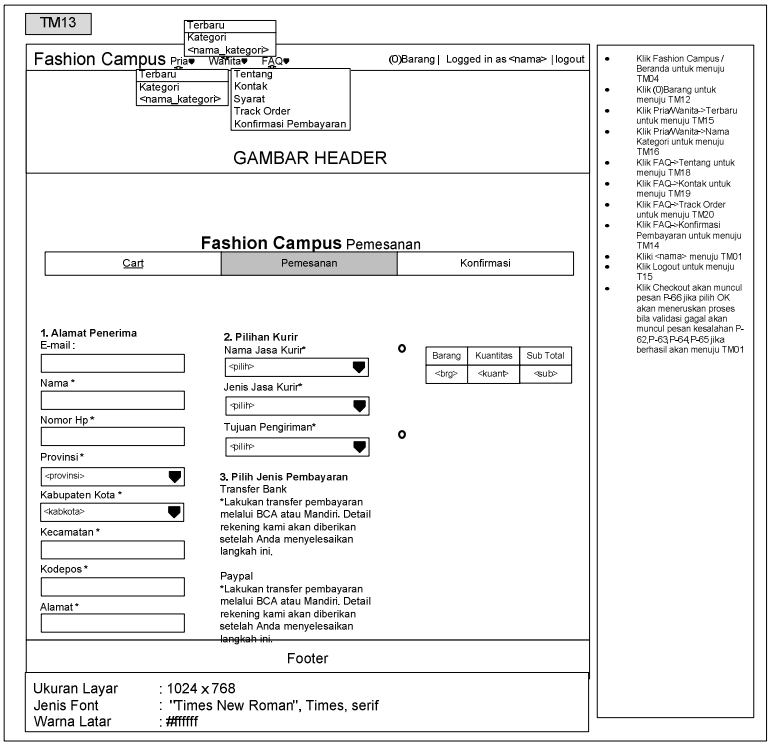
Gambar 3. 152. Perancangan Tampilan Tambah Retur

l. Perancangan Antarmuka Keranjang Belanja



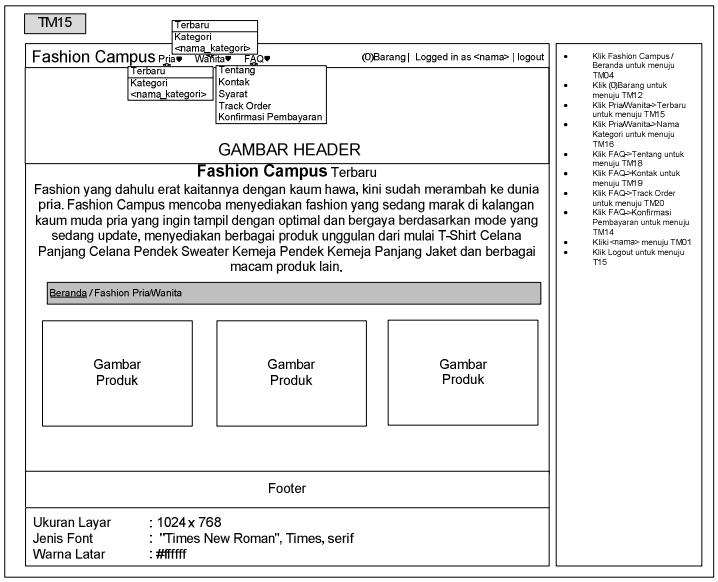
Gambar 3. 153. Perancangan Tampilan Keranjang Belanja

m. Perancangan Antarmuka Pemesanan



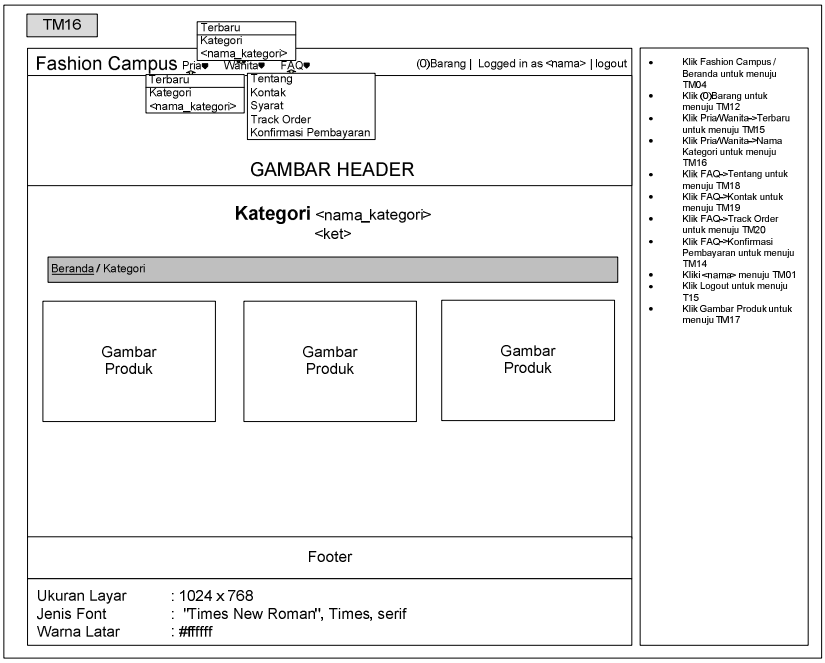
Gambar 3. 154. Perancangan Tampilan Pemesanan

n. Perancangan Antarmuka Barang Terbaru



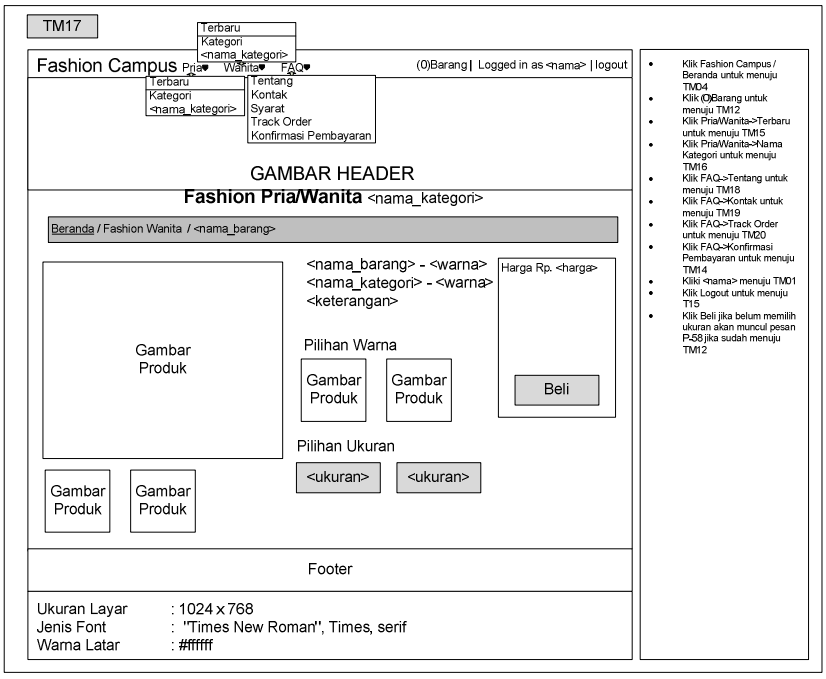
Gambar 3. 155. Perancangan Tampilan Barang Terbaru

o. Perancangan Antarmuka Kategori



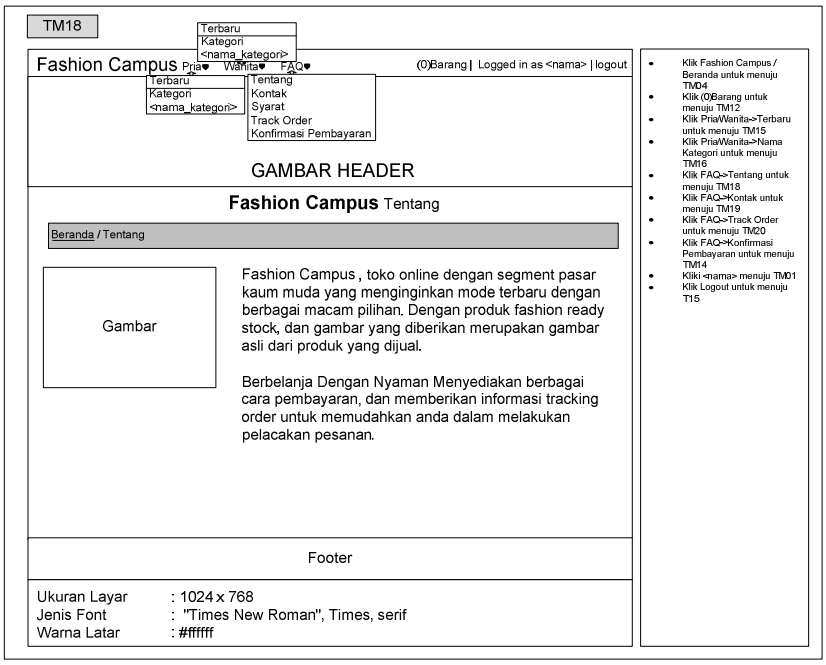
Gambar 3. 156. Perancangan Tampilan Kategori

p. Perancangan Antarmuka Datail Barang



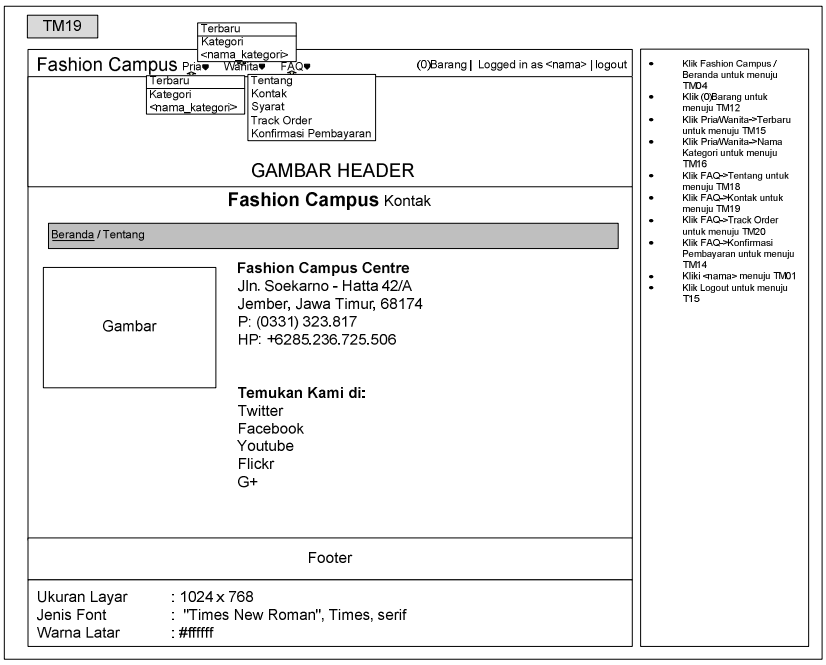
Gambar 3. 157. Perancangan Tampilan Detail Barang

q. Perancangan Antarmuka Tentang



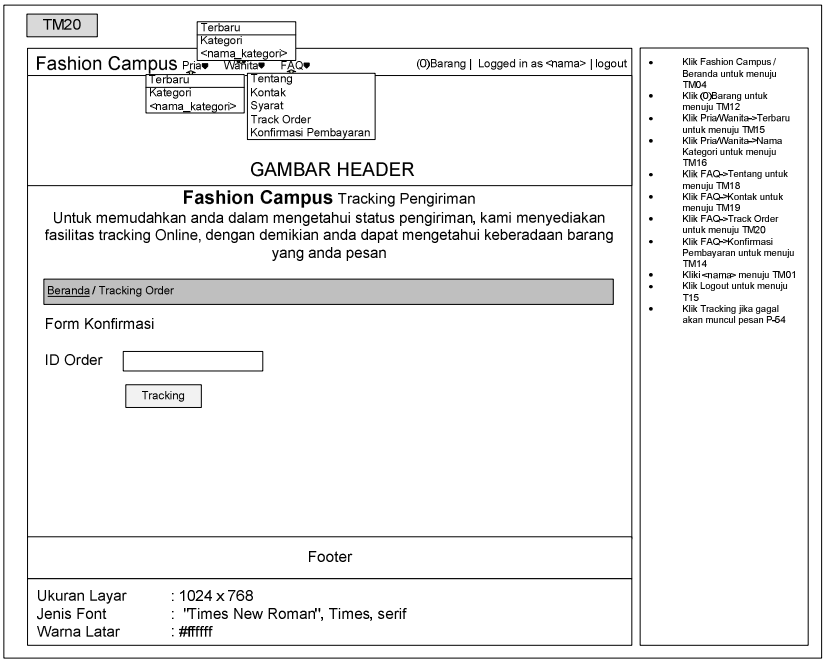
Gambar 3. 158. Perancangan Tampilan Tentang

r. Perancangan Antarmuka Kontak



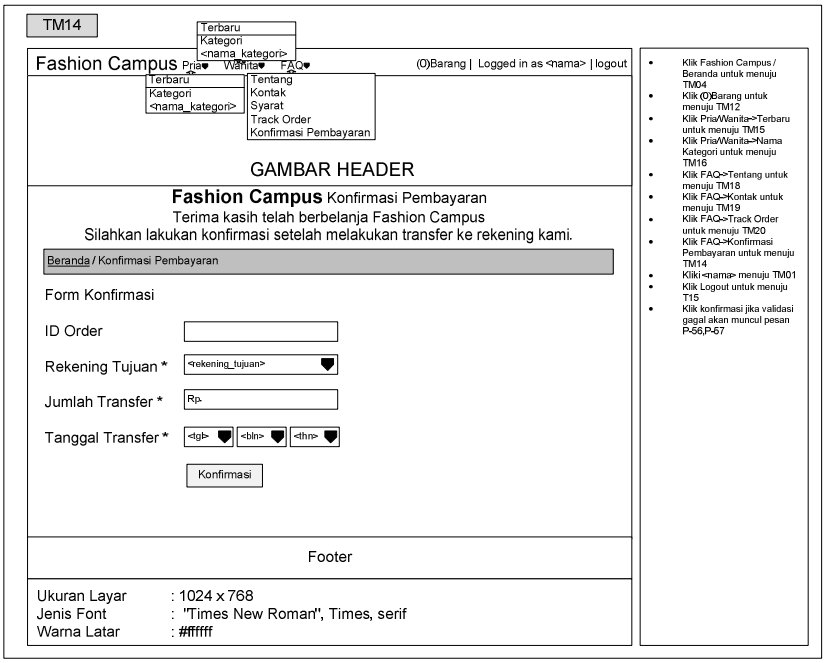
Gambar 3. 159. Perancangan Tampilan Kontak

s. Perancangan Antarmuka *Track* Order



Gambar 3. 160. Perancangan Tampilan *Track* Order

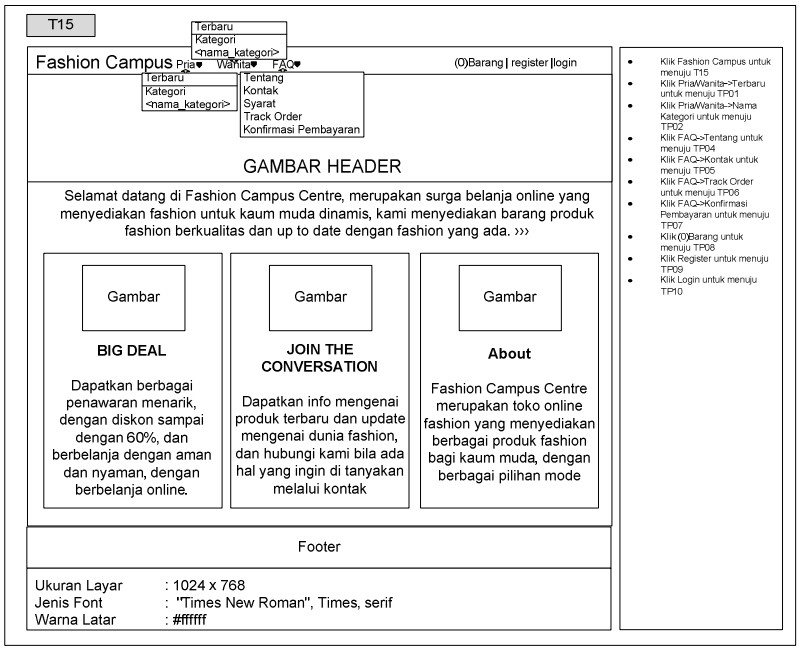
t. Perancangan Antarmuka Konfirmasi Pembayaran



Gambar 3. 161. Perancangan Tampilan Konfirmasi Pembayaran

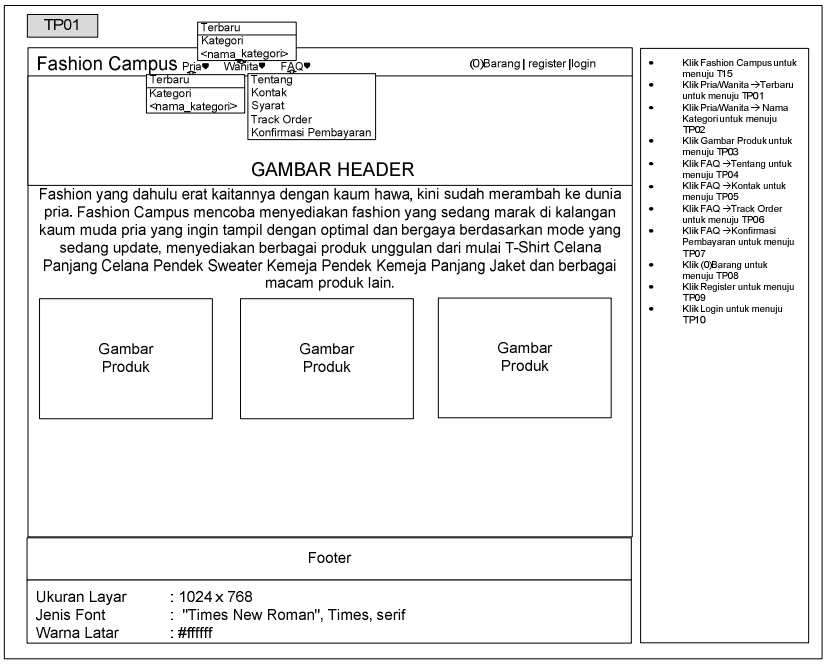
6. Perancangan Antarmuka Pengunjung

a. Perancangan Antarmuka Halaman Utama



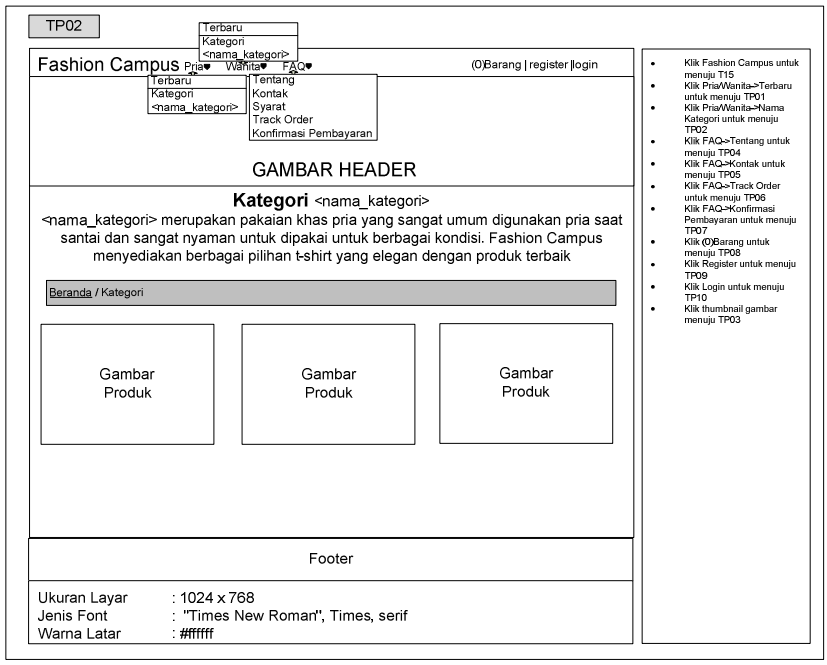
Gambar 3. 162. Perancangan Tampilan Halaman Utama

b. Perancangan Antarmuka Terbaru



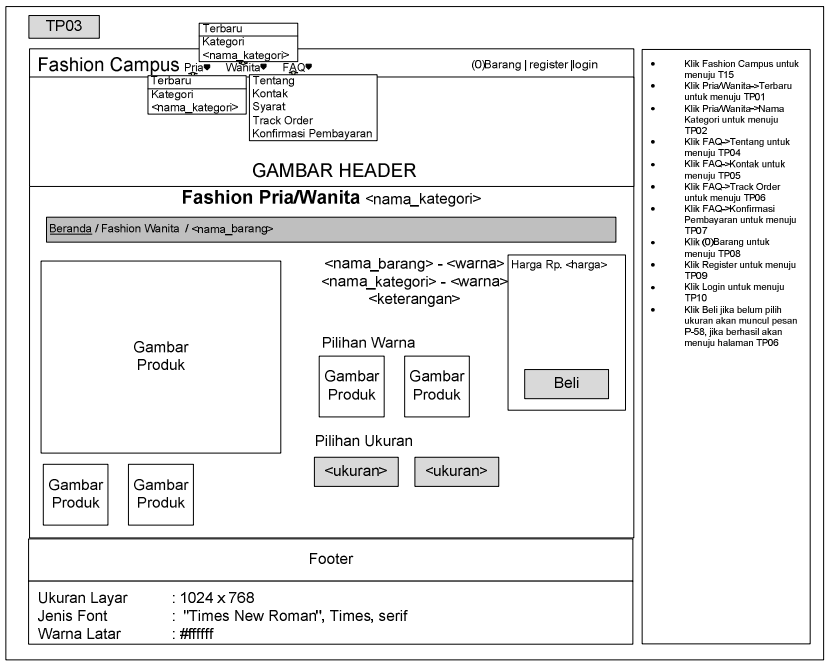
Gambar 3. 163. Perancangan Tampilan Barang Terbaru

c. Perancangan Antarmuka Kategori



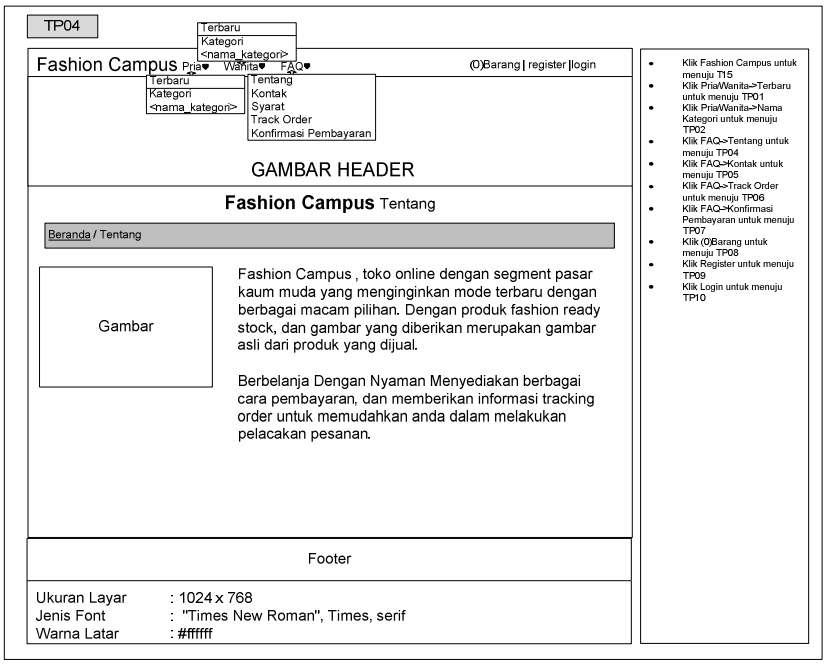
Gambar 3. 164. Perancangan Tampilan Kategori

d. Perancangan Antarmuka Detail Barang



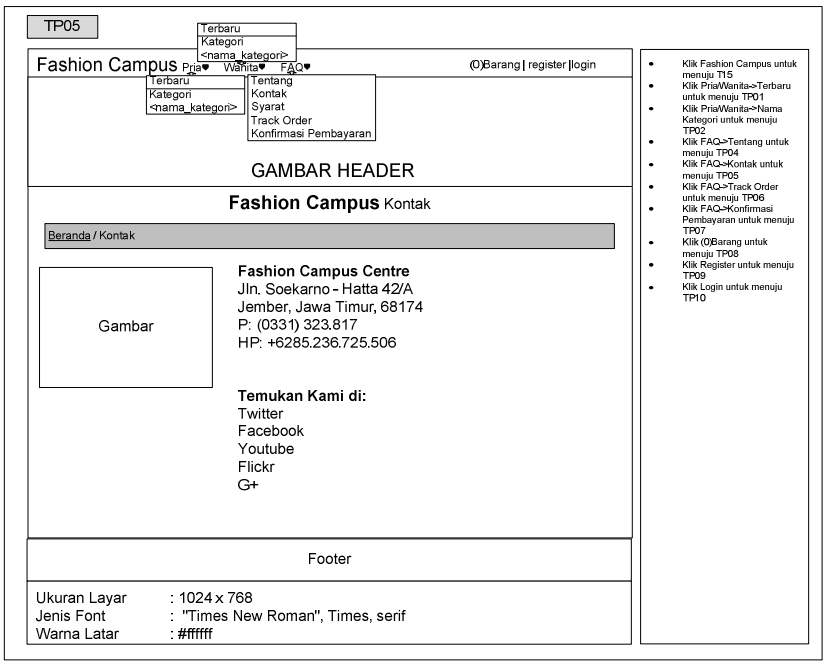
Gambar 3. 165. Perancangan Tampilan Detail Barang

e. Perancangan Antarmuka Tentang



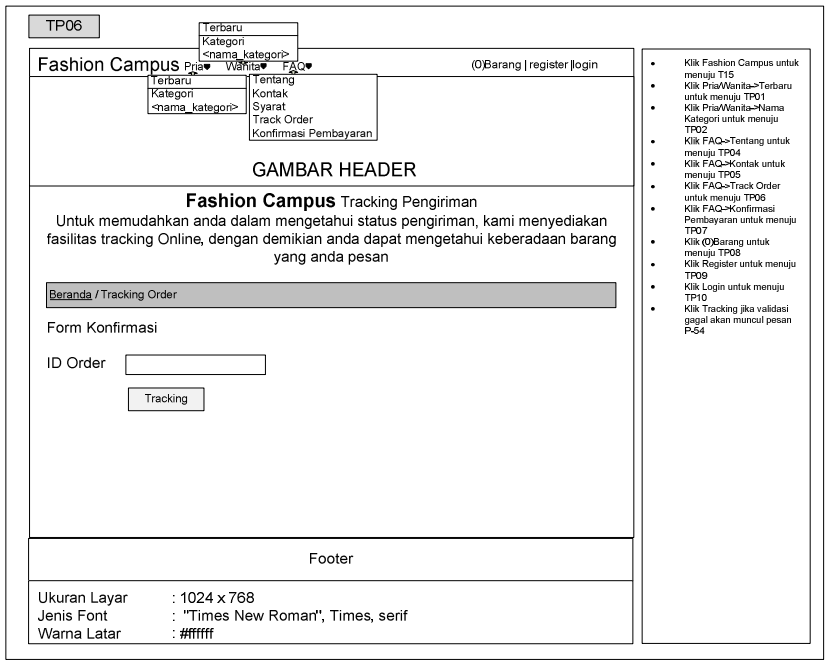
Gambar 3. 166. Perancangan Tampilan Tentang

f. Perancangan Antarmuka Kontak



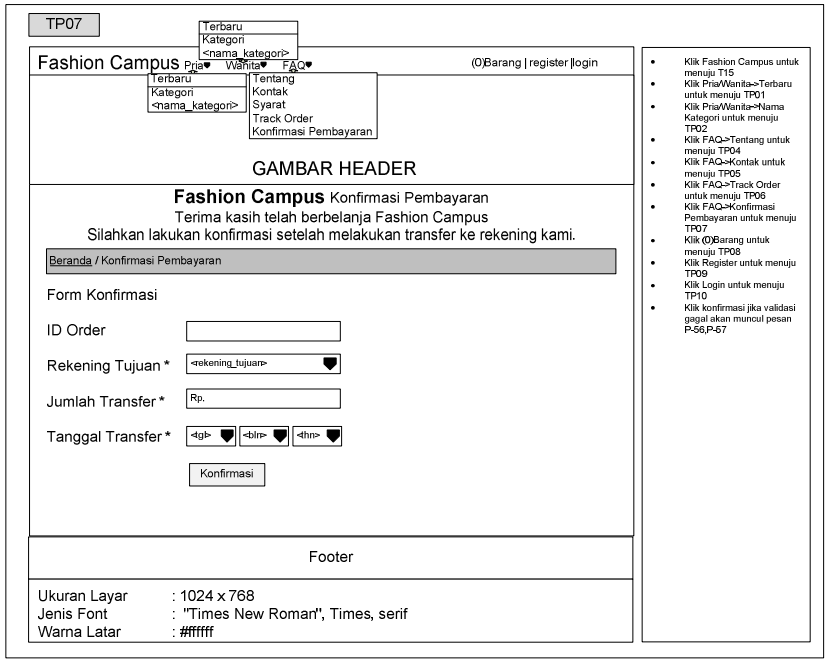
Gambar 3. 167. Perancangan Tampilan Kontak

g. Perancangan Antarmuka *Track* Order



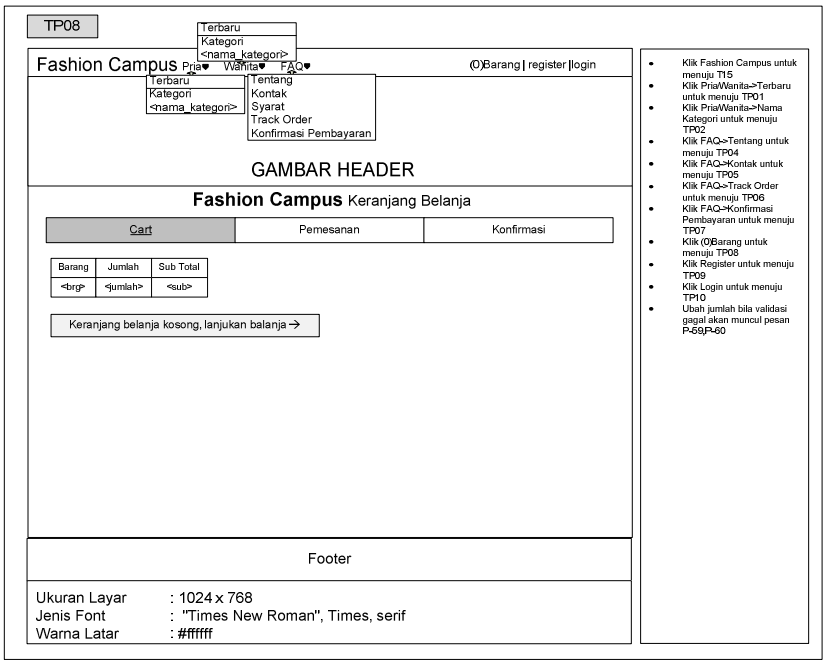
Gambar 3. 168. Perancangan Tampilan *Track* Order

h. Perancangan Antarmuka Konfirmasi Pembayaran



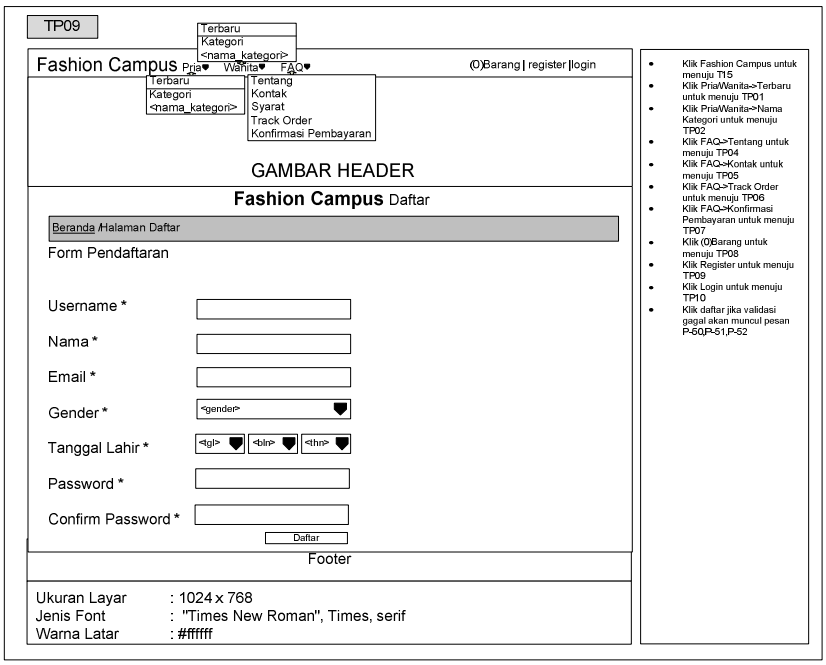
Gambar 3. 169. Perancangan Tampilan Konfirmasi Bayar

i. Perancangan Antarmuka Keranjang Belanja



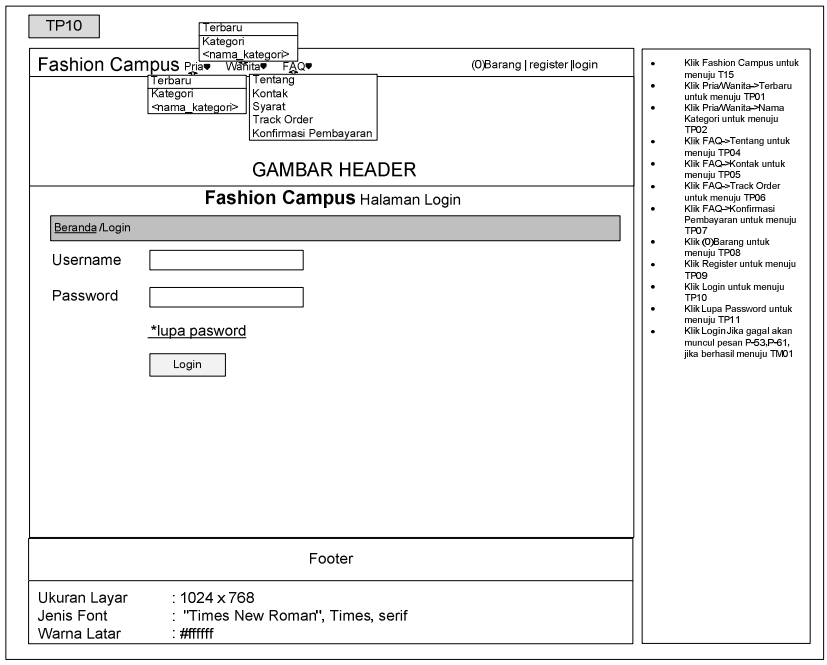
Gambar 3. 170. Perancangan Tampilan Keranjang Belanja

j. Perancangan Antarmuka Registrasi



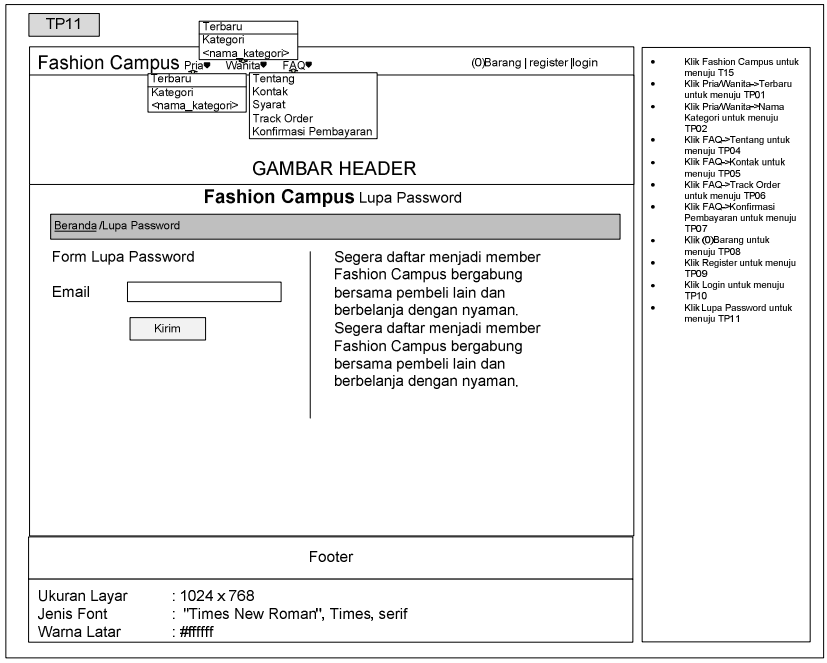
Gambar 3. 171. Perancangan Tampilan Registrasi

k. Perancangan Antarmuka *Login*



Gambar 3. 172. Perancangan Tampilan *Login*

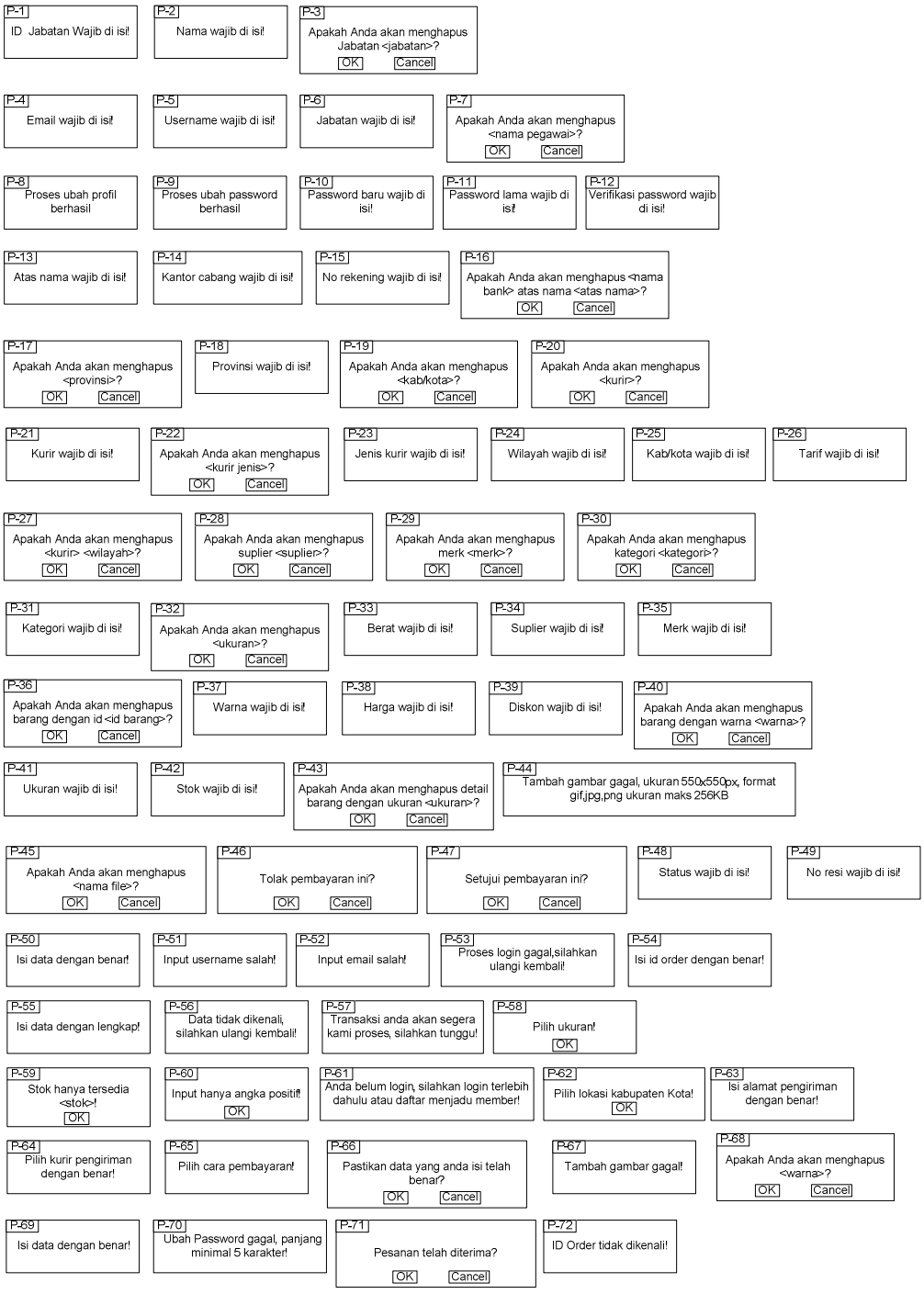
l. Perancangan Antarmuka Lupa *Password*



Gambar 3. 173. Perancangan Tampilan Lupa *Password*

3.2.3.2 Perancangan Pesan

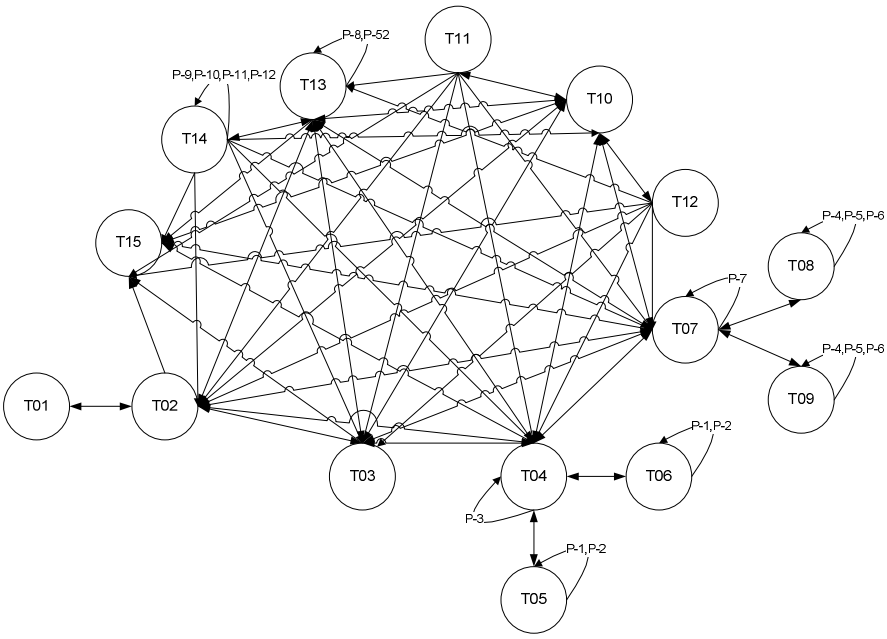
Perancangan pesan digunakan untuk memudahkan pengguna untuk menggunakan aplikasi dan membertahu bila terdapat kesalahan. Berikut rancangan pesan kesalahan untuk Campus Fashion Centre.



Gambar 3. 174. Perancangan Tampilan Pesan

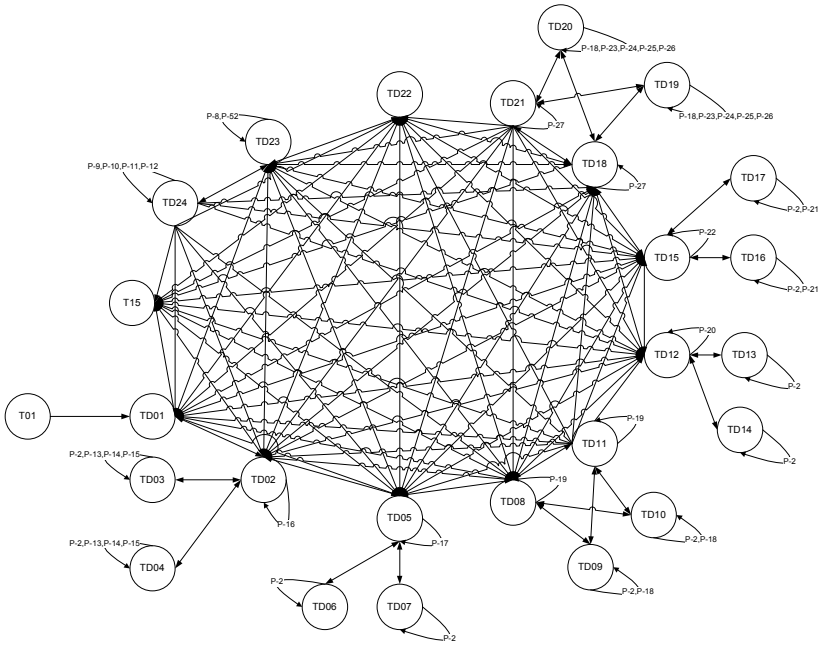
3.2.4 Jaringan Semantik

3.2.4.1.Jaringan Semantik Pimpinan



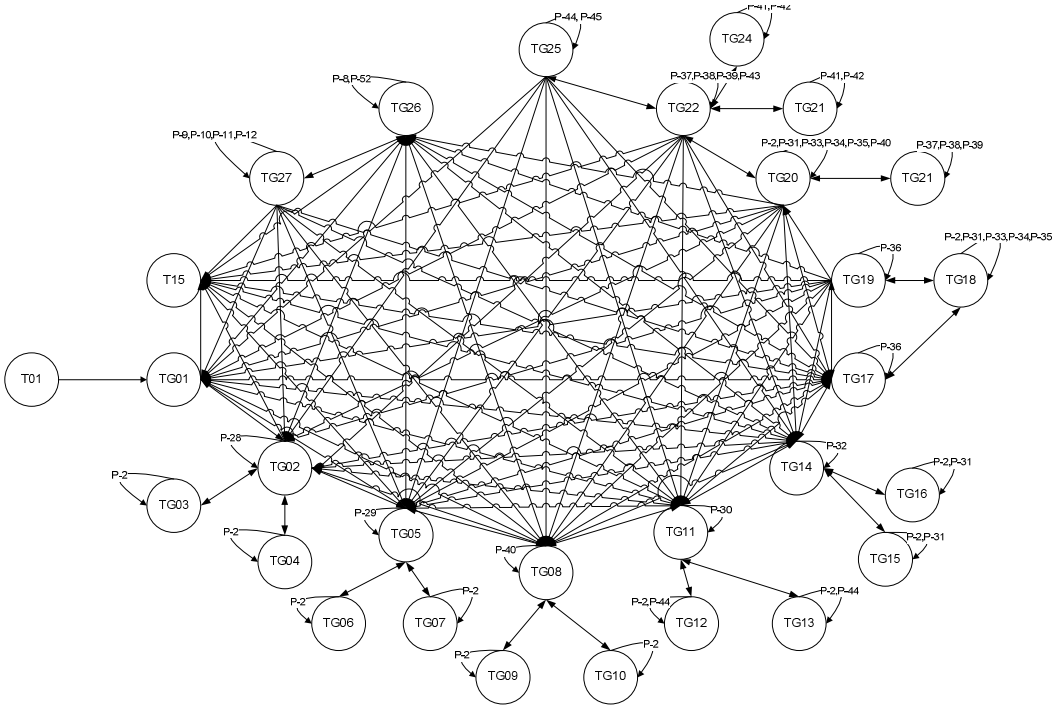
Gambar 3. 175. Jaringan Semantik Pimpinan

3.2.4.2.Jaringan Semantik *Database* Administrator



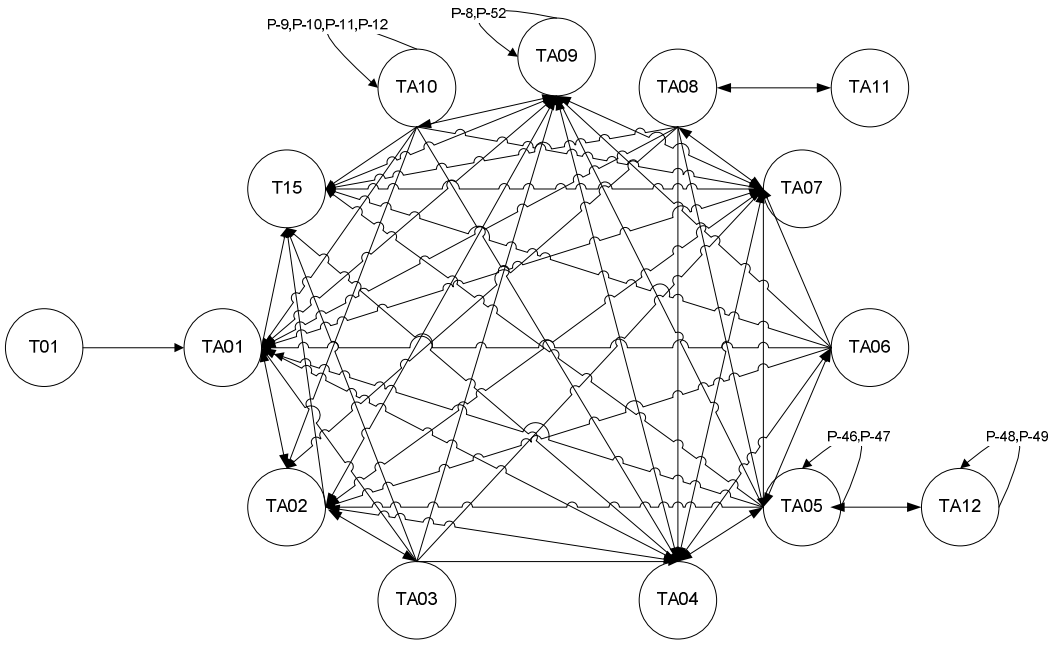
Gambar 3. 176. Perancangan Semantik *Database* Administrator

3.2.4.3.Jaringan Semantik Gudang



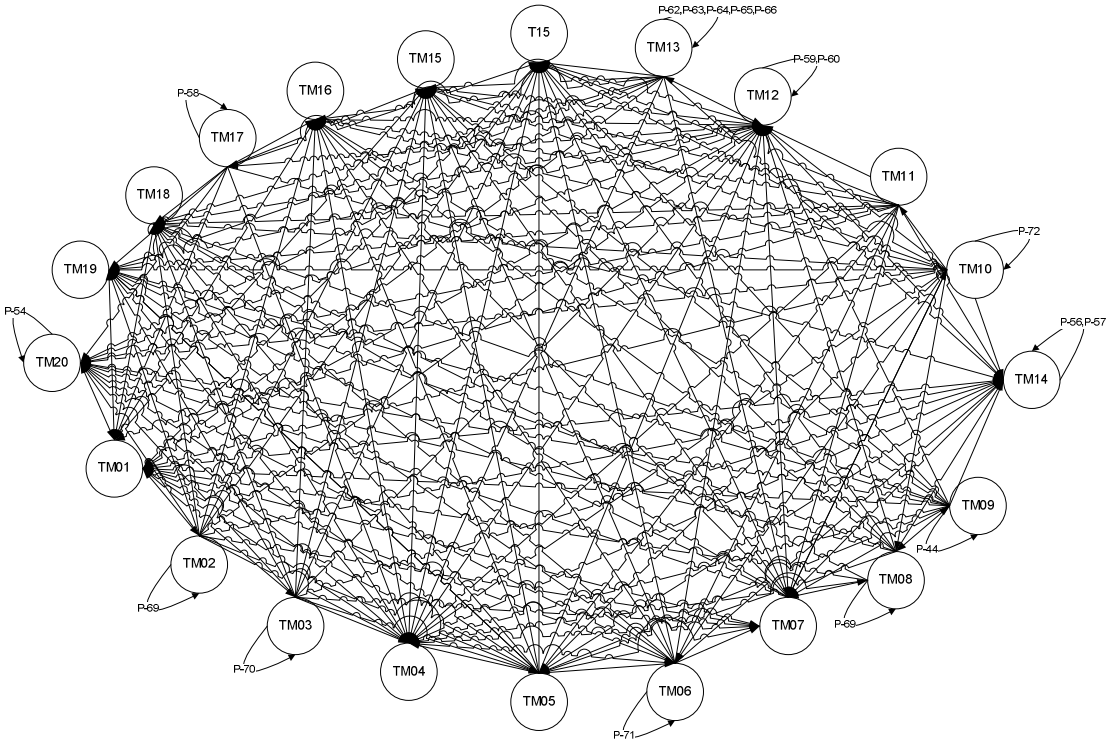
Gambar 3. 177. Perancangan Semantik Gudang

3.2.4.4.Jaringan Semantik Akunting



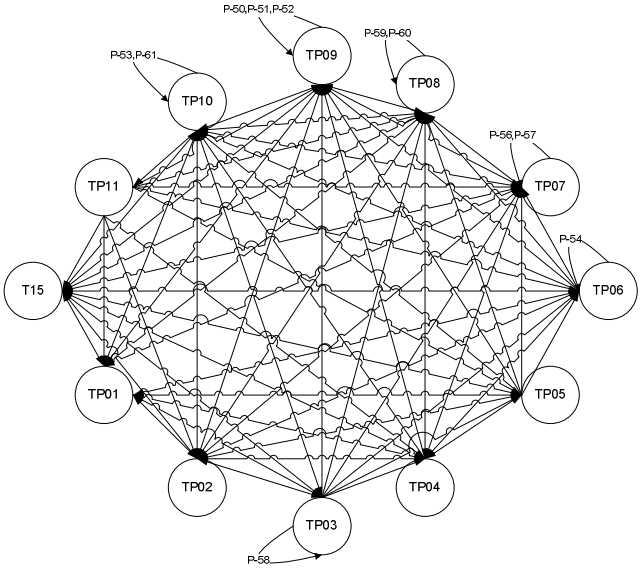
Gambar 3. 178. Perancangan Semantik Akunting

3.2.4.5.Jaringan Semantik *Member*



Gambar 3. 179. Perancangan Semantik *Member*

3.2.4.6.Jaringan Semantik Pengunjung



Gambar 3. 180. Perancangan Semanting Pengunjung

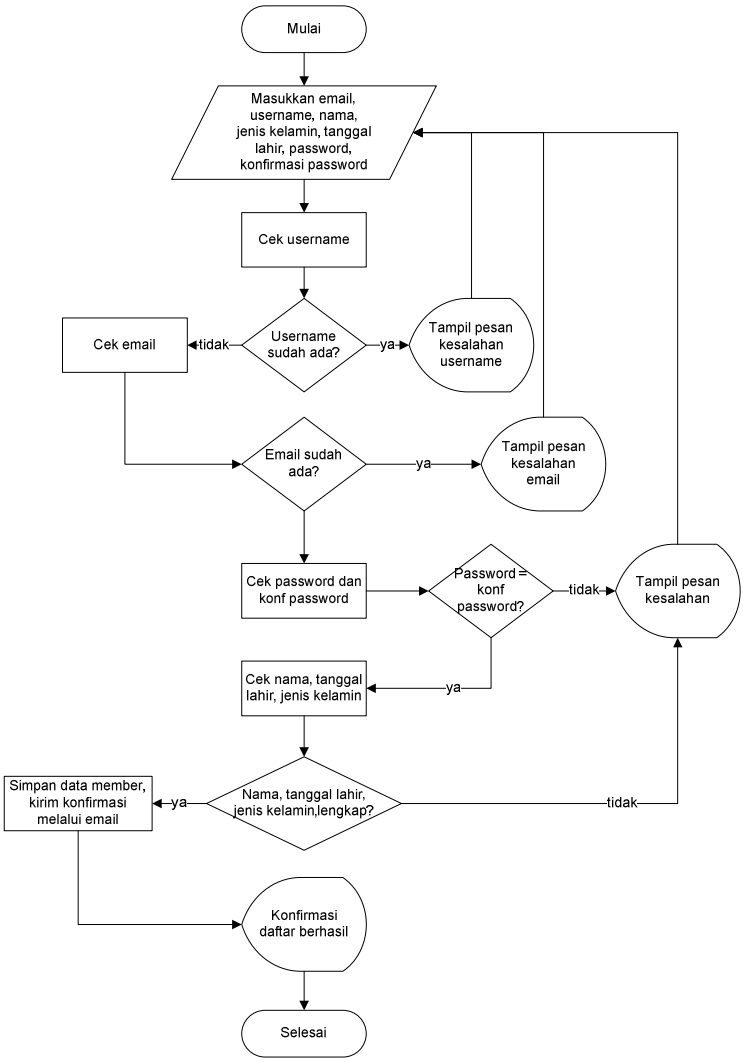
3.2.5 Perancangan Prosedural

Perancangan prosedural adalah tata cara atau urutan langkah-langkah untuk melakukan sebuah proses. Prosedural akan digunakan sebagai algoritma dasar dalam pengkodean prosedur yang ada. *Tools* yang digunakan adalah diagram flowchart.

Adapun perancangan prosedural Campus Fashion Centre adalah sebagai berikut:

3.2.5.1.Perancangan Prosedural Daftar

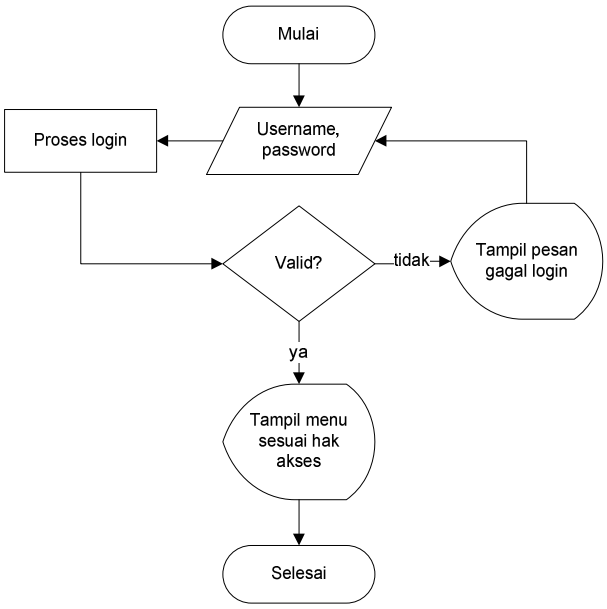
Diagram alir prosedur daftar adalah sebagai berikut:



Gambar 3. 181. Perancangan Prosedural Flowchart Daftar

3.2.5.2.Perancangan Prosedural *Login*

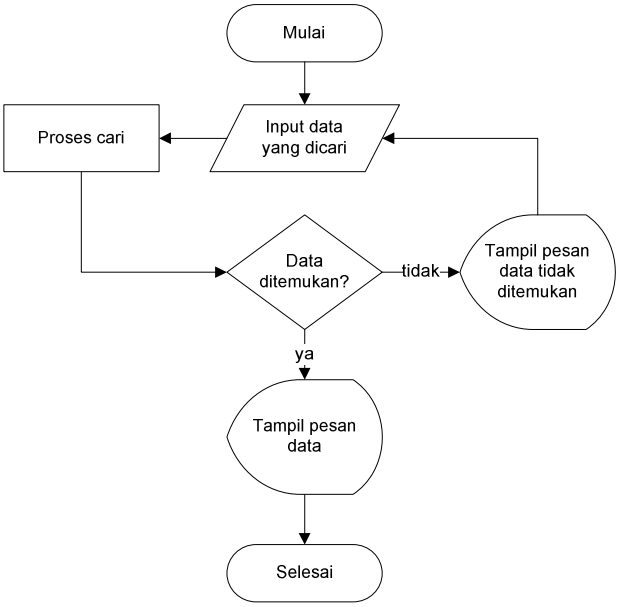
Diagram alir prosedur login adalah sebagai berikut:



Gambar 3. 182. Flowchart *Login*

3.2.5.3.Perancangan Prosedural Cari

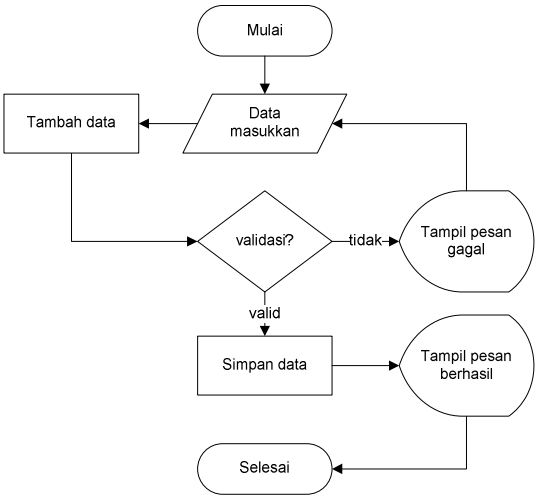
Diagram alir prosedur cari adalah sebagai berikut:



Gambar 3. 183. Flowchart Cari

3.2.5.4.Perancangan Prosedural Tambah

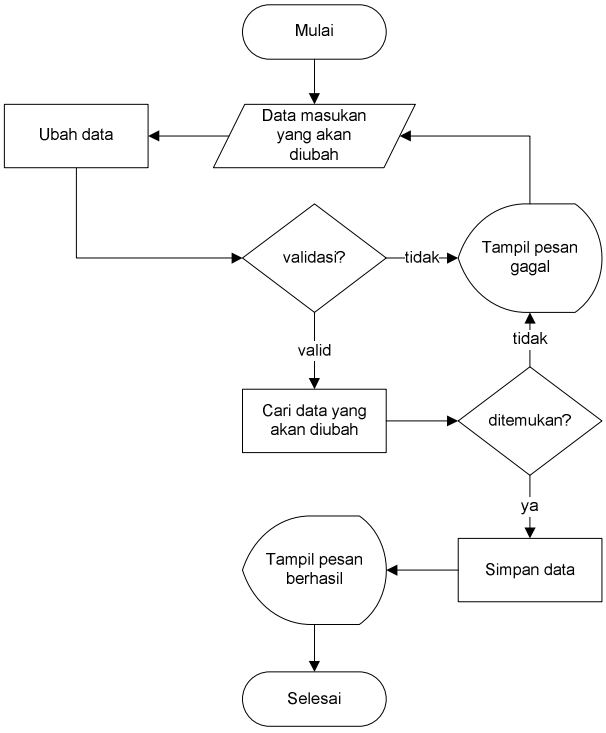
Diagram alir prosedur tambah adalah sebagai berikut:



Gambar 3. 184. Flowchart Tambah

3.2.5.5.Perancangan Prosedural Ubah

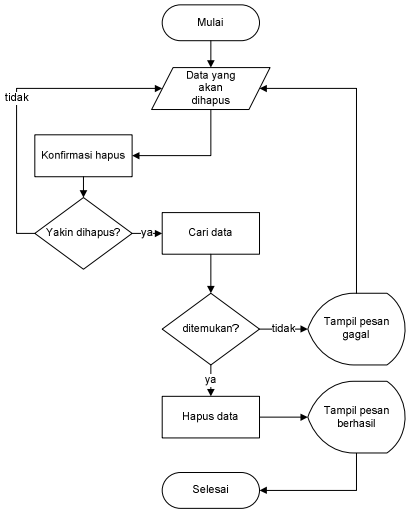
Diagram alir prosedur ubah adalah sebagai berikut:



Gambar 3. 185. Flowchart Ubah

3.2.5.6.Perancangan Prosedural Hapus

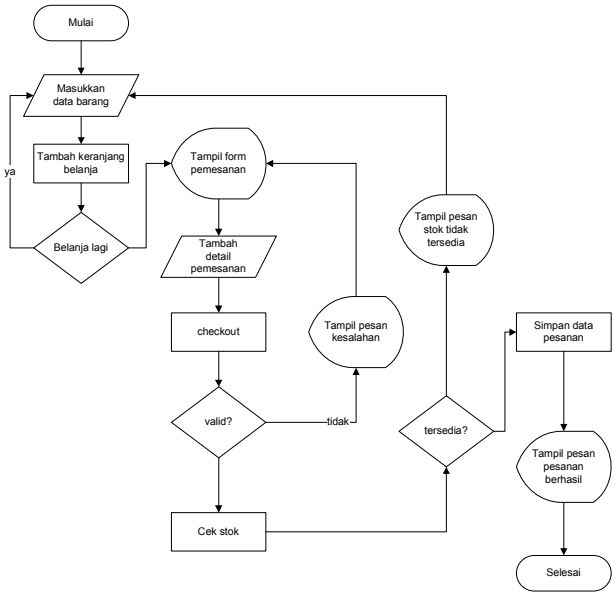
Diagram alir prosedur hapus adalah sebagai berikut:



Gambar 3. 186. Flowchart Hapus

3.2.5.7.Perancangan Prosedural Pemesanan

Diagram alir prosedur pemesanan adalah sebagai berikut:



Gambar 3. 187. FLowchart Pemesanan