BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai tahap yang bertujuan untuk memahami sistem, mengetahui kekurangan sistem dan menentukan kebutuhan sistem penjualan dan pembelian pada toko Rumah Buku. Dengan menganalisis prosedur sistem yang sedang berjalan dapat di evaluasi sehingga dapat dibuat satu usulan untuk sistem yang baru dari evaluasi tersebut.

3.1.1 Analisis Masalah

Dari hasil penelitian yang dilakukan pada toko Rumah Buku maka diperoleh permasalahan bahwa toko Rumah Buku kesulitan dalam mempromosikan dan memasarkan buku ke seluruh wilayah di Indonesia. Saat ini proses pemasaran buku dilakukan secara langsung yaitu pembeli langsung datang ke Rumah Buku untuk membeli buku. Promosi produk buku yang dijual hanya lewat mulut ke mulut dan lewat pamflet dan spanduk yang dipasang didepan toko, mengakibatkan calon pembeli yang memiliki kendala dalam jarak dan waktu tidak mendapatkan informasi yang dibutuhkan untuk berbelanja di Rumah Buku.

3.1.2 Analisis Prosedur yang sedang Berjalan

Analisis aliran data atau prosedur sangat diperlukan untuk mengetahui proses yang berjalan di suatu perusahaan dan sebagai dasar untuk membuat atau memperbaiki serta mempermudah prosedur yang sedang berjalan di perusahaan agar menjadi lebih baik lagi. Berdasarkan hasil wawancara dengan *supervisor*

toko Rumah Buku menyatakan perosedur bisnis yang sedang berjalan adalah sebagai berikut :

1. Prosedur penjualan buku
2. Prosedur pengadaan buku
3. Prosedur kedatangan buku

Gambaran tentang prosedur yang terjadi saat ini pada toko Rumah Buku dapat dilihat didalam *activity diagram* sistem yang sedang berjalan.

3.1.2.1 *Activity Diagram* Sistem yang sedang Berjalan

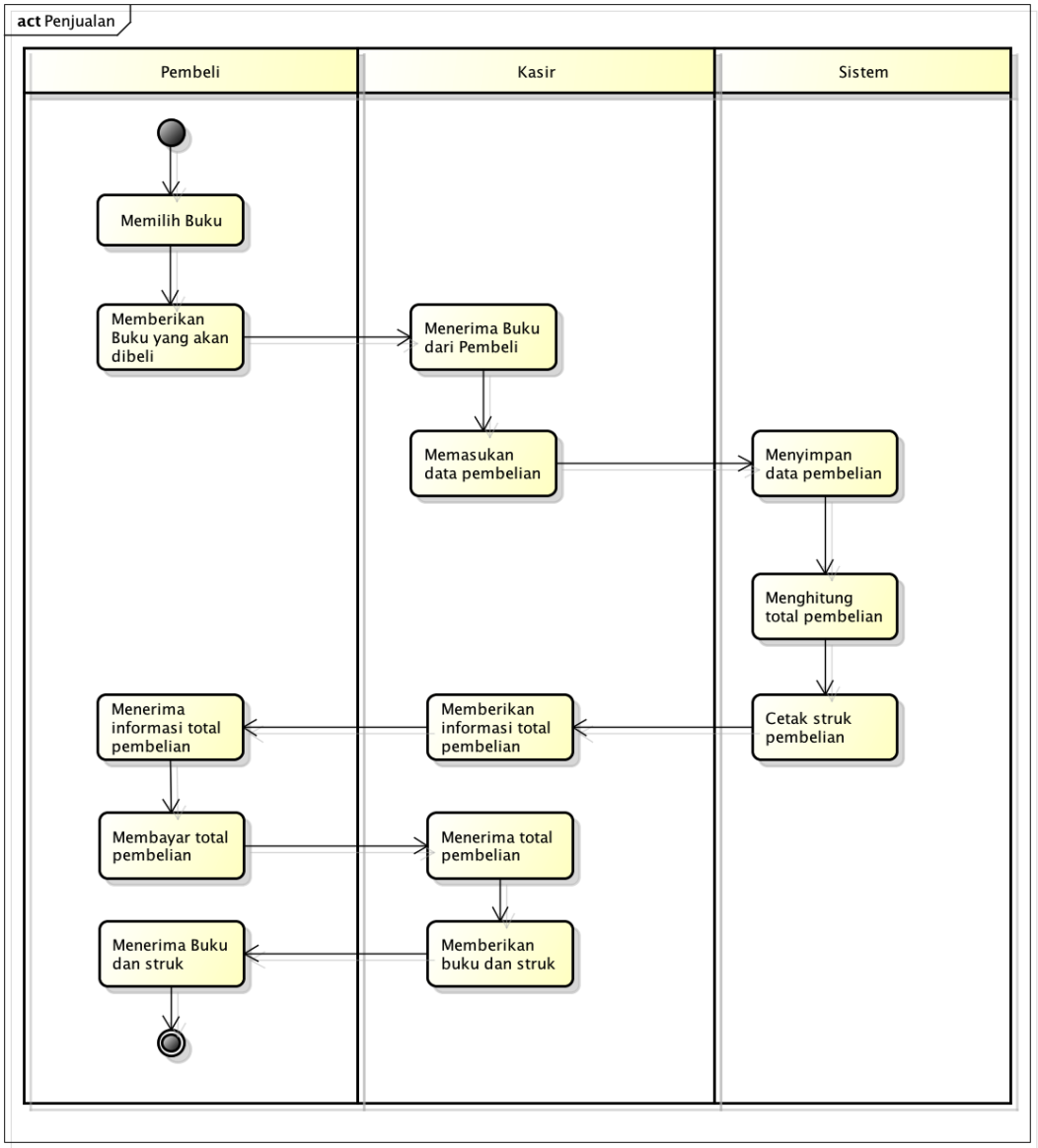
Acitivity diagram digunakan untuk memodelkan aspek dinamis dari sistem. *Activity diagram* berfungsi memodelkan alur kerja sebuah proses bisnis dan urutan aktivitas pada suatu proses. *Activity Diagram* sistem yang sedang berjalan di toko Rumah Buku adalah sebagai berikut:

3.1.2.2 *Activity Diagram* Penjualan Buku yang sedang Berjalan

Kegiatan yang dilakukan pada saat ini pada sistem penjualan adalah sebagai berikut :

1. Pembeli memilih buku yang ingin dibeli, kemudian pembeli melakukan pengecekan buku yang akan dibeli dan jika sesuai, maka pembeli melakukan pembayaran.
2. Pembeli melakukan pembayaran sesuai dengan total harga yang tertera.
3. Kasir menyimpan seluruh data pembelian ke sistem, kemudian kasir akan memberikan barang yang dibeli dan struk pembelian kepada pembeli.

Proses kegiatan penjualan yang saat ini sedang dilaksanakan dapat dilihat pada *activity diagram* pada Gambar 3.1



Gambar 3.1 *Activity diagram* penjualan yang sedang berjalan

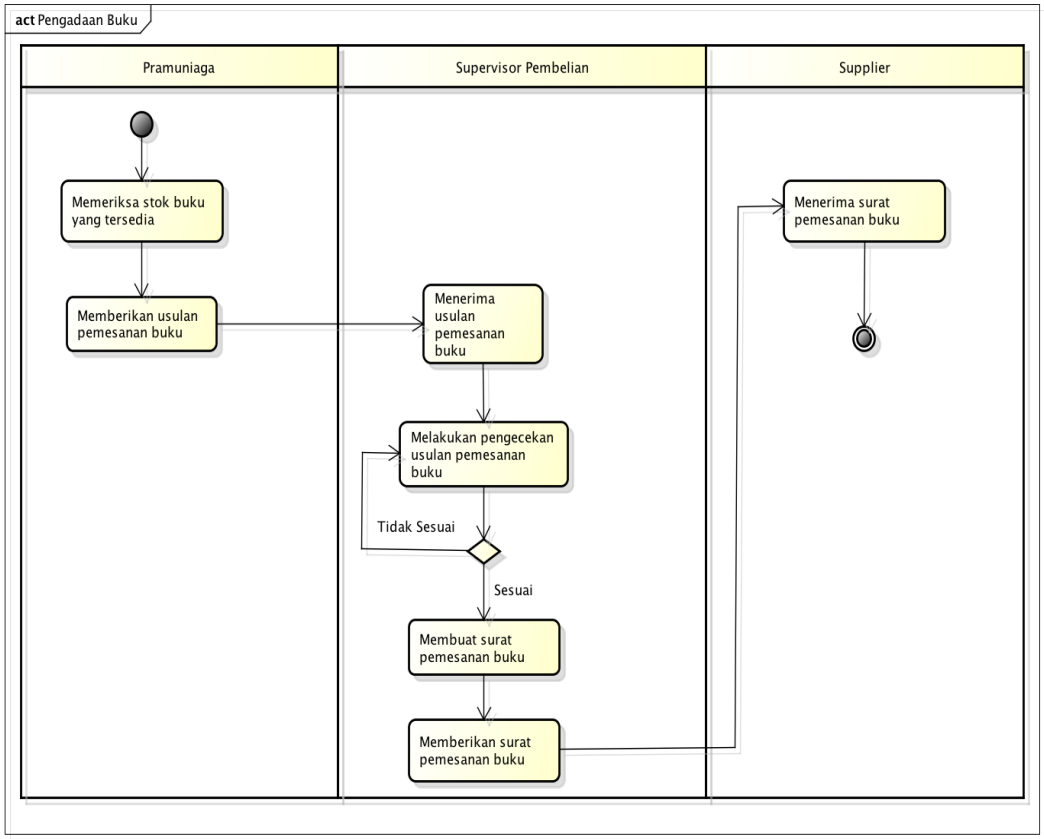
3.1.2.3 *Activity Diagram* Pengadaan Buku yang sedang Berjalan

Kegiatan yang dilakukan pada saat ini pada sistem pengadaan barang adalah sebagai berikut :

1. Pramuniaga memeriksa stok buku yang tersedia.
2. Jika buku berada pada stok minimal, maka pramuniaga akan mengusulkan ke *supervisor* pembelian untuk melakukan pemesanan.

3. *Supervisor* pembelian akan melakukan pengecekan terhadap usulan dari pramuniaga seperti jumlah buku yang akan dipesan.
4. Jika telah sesuai, maka *supervisor* pembelian akan membuat surat pemesanan buku ke *supplier*.

Proses kegiatan pengadaan barang yang saat ini sedang dilaksanakan dapat dilihat pada *activity diagram* pada Gambar 3.2



Gambar 3.2 *Activity diagram* pengadaan buku yang sedang berjalan

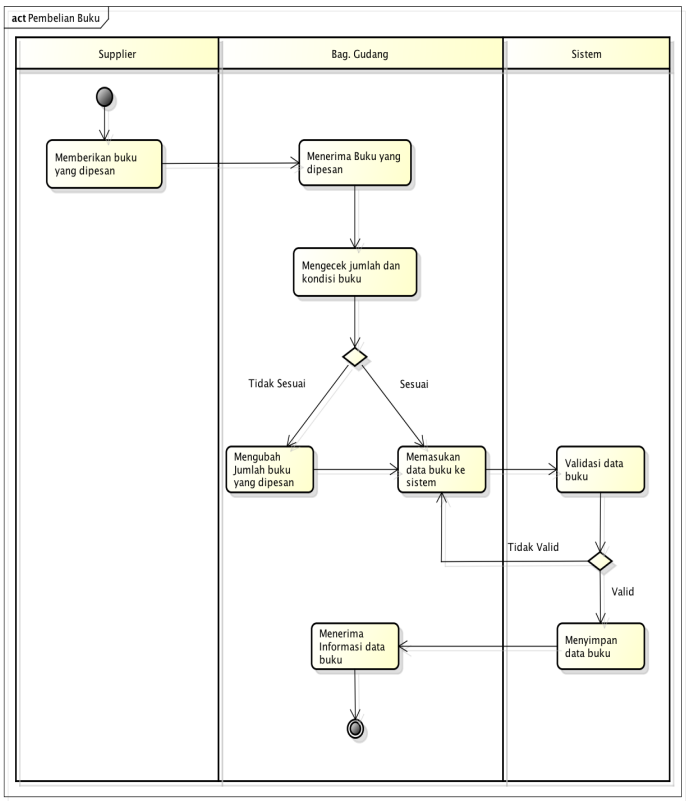
3.1.2.4 *Activity Diagram* Kedatangan Buku yang sedang Berjalan

Kegiatan yang dilakukan pada saat ini dalam sistem kedatangan buku adalah sebagai berikut :

1. *Supplier* akan memberikan barang yang dipesan oleh *supervisor* pembelian kepada bagian gudang.

2. Bagian gudang akan mengecek kondisi dan jumlah barang yang datang apakah sudah sesuai dengan pemesanan, kalau tidak sesuai maka akan membuat nota pemesanan baru.
3. Buku yang sampai ditempatkan sesuai dengan jenis kategori.

Proses kegiatan pembelian buku yang saat ini sedang dilaksanakan dapat dilihat pada *activity diagram* pada Gambar 3.3



Gambar 3.3 *Activity diagram* pembelian buku yang sedang berjalan

3.1.3 Aturan Bisnis

Aturan bisnis yang sedang berjalan pada toko Rumah Buku adalah sebagai berikut :

1. Rumah Buku hanya melayani pemesanan lingkup domestik saja.
2. Konsumen wajib menyertakan data pribadinya untuk kelengkapan proses penjualan seperti alamat lengkap dan alamat tujuan pengiriman barang.

3. Transaksi penjualan kepada konsumen Rumah Buku akan diberikan potongan harga 15% sampai 25% dari tiap buku yang dibeli sedangkan untuk toko online diberikan diskon 20% untuk semua buku.
4. Konsumen berhak menukar barang yang telah dibeli dikarenakan barang yang diterima mengalami kerusakan akibat cacat produksi ataupun kerusakan pada saat pengiriman dengan waktu retur adalah 2 hari setelah penerimaan buku.
5. Stok minimal buku adalah tiga buku, apabila stok buku mencapai stok minimal, maka akan dilakukan pemesanan kembali ke *supplier*.
6. Apabila konsumen ingin membeli buku dengan jumlah melebihi stok maka konsumen bisa melakukan pemesanan terlebih dahulu dan pihak Rumah Buku akan memberi tahu jika stok sudah ada atau sudah tidak tersedia melalui email atau telepon.
7. Stok buku dipastikan selalu ada dan jika suatu saat stok habis dan konsumen sudah membayar buku yang sudah dipesan maka akan dicarikan buku kepada penerbit di cabang yang lain dan apabila memang benar buku tidak ada maka uang akan dikembalikan oleh pihak Rumah Buku dengan cara pengembalian dibicarakan interal antara konsumen dengan pihak Rumah Buku.

3.1.4 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional menggambarkan keadaan sistem yang ada pada toko Rumah Buku, diantaranya perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.5 Analisis Perangkat Keras

Spesifikasi perangkat keras komputer yang ada di toko Rumah Buku Bandung dapat dilihat pada tabel 3.1

Tabel 3.1 Spesifikasi perangkat keras

No	Perangkat Keras	Spesifikasi
1	Prosessor	Intel Pentium 4 2.8 Ghz
2	Monitor	Monitor VGA 15”
3	VGA	Onboard 64 Mb
4	Memori	DDR2 1GB
5	Hard Disk	120 GB
6	Optical Drive	CD-RW 52x
7	Keyboard	Keyboard
8	Mouse	Mouse
9	Printer	Hitam putih dan warna
10	Jaringan Internet	Ada

Sedangkan untuk sistem yang akan dibangun, dibutuhkan perangkat keras dengan spesifikasi minimum dapat dilihat pada tabel 3.2 :

Tabel 3.2 Spesifikasi perangkat keras minimal

No	Perangkat Keras	Spesifikasi
1	Prosessor	Intel Pentium 4 1.3 Ghz
2	Monitor	Monitor VGA 15”
3	VGA	Onboard 32 Mb
4	Memori	DDR2 512GB
5	Hard Disk	2 GB
6	Keyboard	Keyboard
7	Mouse	Mouse
8	Printer	Hitam putih dan warna
9	Jaringan Internet	Ada

Dari analisis yang telah dijelaskan, dapat disimpulkan bahwa spesifikasi perangkat keras yang dimiliki Rumah Buku telah memenuhi standar spesifikasi minimum perangkat keras yang dibutuhkan untuk dapat mengimplementasikan website *E- Commerce* pada Rumah Buku.

3.1.6 Analisis Perangkat Lunak

Sistem operasi yang digunakan pada toko Rumah Buku yaitu Microsoft Wondows XP dan *browser* yang digunakan yaitu Mozilla Firefox 3.6. Sedangkan untuk pembangunan aplikasi, dibutuhkan perangkat lunak dengan spesifikasi yang dapat dilihat pada tabel 3.3

Tabel 3.3 Spesifikasi perangkat lunak

No	Perangkat Keras	Spesifikasi
1	Prosessor	Intel Pentium 4 1.3 Ghz
2	Web Server	Wamp
3	Basis Data	MySql
4	Software Pembangun	Dreamweaver CS 3
5	Browser	Mozilla Firefox, Chrome, Safari
6	Bahasa Pemrograman	PHP

Sedangkan untuk menggunakan aplikasi yang dibangun dibutuhkan perangkat lunak dengan spesifikasi minimal yang dapat dilihat pada tabel 3.4

Tabel 3.4 Spesifikasi perangkat lunak

No	Perangkat Keras	Spesifikasi
1	Prosessor	Intel Pentium 4 1.3 Ghz
2	Web Server	Wamp
3	Browser	Mozilla Firefox, Safari, Chrome

Dari analisis diatas dapat disimpulkan bahwa spesifikasi perangkat lunak yang dimiliki Rumah Buku telah memenuhi standar spesifikasi minimum perangkat lunak yang dibutuhkan untuk dapat mengimplementasikan website *E-Commerce* pada Rumah Buku.

3.1.7 Analisis Pengguna

Suatu sistem akan berjalan optimal apabila ditunjang oleh perangkat pikir yang memiliki kemampuan dalam menjalankan suatu sistem. Adapun analisis pengguna (*user*) di toko Rumah Buku saat ini yang memiliki rincian sebagai berikut :

Supervisor

Umur	:	23 – 45 tahun
Pendidikan Terakhir	:	Strata I (S1)
Kemampuan yang dimiliki	:	Mampu menggunakan perangkat lunak *Microsoft Office* dalam menjalankan tugasnya.

Bagian Gudang

Umur	:	21-30 tahun
Pendidikan terakhir	:	SMA - Diploma 3 (D3)
Kemampuan yang dimiliki	:	Mampu menggunakan komputer dengan cukup baik.

Sistem yang akan dibangun ini digunakan oleh tiga jenis pengguna utama yaitu *admin,* petugas dan konsumen. *Admin* dapat melakukan pengelolaan terhadap aplikasi ini seperti tambah data, hapus data, edit semua data termasuk data petugas. Petugas dapat melakukan pengolahan data barang dan transaksi penjualan, sedangkan konsumen hanya dapat melakukan pemesanan barang. Analisis pengguna dari aplikasi *e-commerce* yang dibangun dapat dilihat pada tabel 3.5

Tabel 3.5 Analisis pengguna

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Keterampilan
Admin	Mengelola data yang terdapat dalam sistem	Memiliki semua hak akses di dalam sistem termasuk pengolahan data petugas	Bisa menggunakan komputer, mengerti cara penggunaan untuk *web*, dan dapat menggunakan *internet*, dan menguasai mysql dan php.
Petugas	Mengelola data yang terdapat pada sistem	Melakukan pemasukan data buku, mengolah transaksi penjualan, menyetujui pesanan dan	Bisa menggunakan komputer, mengerti cara penggunaan untuk *web* dan mengerti bisnis *rule* perusahaan.

		membuat laporan penjualan.	
Konsumen	Dapat melakukan transaksi pemesanan buku	Melakukan pembelian barang secara *online*	Dapat menggunakan komputer dan dapat menggunakan *internet* untuk *browsing*

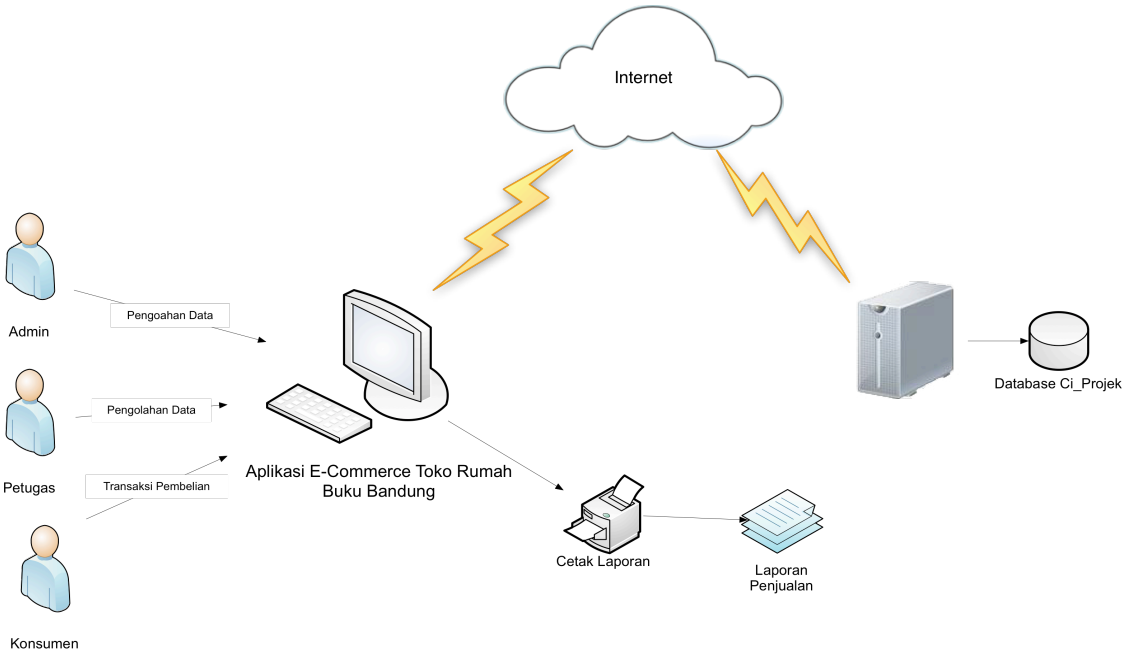
Melihat hasil analisis pengguna yang ada pada sistem yang sedang berjalan maka diperlukan adanya penambahan *user* serta peningkatan kualifikasi pada *user*. Untuk memenuhi kebutuhan sebagai pengguna sistem tersebut maka diperlukan adanya pelatihan dalam penggunaan sistem terhadap *user*. Dengan diadakannya pelatihan diharapkan *user* yang ada akan sebaik mungkin menggunakan aplikasi yang dibangun.

3.1.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk mengetahui proses informasi yang mengalir melalui perangkat lunak. Analisis dilakukan dengan mendeskripsikan fungsi sistem yang dibangun, mengidentifikasi aktor, membuat *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram* dan *deployment diagram*.

3.1.8.1 Deskripsi Umum Sistem

Sistem yang dibuat merupakan aplikasi *e-commerce* yang digunakan pada toko Rumah Buku Bandung. Sistem yang dibangun dapat melayani konsumen yang hendak melakukan transaksi pembelian buku secara *online*, selain itu juga membantu petugas dalam pengelolaan stok buku yang ada di Toko Rumah Buku Bandung. Sistem yang dibangun akan diimplementasikan pada toko Rumah Buku yang terhubung langsung kedalam *database* Ci_Projek.



Gambar 3.4 Perspektif produk

Seperti terlihat pada gambar 3.4, petugas dan *admin* terhubung ke dalam sistem. Semua data transaksi jual beli disimpan dalam *database* yang hanya dapat diakses oleh petugas dan admin yang ada di toko Rumah Buku. *Admin* dan dapat mengakses semua fungsionalitas yang terdapat dalam aplikasi yang dibangun dan petugas hanya dapat mengakses fungsionalitas pengolahan data master dan transaksi saja sehingga hasil yang diperoleh berupa informasi dan laporan dari hasil penjualan tiap bulan.

3.1.8.2 Fungsi website penjulan *online*

Secara umum, *website* penjualan *online* berfungsi sebagai media untuk melakukan penjualan buku pada toko Rumah Buku dengan melakukan pemesanan barang secara *online*. Fungsi utama perangkat lunak yang dibangun dibagi menjadi beberapa fasilitas untuk setiap aktor yang meliputi menu-menu apa saja yang ada pada aplikasi yang dibagun.

1. Fasilitas yang disediakan untuk aktor *konsumen* yaitu :
    a. Setiap konsumen dapat melakukan pencarian buku dan menyebarkan detail buku ke jejaring sosial konsumen seperti *facebook* dan *twitter*.
    b. Konsumen dapat melakukan pembelian secara *online.*
    c. Konsumen dapat mengirimkan pertanyaan kepada admin rumah buku menggunakan fasilitas kontak kami.
    d. Konsumen dapat melakukan retur buku sesuai dengan buku yang dibeli.
    e. Konsumen dapat melakukan konfirmasi pembayaran sesuai dengan jumlah harga yang harus dibayar.
2. Fasilitas yang disediakan untuk *admin* yaitu :
    a. Setiap *admin* dapat mengelola data petugas dimana *admin* bisa menambah, mengubah dan menghapus data *user* (admin dan petugas) beserta grup *user*.
    b. *Admin* dapat melakukan *backup database*.
    c. *Admin* dapat mengolah data *master* yaitu data buku, kategori, subkategori, dan penerbit dimana data master tersebut dapat ditambah, diubah dan dihapus oleh admin.
    d. *Admin* dapat mengolah data transaksi yaitu data pemesanan, pesan buku, kirim promosi, retur dan balas kontak dimana data master tersebut dapat ditambah, diubah dan dihapus oleh admin.
    e. Admin dapat membuat laporan penjualan dimana di dalam menu laporan penjualan admin dapat memilih periode bulan dan tahun penjualan yang akan dicetak.

f. Admin dapat membuat laporan penjualan dimana di dalam menu laporan penjualan admin dapat memilih periode bulan dan tahun penjulaan yang akan dicetak.

g. *Logout* (proses untuk keluar dari sistem).

3. Fasilitas yang disediakan untuk petugas yaitu :

a. Petugas dapat mengolah data *master* yaitu data buku, kategori, subkategori, dan penerbit dimana data master tersebut dapat ditambah, diubah dan dihapus oleh admin.

b. Petugas dapat mengolah data transaksi yaitu data pemesanan, pesan buku, kirim promosi, retur dan balas kontak dimana data master tersebut dapat ditambah, diubah dan dihapus oleh admin.

c. Petugas dapat membuat laporan penjualan dimana di dalam menu laporan penjualan admin dapat memilih periode bulan dan tahun penjualan yang akan dicetak.

d. Petugas dapat membuat laporan penjualan dimana di dalam menu laporan penjualan admin dapat memilih periode bulan dan tahun penjulaan yang akan dicetak.

e. *Logout* (proses untuk keluar dari sistem).

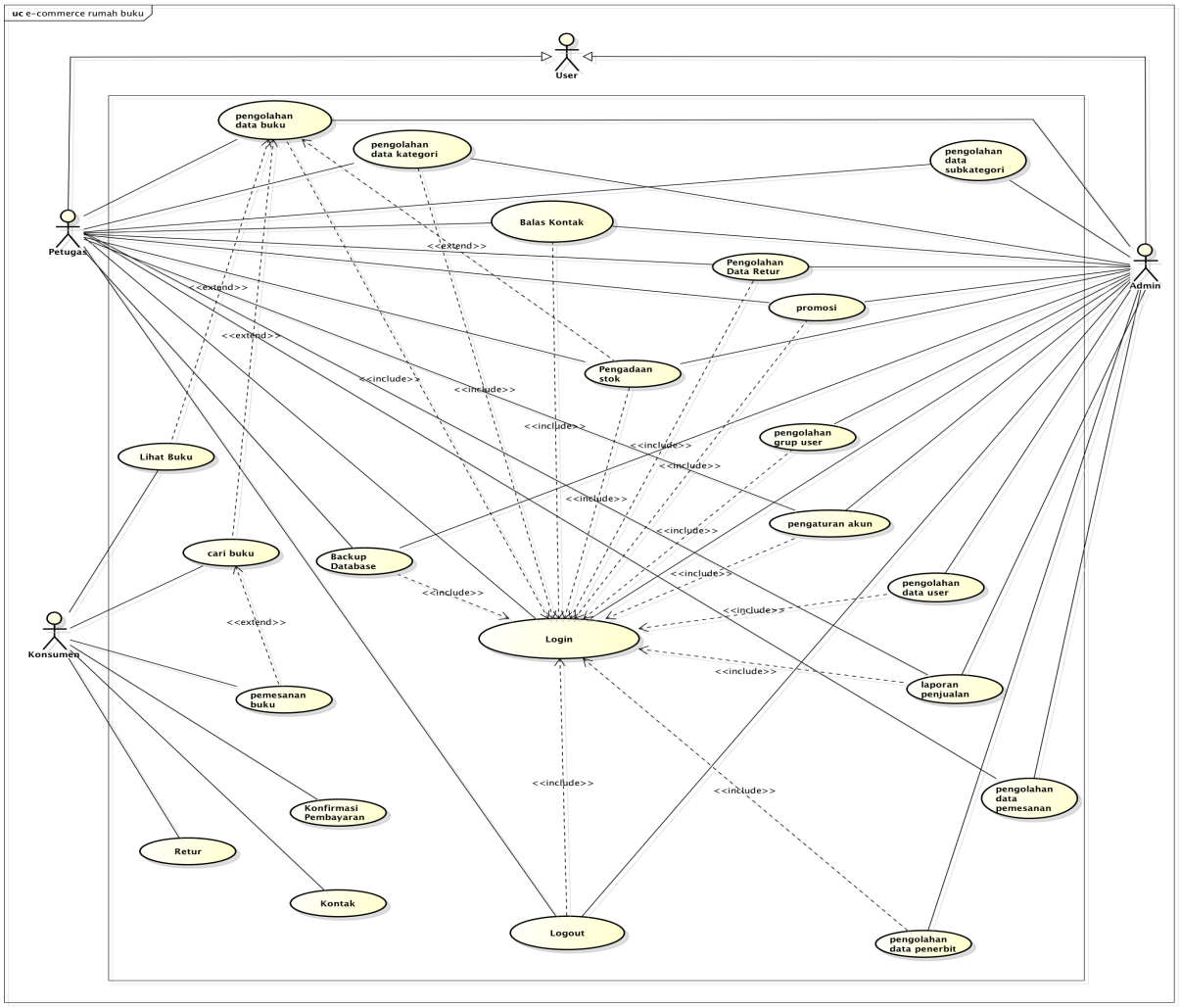
3.1.8.3 Identifikasi Aktor

Aplikasi *e-commerce* yang dibangun akan digunakan oleh tiga orang aktor yaitu *admin*, konsumen dan petugas. Deskripsi dari masing-masing aktor adalah sebagai berikut :

1. Aktor pertama adalah *admin* sebagai pengelola yang semua hak akses yang ada pada sistem termasuk untuk menambah, mengubah dan menghapus petugas.
2. Aktor kedua adalah konsumen sebagai pengunjung yang memiliki hak untuk melakukan transaksi pembelian.
3. Aktor ketiga adalah petugas yang memiliki hak akses untuk memasukan data barang, mengolah data penjualan dan membuat laporan.

3.1.8.4 *Usecase* Diagram

Pemodelan *use case* adalah pemodelan sistem dari perspektif pandangan pemakai akhir *(end user)*. Model *use case* adalah pandangan dari luar sistem, sementara model rancangan adalah pandangan dari dalam. Model *use case* menangkap penggunaan-penggunaan sistem, sedangkan model rancangan merepresentasikan pembangunan dari sistem. *Use case diagram* dari aplikasi *e-commerce* pada toko Rumah Buku dapat dilihat pada gambar 3.5



Gambar 3.5 *Use case e-commerce* pada toko Rumah Buku

3.1.8.5 Scenario Diagram dan Sequence Diagram

Scenario diagram berfungsi untuk menjelaskan lebih detail mengenai modul-modul yang terdapat dalam *website e-commerce* yang dibangun. *Sequence diagram* digunakan untuk menggambarkan interaksi antar objek didalam dan disekitar sistem yang berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antara dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Scenario* diagram dan *sequence* diagram dari *use case e-commerce* pada toko Rumah Buku adalah sebagai berikut :

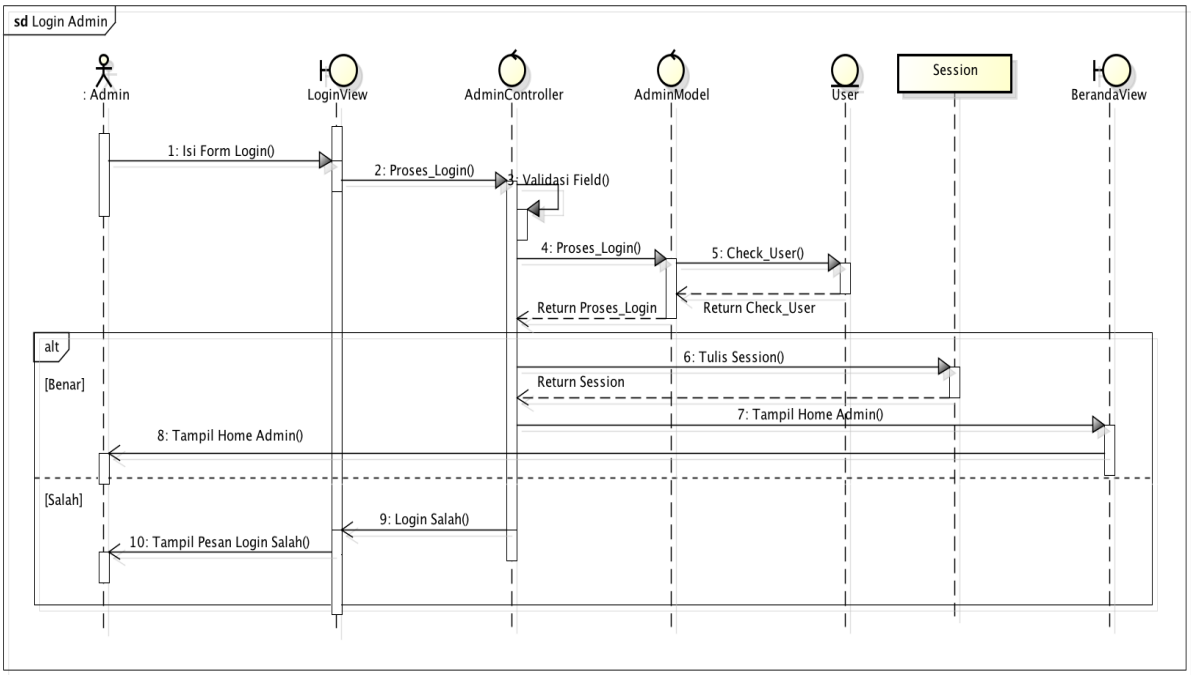
1. *Login*

   Interaksi antara aktor dengan *use case login* dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.6

Tabel 3.6 Skenario *use case login admin*

Nama *Use Case*	*Login*
Aktor	*Admin,* Petugas
Tujuan	Masuk kedalam sistem sebagai *admin*
Deskripsi	Proses *login* merupakan proses autentikasi untuk menggunakan sistem.
Keadaan Awal	Menampilkan halaman *login admin*
Aktor	Sistem
1. Mengisi *form login*	2. Sistem merespon dengan memeriksa *username* dan *password* yang dimasukkan, jika *username* dan *password* terdapat didalam *database*, maka sistem menampilkan halaman *administrator* jika tidak, maka halaman *login* akan ditampilkan ulang dan menampilkan pesan kesalahan.
Kondisi Akhir	Hak akses pengguna sebagai *admin* dan petugas sudah dipastikan, sistem menampilkan halaman utama *administrator*
Skenario Alternatif – *Login Admin* Gagal	
Deskripsi	Jika data yang diisi tidak *valid*, maka sistem kembali menampilkan *form login admin* dan memberi pesan bahwa data yang dimasukkan tidak *valid*

Sequence diagram login admin dapat dilihat pada gambar 3.6



Gambar 3.6 *Sequence diagram login*

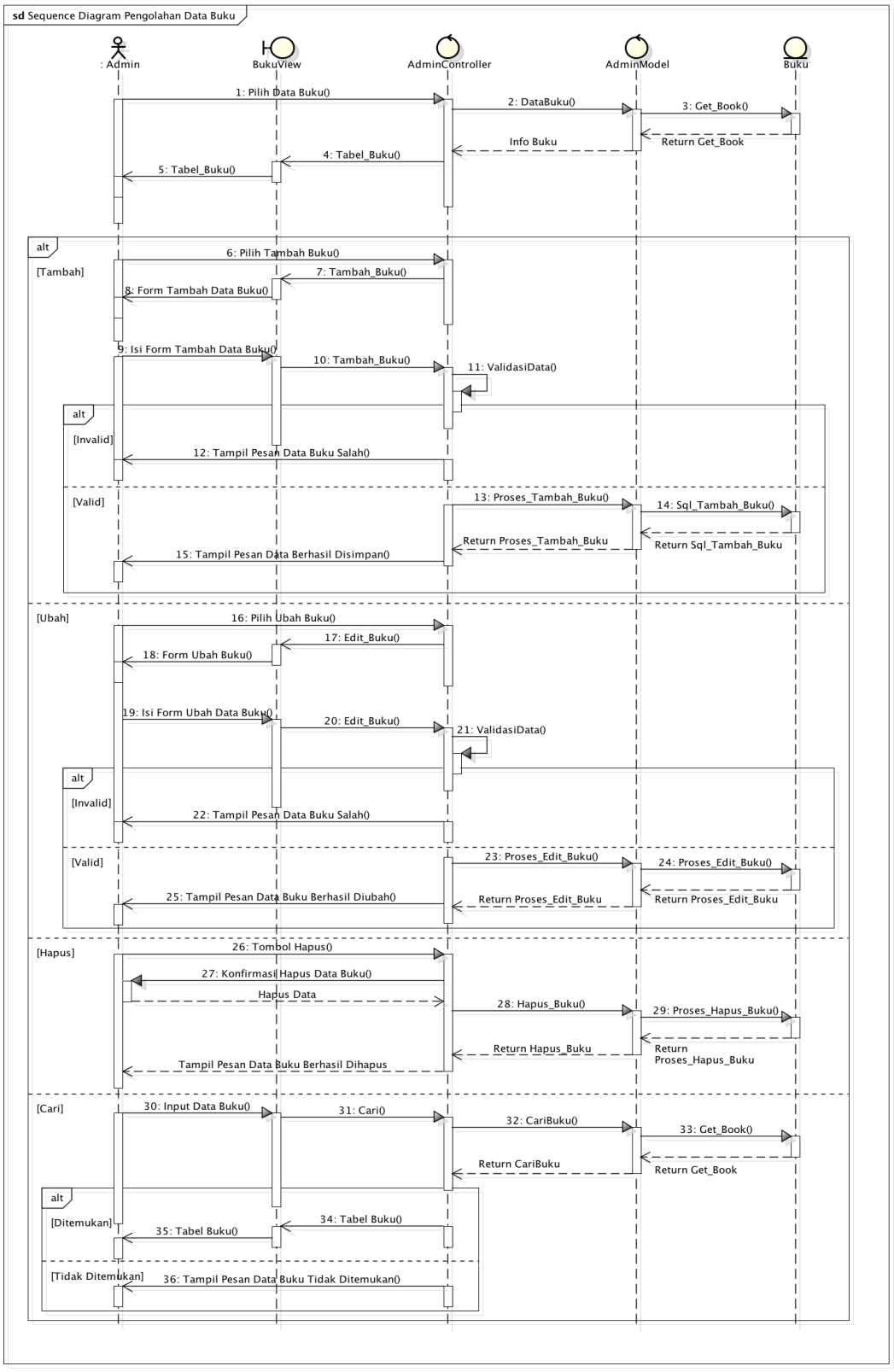
2. Pengolahan Data Buku

Interaksi antara aktor dengan *use case* pengolahan data buku dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.7

Tabel 3.7 Skenario *Use Case* Pengolahan Data Buku

Nama *Use Case*	Pengolahan Data Buku
Aktor	*Admin,* Petugas
Tujuan	Menampilkan daftar buku dan menampilkan menu utama *administrator* untuk mengolah data buku
Deskripsi	Pengguna sebagai *admin,* petugas dapat melakukan kegiatan mencari, menambah, mengubah, dan menghapus data buku
Keadaan Awal	*Admin,* petugas sedang berada dalam sesi *login* sebagai *admin*
Aktor	Sistem
1. Memilih menu "tambah buku"	2. Menampilkan *form* tambah data buku
3. Mengisi *form* tambah data buku dan menekan tombol simpan	4. Sistem memeriksa data yang dimasukkan. Jika data yang dimasukkan lengkap, maka data akan disimpan ke *database* dan sistem akan mengarahkan ke halaman buku, tetapi jika data yang dimasukkan tidak lengkap, maka sistem akan menampilkan ulang *form* tambah data dan menampilkan pesan kesalahan.
5. Memilih menu "ubah buku" yang diingikan	6. Sistem menampilkan *form* isian data buku lengkap dengan data yang akan diubah.
7. Mengubah data buku pada *form* ubah dan menekan tombol simpan	8. Sistem akan memeriksa data masukkan. Jika data yang dimasukkan lengkap, maka data yang baru akan disimpan, jika data yang dimasukkan tidak lengkap maka sistem akan menampilkan pesan kesalahan dan menampilkan ulang *form* ubah data buku
9. Memilih data buku yang ingin dihapus.	10. Sistem akan menampilkan pesan konfirmasi apakah data akan dihapus
11. Memilih menu pilihan "OK" pada pesan hapus data yang diperlihatkan oleh sistem	12. Sistem menghapus data yang diinginkan.
13. Memilih menu pilihan "Cancel" pada pesan hapus data yang diperlihatkan oleh sistem	14. Sistem tidak melakukan penghapusan data buku yang diinginkan
15. Memilih menu cari dan memasukan data buku yang diinginkan dengan memasukan data buku seperti judul buku.	16. Sistem menampilkan data buku yang diinginkan
Kondisi Akhir	*Admin,* petugas berhasil melakukan kegiatan mencari, menambah, mengubah, dan menghapus data buku

Sequence diagram dari pengolahan data buku dapat dilihat pada gambar 3.7



Gambar 3.7 *Sequence diagram* pengolahan data buku

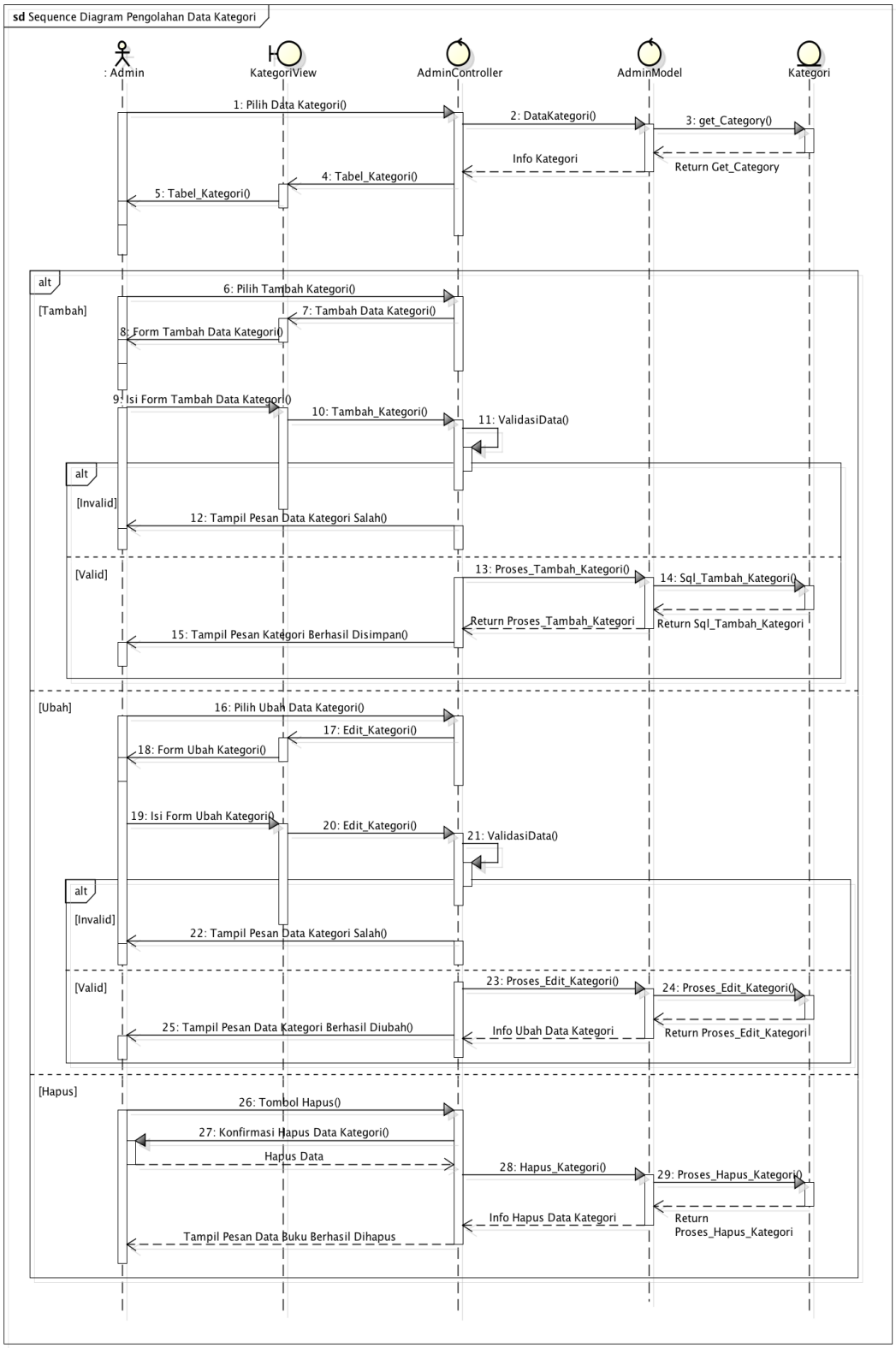
3. Pengolahan Data Kategori

Interaksi antara aktor dengan *use case* pengolahan kategori buku dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.8

Tabel 3.8 Skenario *use case* pengolahan data kategori

Nama *Use Case*		Pengolahan Data Kategori
Aktor		*Admin,* petugas
Tujuan		Menampilkan daftar kategori buku dan menampilkan menu utama *administrator* untuk mengolah data kategori
Deskripsi		Pengguna sebagai *admin,* petugas dapat melakukan kegiatan menambah, mengubah, dan menghapus data kategori buku
Keadaan Awal		Admin, petugas sedang berada di menu data kategori
Aktor		Sistem
1. Memilih menu tambah kategori		2. Menampilkan form tambah kategori
3. Mengisi form tambah kategori dan menekan tombol simpan		4. Memeriksa data yang dimasukkan. Jika data yang dimasukkan lengkap, maka data akan disimpan ke *database* dan sistem akan mengarahkan ke halaman kategori, tetapi jika data yang dimasukkan tidak lengkap, maka sistem akan menampilkan ulang *form* tambah kategori dan menampilkan pesan kesalahan
5. Memilih menu ubah data kategori buku yang diinginkan		6. Sistem menampilkan *form* isian data kategori lengkap dengan data yang akan diubah.
7. Mengubah data kategori pada *form* ubah dan menekan tombol simpan		8. Sistem akan memeriksa data yang dimasukan. jika data yang dimasukkan lengkap, maka data yang baru akan disimpan, jika data yang dimasukkan tidak lengkap, maka sistem akan menampilkan pesan kesalahan dan menampilkan ulang *form* isian ubah data kategori buku
9. Memilih kategori yang akan dihapus.		10. Menampilkan pesan konfiramsi apakah kategori akan dihapus
11. *Admin* menekan pilihan ”OK” pada pesan hapus data yang diperlihatkan oleh sistem		12. Sistem menghapus data yang diinginkan
13. *Admin* menekan pilihan ”Cancel” pada pesan hapus data yang diperlihatkan oleh sistem		14. Sistem tidak melakukan melakukan penghapusan kategori
Kondisi Akhir	*Admin* berhasil melakukan kegiatan menambah, mengubah, dan menghapus data kategori buku	

Sequence diagram pengolahan data kategori dapat dilihat pada gambar 3.8



Gambar 3.8 *Sequence diagram* pengolahan data kategori

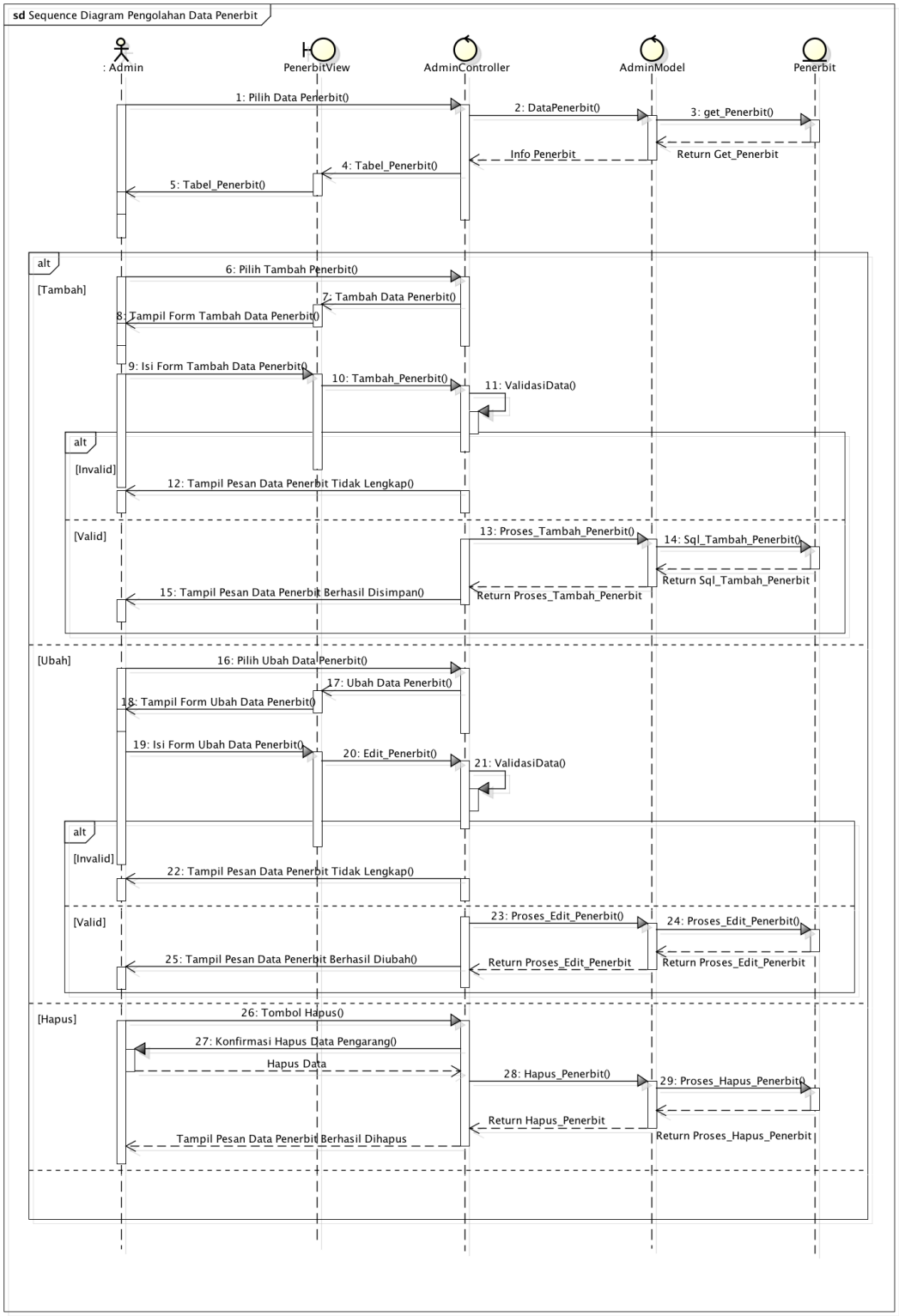
4. Pengolahan Data Penerbit

Interaksi antara aktor dengan *use case* pengolahan data penerbit dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.9

Tabel 3.9 Skenario *Use Case* Pengolahan Data Penerbit

Nama *Use Case*		Pengolahan Data Penerbit
Aktor		*Admin,* Petugas
Tujuan		Menampilkan daftar penerbit buku dan menampilkan menu utama administrator untuk mengolahdata penerbit
Deskripsi		*Admin, petugas* dapat melakukan kegiatan menambah, merubah, dan menghapus data penerbit
Keadaan Awal		*Admin,* petugas sedang berada pada halaman penerbit
Aktor		Sistem
1. Memilih menu tambah penerbit		2. Menampilkan form tambah penerbit
3. Mengisi *form* tambah penerbit dan menekan tombol simpan		4. Sistem memeriksa data yang dimasukkan, jika data yang dimasukkan itu lengkap, maka data yang dimasukkan akan disimpan ke *database* dan mengarahkan *admin* ke halaman penerbit, tetapi jika data yang dimasukkan tidak lengkap, maka sistem akan menampilkan ulang *form* tambah data dan menampilkan pesan kesalahan
5. Memilih menu ubah data penerbit yang diingikan		6. Menampilkan *form* isian data penerbit lengkap dengan data yang akan diubah
7. Mengisi data penerbit yang ingin di ubah		8. Sistem akan memeriksa data masukkan *admin*, jika data yang dimasukkan benar, maka data yang baru akan disimpan, jika data yang dimasukkan tidak lengkap maka sistem akan menampilkan pesan kesalahan dan menampilkan ulang *form* isian ubah data penerbit
9. Menghapus data penerbit yang diinginkan.		10. Menampilkan Pesan Konfirmasi apakah data penerbit akan dihapus
11. Memilih menu pilihan ”OK” pada pesan hapus data yang diperlihatkan oleh sistem		12. Sistem menghapus data yang diinginkan
13. Memilih menu pilihan ”Cancel” pada pesan hapus data yang diperlihatkan oleh sistem		14. Sistem tidak melakukan aktivitas penghapusan data penerbit yang diinginkan
Kondisi Akhir	*Admin* berhasil melakukan kegiatan menambah, merubah, dan menghapus data penerbit	

Sequence diagram pengolahan data penerbit dapat dilihat pada gambar 3.9



Gambar 3.9 *Sequence* diagram pengolahan data penerbit

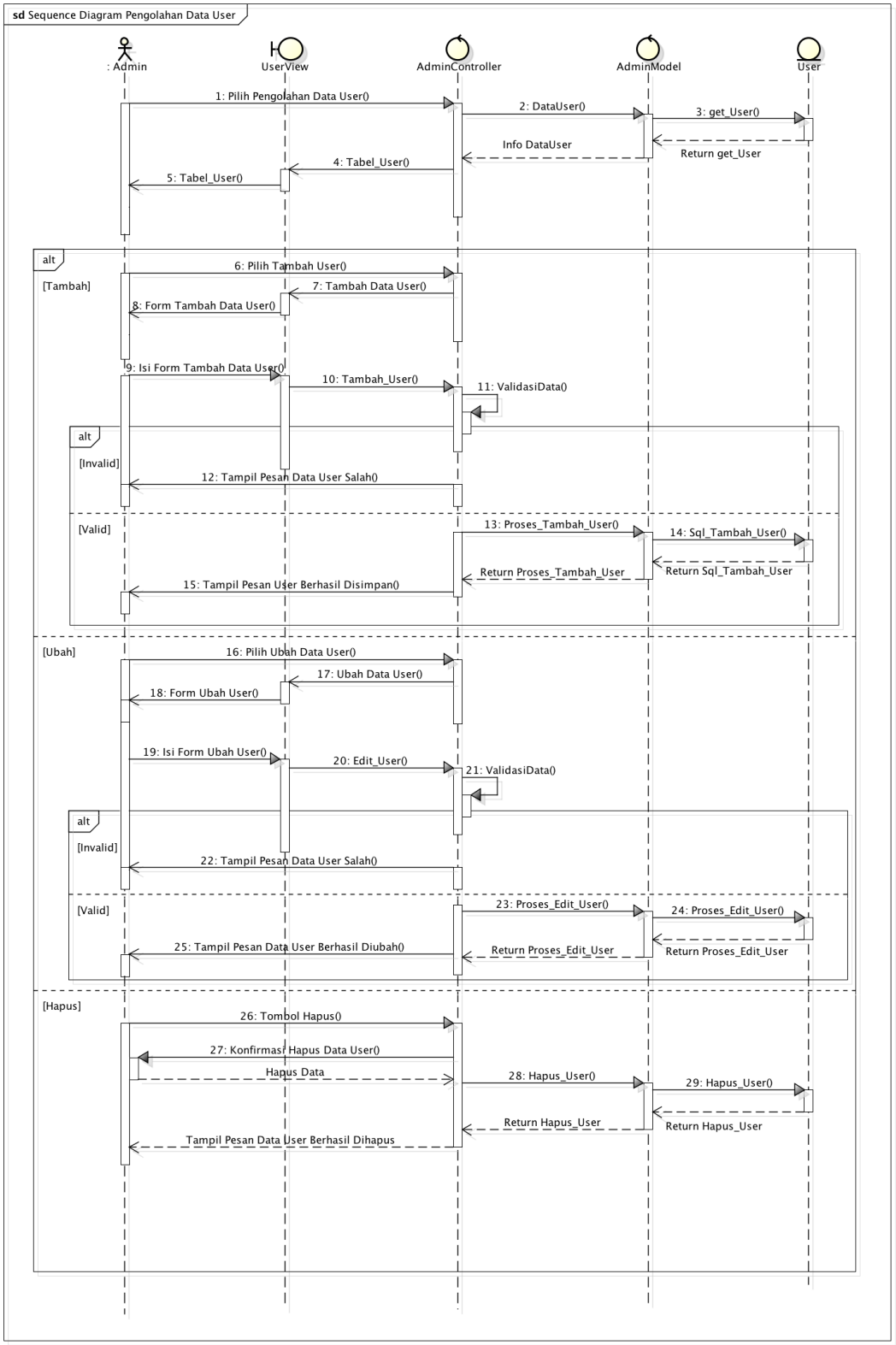
5. Pengolahan Data *User*

Interaksi antara aktor dengan *use case* pengolahan data *admin* dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.10

Tabel 3.10 Skenario *Use Case* Pengolahan Data *User*

Nama *Use Case*		Pengolahan Data *User*
Aktor		*Admin*
Tujuan		Mengolah data *admin,* dan petugas
Deskripsi		*Admin* dapat melakukan kegiatan menambah, mengubah, dan menghapus data *admin* dan petugas
Keadaan Awal		*Admin* sedang berada dalam pada halaman pengolahan data *user*
Aktor		Sistem
1. Memilih menu tambah *user*		2. Menampilkan form tambah data *user*
3. Mengisi form tambah data *user*, kemudian menekan tombol simpan		4. Memeriksa data yang dimasukkan oleh *admin*, jika data yang dimasukkan lengkap, maka akan disimpan ke *database* dan mengarahkan *admin* ke halaman *admin*, tetapi jika data yang dimasukkan tidak lengkap, maka sistem akan menampilkan ulang *form* tambah data admin dan menampilkan pesan kesalahan
5. Memilih menu ubah data *user* yang diinginkan		6. Sistem akan menampilkan *form* isian data *user* lengkap dengan data yang akan diubah
7. Mengubah data *user* dan kemudian menekan tombol simpan		8. Memeriksa data masukkan *admin*, jika data yang dimasukkan lengkap, maka data yang baru akan disimpan, jika data yang dimasukkan tidak *valid* maka sistem akan menampilkan pesan kesalahan dan menampilkan ulang *form* isian ubah data *user*
9. Memilih data admin .		10. Menampilkan pesan konfirmasi apakah data user akan dihapus
11. Menekan pilihan ”OK” pada pesan hapus data yang diperlihatkan oleh sistem		12. Sistem menghapus data yang diinginkan oleh *admin*
13. Menekan menu pilihan ”Cancel” pada pesan hapus data yang diperlihatkan oleh sistem		14. Sistem tidak melakukan proses hapus data *admin*
Kondisi Akhir	*Admin* berhasil melakukan kegiatan menambah, mengubah, dan menghapus data *admin*	

Sequence diagram pengolahan data *user* dapat dilihat pada gambar 3.10



Gambar 3.10 *Sequence diagram* pengolahan data *user*

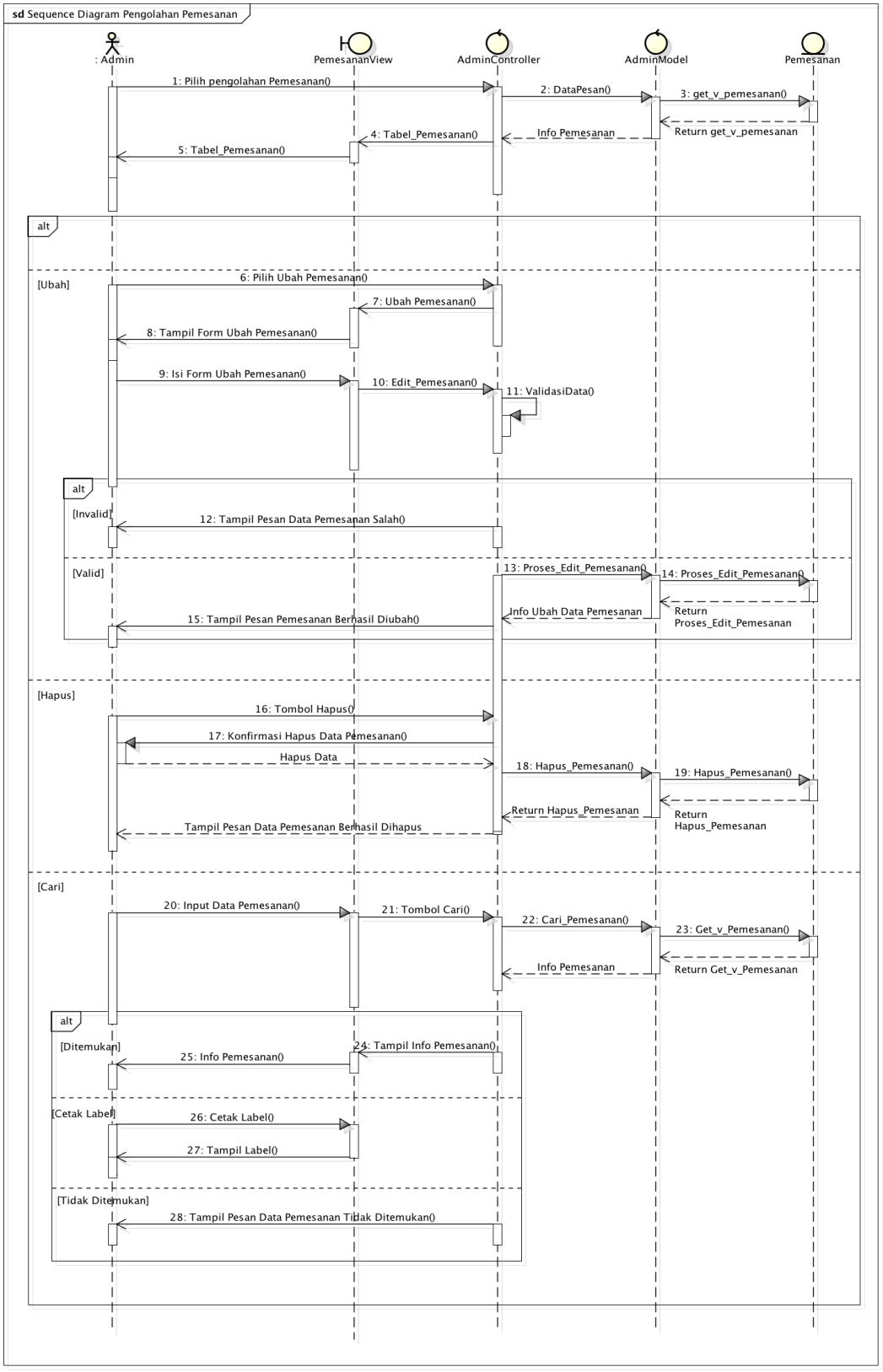
6. Pengolahan Data Pemesanan

Interaksi antara aktor dengan *use case* pengolahan data pemesanan dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.11

Tabel 3.11 Skenario *use case* pengolahan pemesanan

Nama *Use Case*		Pengolahan Pemesanan
Aktor		*Admin,* Petugas
Tujuan		Mengolah data pemesanan
Deskripsi		*Admin* dapat melakukan kegiatan mengubah,mencari, menghapus data pemesanan dan mencetak faktur
Keadaan Awal		*Admin*, petugas sedang berada pada halaman pengolahan pemesanan
Aktor		Sistem
1. Memilih menu ubah data pemesanan yang diinginkan		2. Menampilkan *form* isian data pemesanan lengkap dengan data yang akan diubah
3. Mengisi data pemesanan yang ingin diubah dan menekan tombol simpan		4. Memeriksa data masukkan. jika data yang dimasukkan lengkap, maka data yang baru akan disimpan, jika data yang dimasukkan tidak lengkap maka sistem akan menampilkan pesan kesalahan dan menampilkan ulang *form* isian ubah data *order*
5. Menghapus data pemesanan yang diinginkan.		6. Menampilkan pesan konfirmasi apakah data pemesanan akan dihapus
7. Menekan pilihan ”OK” pada pesan hapus data yang diperlihatkan oleh sistem		8. Sistem menghapus data yang diinginkan
9. menekan pilihan ”Cancel” pada pesan hapus data yang diperlihatkan oleh sistem		10. Sistem tidak melakukan aktivitas penghapusan data pemesanan yang diinginkan
11. Mencari data pemesanan yang diinginkan dengan memasukan data pemesanan.		12. Sistem akan menampilkan data pemesanan yang dicari.
13. Memilih menu pilihan cetak label dari data pemesanan yang ingin dicetak		14. Sistem akan menampilkan label yang siap dicetak
Kondisi Akhir	*Admin,* petugas berhasil melakukan kegiatan mengubah, mencari, menghapus data pemesanan dan cetak faktur pemesanan	

Sequence diagram pengolahan pemesanan dapat dilihat pada gambar 3.11



Gambar 3.11 *Sequence diagram* pengolahan pemesanan

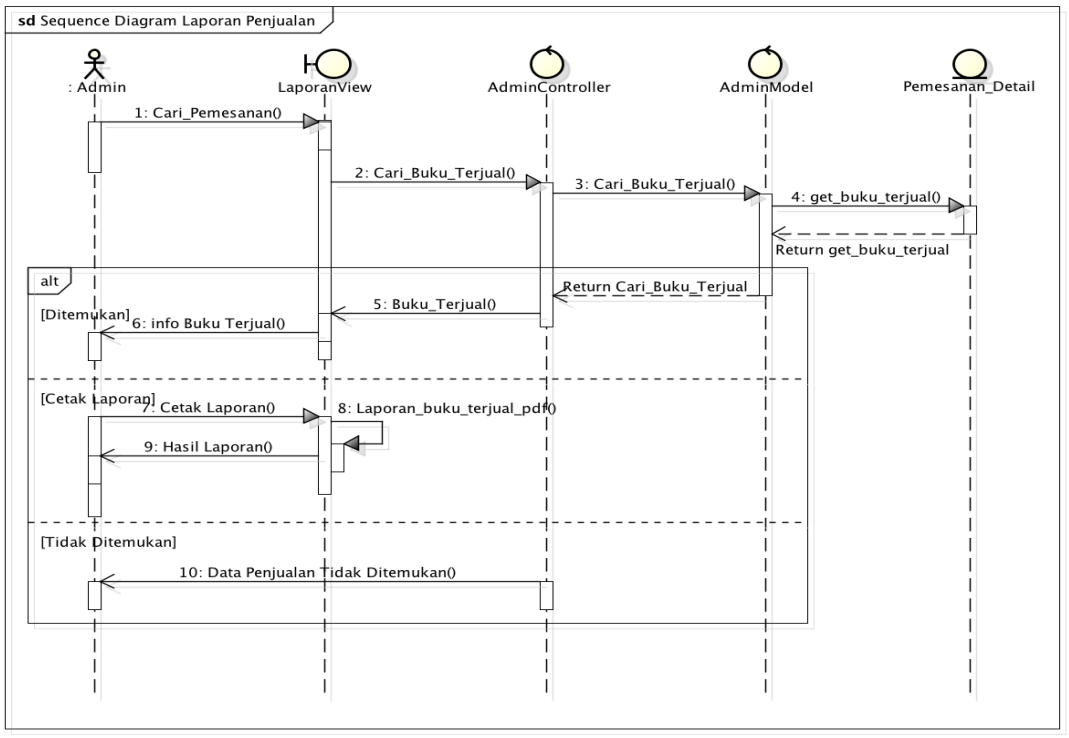
7. Laporan Penjualan

Interaksi antara aktor dengan *use case* laporan penjualan dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.12

Tabel 3.12 Skenario *use case* laporan penjualan

Nama *Use Case*		Laporan Penjualan
Aktor		*Admin,* petugas
Tujuan		menampilkan laporan penjualan
Deskripsi		*Admin,* petugas dapat melakukan kegiatan mencari dan cetak laporan penjualan
Keadaan Awal		Admin, petugas sedang berada pada halaman laporan penjualan
Aktor		Sistem
1. Mencari data penjualan yang diinginkan pada periode tertentu dengan memasukan data penjualan seperti jangka waktu periode yang diinginkan		2. Menampilkan data penjualan yang dicari
3. Memilih tombol cetak laporan		4. Sistem akan mencetak laporan sesuai dengan periode yang dipilih oleh *admin*
Kondisi Akhir	*Admin* berhasil melakukan kegiatan mencari dan mencetak laporan penjualan	

Sequence diagram laporan penjualan dapat dilihat pada gambar 3.12



Gambar 3.12 *Sequence diagram* laporan penjualan

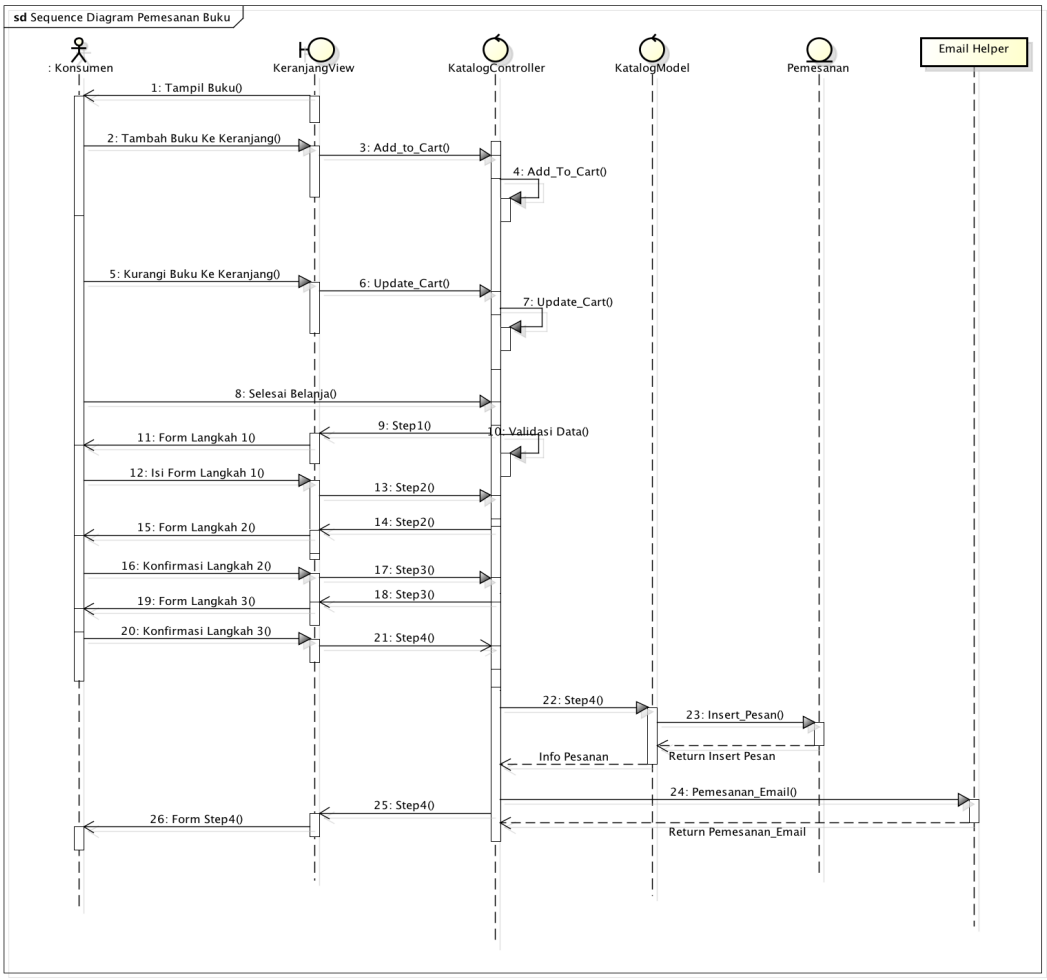
8. Pemesanan Buku

Interaksi antara aktor dengan *use case* pemesanan buku dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.13

Tabel 3.13 Skenario *use case* pemesanan buku

Nama *Use Case*		Pemesanan Buku
Aktor		Konsumen
Tujuan		Konsumen bisa melakukan pembelian
Deskripsi		Konsumen bisa membeli buku
Keadaan Awal		Konsumen membuka halaman buku
Aktor		Sistem
1. Konsumen memilih buku yang akan dibeli		2. Sistem merespon dengan cara mengambil data buku pada *database*, kemudian ditampilkan kepada konsumen
3. Menekan tombol beli pada buku yang diinginkan		4. Sistem merespon dengan menambah data barang yang diinginkan pengguna pada data keranjang (*cart*), melakukan perhitungan jumlah barang dalam keranjang, melakukan perhitungan total harga dalam keranjang
5. Menandai pada buku yang yang akan diubah dan dihapus didalam keranjang belanja		6. Mengurangi data buku yang ada pada keranjang belanja, melakukan perhitungan jumlah buku, total harga dan menampilkan keranjang belanja yang telah diubah.
7. Menekan tombol selesai belanja dalam keranjang belanja		8. Menampilkan halaman pengisian data pemesanan
9. Mengisi data pengiriman yang terdiri dari alamat pengirim dan penerima beserta cara pembayaran		10. mengecek data alamat, jika tidak lengkap maka akan menampilkan pesan kesalahan, jika lengkap maka akan menampilkan rangkuman pemesanan
11. Menekan tombol proses		12. Sistem akan mengarahkan ke halaman langkah 2 yaitu detail pemesanan yang berisi data buku yang dibeli dan alamat pemesan dan pengirim.
13. Menekan tombol lanjutkan		14. Sistem akan mengarahkan ke halaman langkah 3 yang berisi rangkuman pemesanan
15. Menekan tombol kirim pemesanan		16. Sistem akan menyimpan data pemesanan dan mengirimkan email konfirmasi kepada konsumen
Kondisi Akhir	Konsumen berhasil melakukan proses belanja *online*	
Skenario Alternatif – Proses Pemesanan Gagal		
Deskripsi	Jika *form* yang diisi oleh konsumen datanya tidak valid, maka sistem akan kembali menampilkan *form* pengiriman dan memberi pesan kepada kosumen bahwa data yang di masukkan tidak *valid*	

Sequence diagram dari pemesanan buku dapat dilihat pada gambar 3.13



Gambar 3.13 *Sequence diagram* pemesanan buku

9. Pengaturan Akun

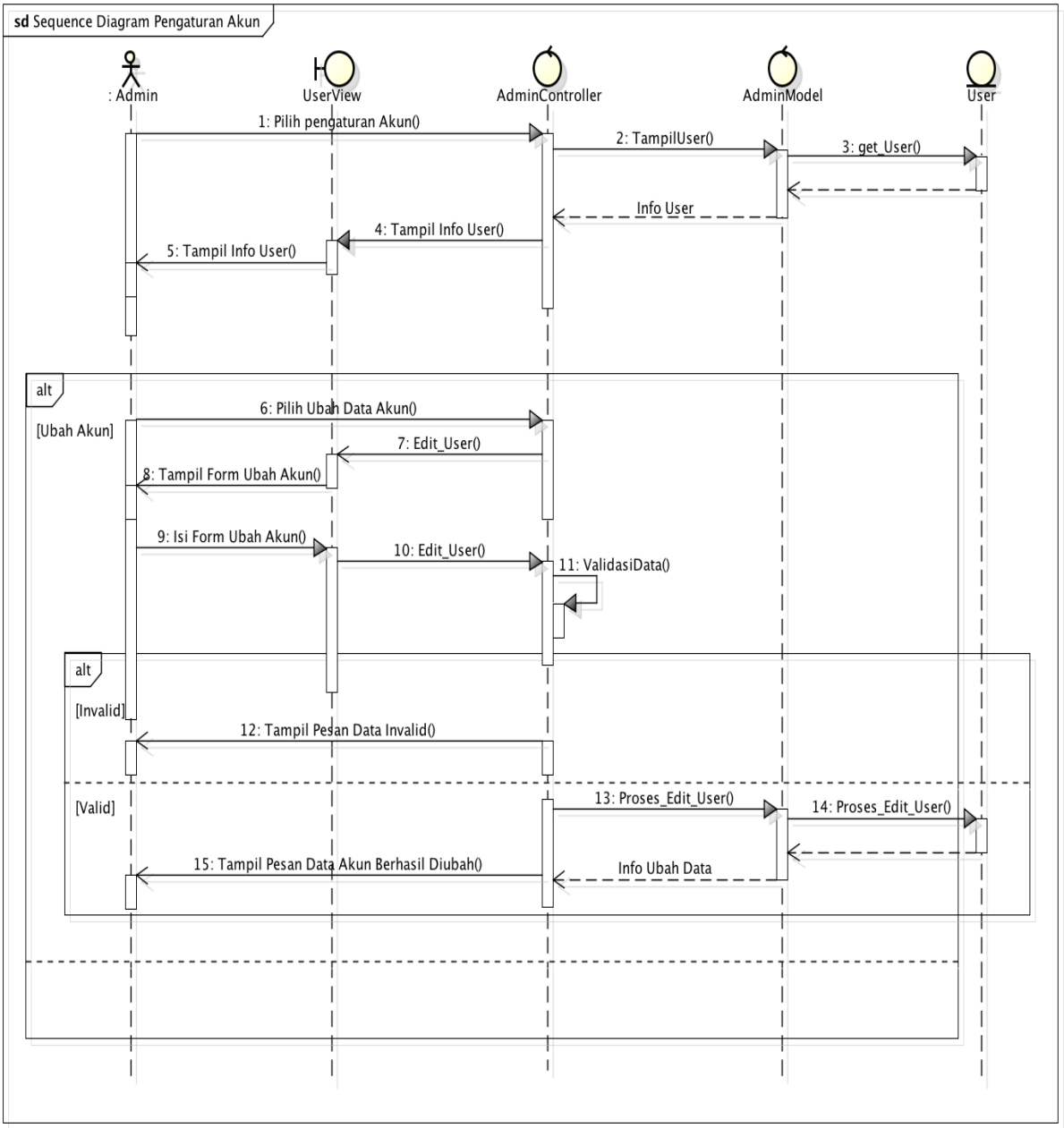
Interaksi antara aktor dengan *use case* pengolahan akun dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.14

Tabel 3.14 Skenario *use case* pengolahan akun

Nama *Use Case*	Pengaturan Akun
Aktor	*Admin* / petugas
Tujuan	mengatur akun *admin*/ petugas
Deskripsi	*Proses* untuk mengatur data akun *admin*/ petugas
Keadaan Awal	*admin*/ petugas membuka halaman utama *admin*
Aktor	Sistem
1. memilih menu akun saya	2. Menampilkan form pengolahan data akun
3. Mengisi data ubah akun	4. Mengecek data masukan, jika data tidak lengkap maka akan tampil pesan kesalahan, jika data lengkap maka data akan disimpan pada *database*

5. Memilih Ubah *Password*	6. Menampilkan form ubah *password*
7. Mengisi form ubah password	8. Mengecek data masukan, jika data tidak sesuai maka akan tampil pesan kesalahan, jika data lengkap maka data akan disimpan pada *database*
Kondisi Akhir	*admin*/ petugas berhasil mengolah data akun

Sequence diagram dari proses pengolahan data akun dapat dilihat pada gambar 3.14



Gambar 3.14 *Sequence diagram* pengaturan akun

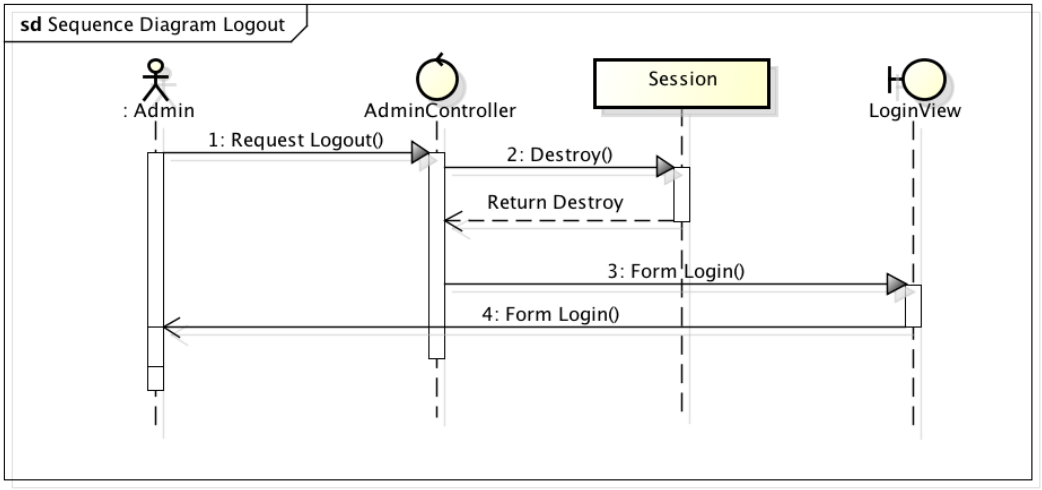
10. *Logout*

Interaksi antara aktor dengan *use case logout* dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.15

Tabel 3.15 Skenario *use case logout*

Nama *Use Case*		Logout (*admin*)
Aktor		*Admin,* Petugas
Tujuan		Mengakhiri sesi dan keluar dari sistem
Deskripsi		*Proses* untuk keluar dari sistem
Keadaan Awal		*Admin,* petugas membuka halaman utama *admin*
Aktor		Sistem
4. *Admin* memilih menu *logout*		5. Sistem menghentikan sesi *admin* dan mengarahkan ke halaman *login admin*
Kondisi Akhir	*Admin,* petugas berhasil *logout* dari sistem	

Sequence diagram dari proses *logout* dapat dilihat pada gambar 3.15



Gambar 3.15 *Sequence* diagram *logout*

11. Cari Buku

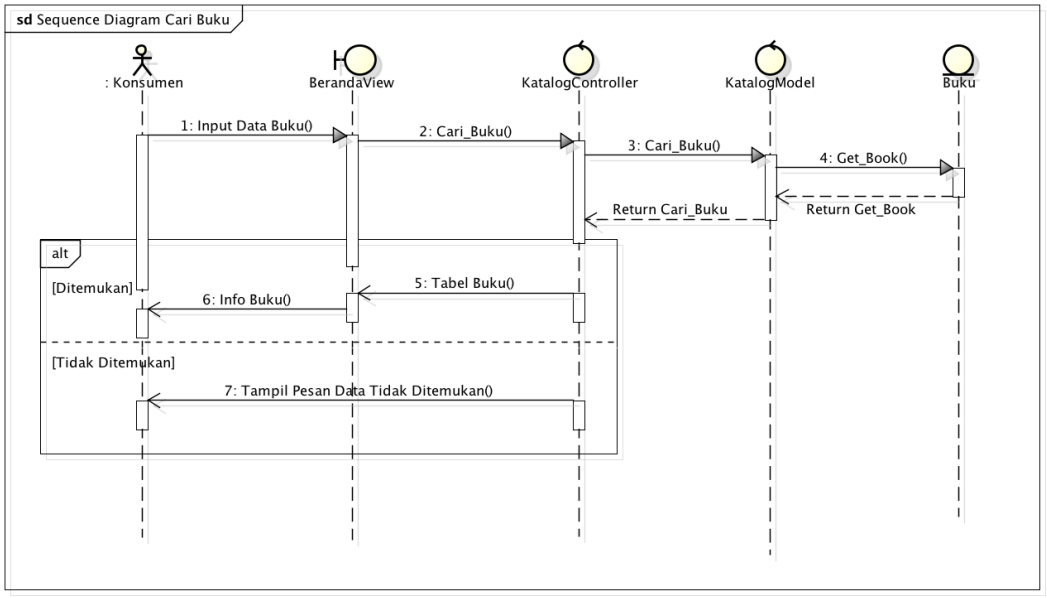
Interaksi antara aktor dengan *use case* cari buku dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.16

Tabel 3.16 Skenario *use case* cari buku

Nama *Use Case*	Cari Buku
Aktor	Konsumen
Tujuan	Mencari data buku
Deskripsi	*Proses* untuk mencari dan menampilkan data buku

Keadaan Awal		Menampilkan halaman beranda
Aktor		Sistem
1. Memasukan data buku yang akan dicari		2. Mencari data buku pada *database*, jika data ditemukan maka akan menampilkan list data produk, jika data tidak ditemukan maka akan menampilkan pesan data tidak ditemukan
Kondisi Akhir	Buku yang dicari berhasil ditampilkan	

Sequence diagram dari proses cari buku dapat dilihat pada gambar 3.16



Gambar 3.16 *Sequence* diagram cari buku

12. Retur

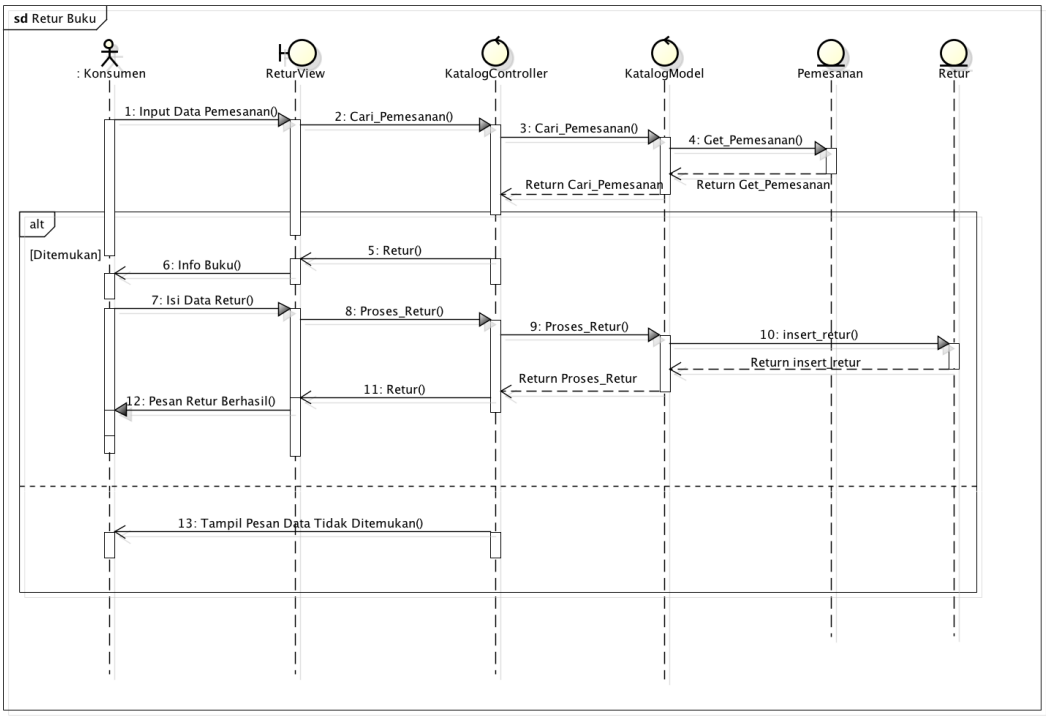
Interaksi antara aktor dengan *use case* retur dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.17

Tabel 3.17 Skenario *use case* retur

Nama *Use Case*	Retur
Aktor	Konsumen
Tujuan	Melakukan retur buku
Deskripsi	*Proses* untuk mengembalikan buku yang dibeli dengan mengisi data buku yang diretur
Keadaan Awal	Konsumen berada pada halaman retur
Aktor	Sistem
1. Memasukan nomor faktur dan nomor resi pengiriman	2. Mencari data buku pada *database*, jika data ditemukan maka akan menampilkan list data buku yang dibeli oleh konsumen, dan apabila nomor resi dan nomor faktur tidak cocok maka akan menampilkan pesan bahwa nomor resi dan kode

	faktur tidak cocok
3. Memasukan jumlah buku yang akan di retur beserta alasan retur	4. Sistem menyimpan data retur ke *database,* kemudian sistem akan menampilkan pesan bahwa data retur akan segera di proses.
Kondisi Akhir	Buku yang dicari berhasil ditampilkan

Sequence diagram dari proses retur buku dapat dilihat pada gambar 3.17



Gambar 3.17 *Sequence* diagram Retur

13. Pengolahan Data Retur

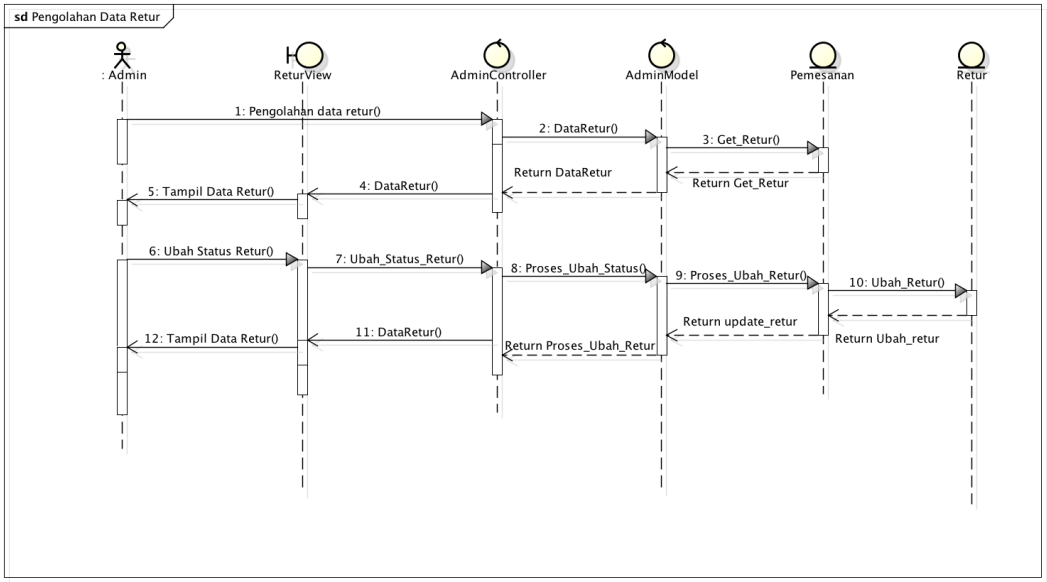
Interaksi antara aktor dengan *use case* pengolahan data retur dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.18

Tabel 3.18 Skenario *use case* pengolahan data retur

Nama *Use Case*	Pengolahan data retur
Aktor	*Admin*, petugas
Tujuan	Mengubah status retur barang
Deskripsi	*Proses* untuk mengubah status retur barang
Keadaan Awal	*Admin,* petugas sedang berada pada halaman pengolahan data retur
Aktor	Sistem
1. Melihat data retur buku, mengecek alasan retur dan mengubah status retur apakah diterima atau ditolak	2. Sistem menyimpan data retur kedalam *database*

Kondisi Akhir	Data retur berhasil disimpan

Sequence diagram dari proses cari buku dapat dilihat pada gambar 3.18



Gambar 3.18 *Sequence* diagram pengolahan data retur

14. Pengolahan data subkategori

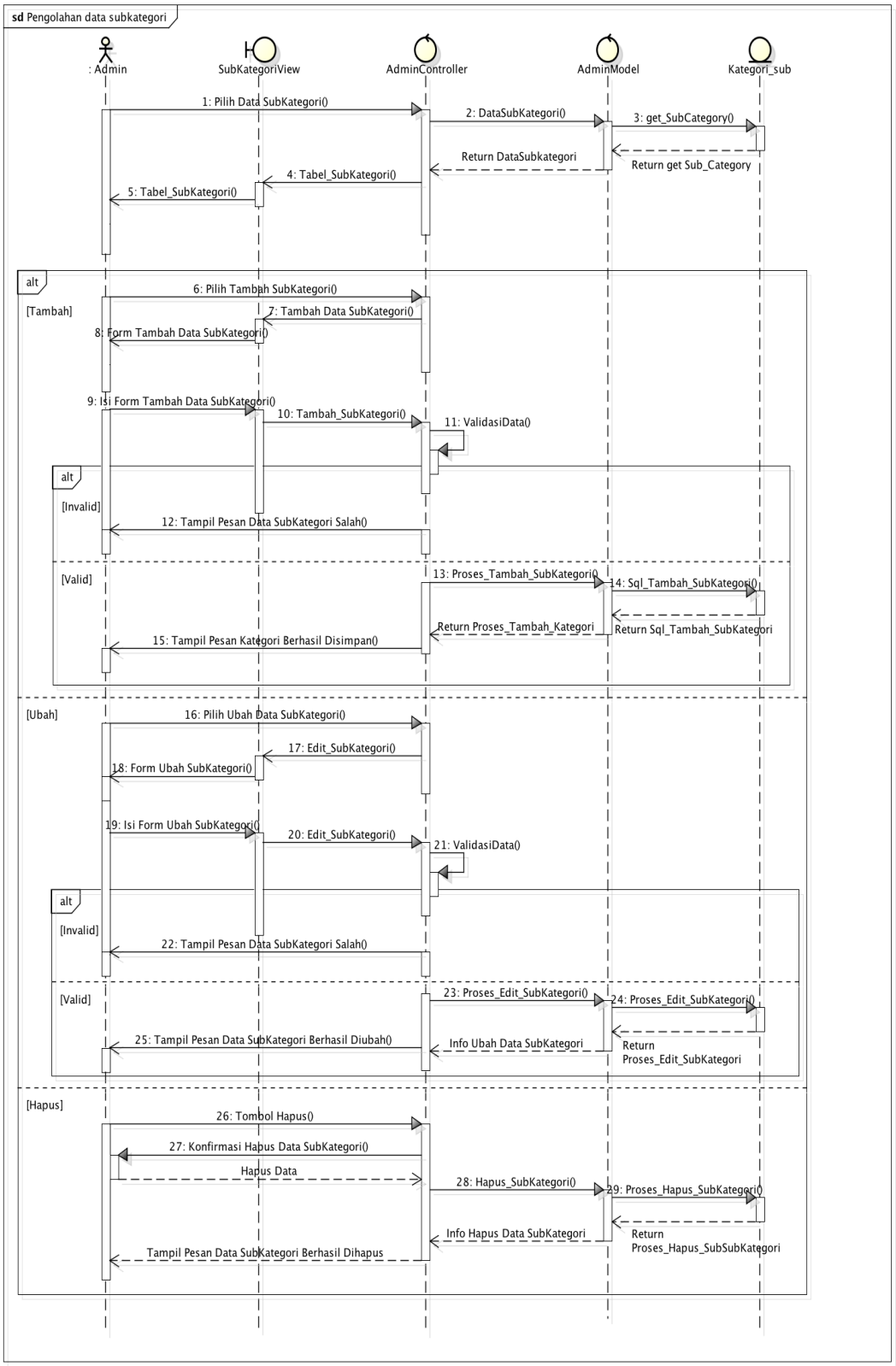
Interaksi antara aktor dengan *use case* pengolahan data subkategori buku dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.19

Tabel 3.19 Skenario *use case* pengolahan data kategori

Nama *Use Case*	Pengolahan Data Subkategori
Aktor	*Admin,* petugas
Tujuan	Menampilkan daftar subkategori buku dan menampilkan menu utama *administrator* untuk mengolah data subkategori
Deskripsi	Pengguna sebagai *admin,* petugas dapat melakukan kegiatan menambah, mengubah, dan menghapus data subkategori buku
Keadaan Awal	Admin, petugas sedang berada di menu data subkategori
Aktor	Sistem
1. Memilih menu tambah subkategori	2. Menampilkan form tambah subkategori
3. Mengisi form tambah subkategori dan menekan tombol simpan	4. Memeriksa data yang dimasukkan. Jika data yang dimasukkan lengkap, maka data akan disimpan ke *database* dan sistem akan mengarahkan ke halaman subkategori, tetapi jika data yang dimasukkan tidak lengkap, maka sistem akan menampilkan ulang *form* tambah subkategori dan menampilkan pesan kesalahan

5. Memilih menu ubah data subkategori yang diinginkan	6. Sistem menampilkan *form* isian data subkategori lengkap dengan data yang akan diubah.
7. Mengubah data subkategori pada *form* ubah dan menekan tombol simpan	8. Sistem akan memeriksa data yang dimasukan. jika data yang dimasukkan lengkap, maka data yang baru akan disimpan, jika data yang dimasukkan tidak lengkap, maka sistem akan menampilkan pesan kesalahan dan menampilkan ulang *form* isian ubah data subkategori
9. Memilih subkategori yang akan dihapus.	10. Menampilkan pesan konfiramsi apakah subkategori akan dihapus
11. *Admin* menekan pilihan ”OK” pada pesan hapus data yang diperlihatkan oleh sistem	12. Sistem menghapus data subkategori yang diinginkan
13. *Admin* menekan pilihan ”Cancel” pada pesan hapus data yang diperlihatkan oleh sistem	14. Sistem tidak melakukan melakukan penghapusan kategori
Kondisi Akhir	*Admin* berhasil melakukan kegiatan menambah, mengubah, dan menghapus data subkategori buku

Sequence diagram pengolahan data subkategori dapat dilihat pada gambar 3.19



Gambar 3.19 *Sequence diagram* pengolahan data subkategori

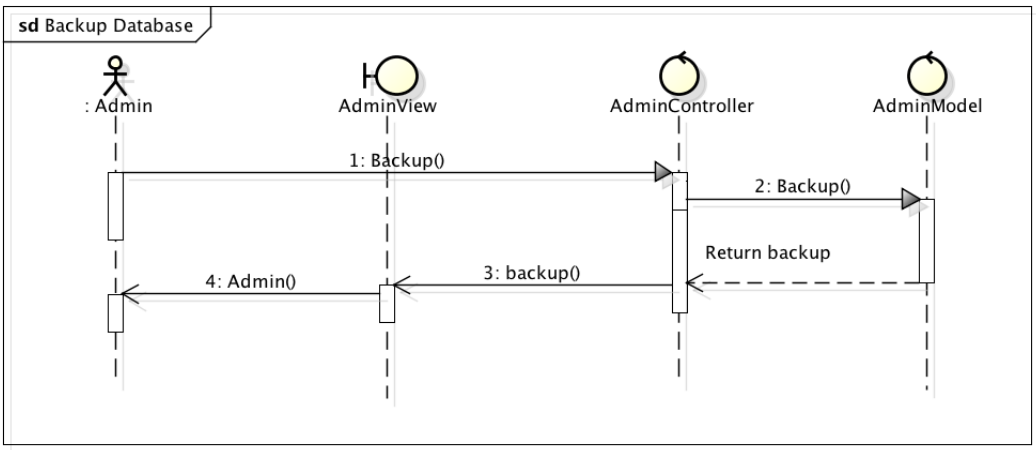
15. *Backup Database*

Interaksi antara aktor dengan *use case backup database* dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.20

Tabel 3.20 Skenario *use case backup database*

Nama *Use Case*		Pengolahan data retur
Aktor		*Admin*, petugas
Tujuan		*Backup database*
Deskripsi		*Proses* untuk backup *database* sistem
Keadaan Awal		*Admin,* petugas sedang berada halaman *admin*
Aktor		Sistem
1. Memilih menu *backup database*		2. Sistem mengambil semua tabel yang ada di database sistem.
Kondisi Akhir	Tabel sudah di backup dalam bentuk *file*	

Sequence diagram dari proses backup *database* dilihat pada gambar 3.20



Gambar 3.20 *Sequence* diagram *backup database*

16. Konfirmasi Pembayaran

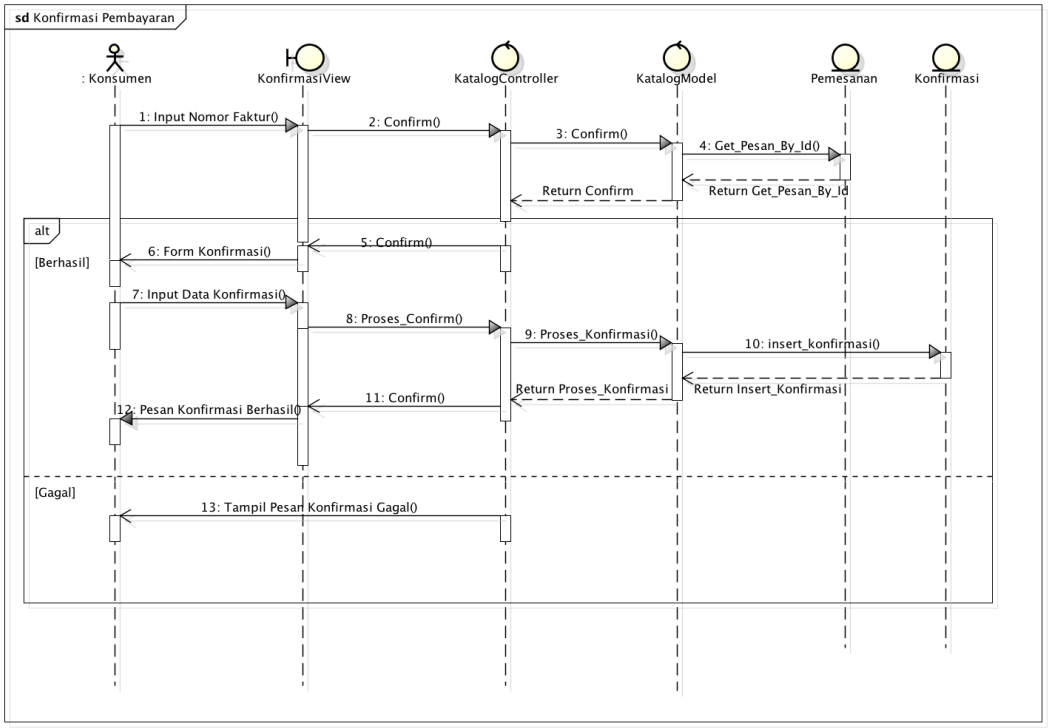
Interaksi antara aktor dengan konfirmasi pembayaran dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.21

Tabel 3.21 Skenario *use case* konfirmasi pembayaran

Nama *Use Case*	Konfirmasi pembayaran
Aktor	Konsumen
Tujuan	Konfirmasi Pembayaran
Deskripsi	*Proses* untuk konfirmasi pembayaran
Keadaan Awal	Konsumen sedang berada pada halaman konfirmasi

	pembayaran
Aktor	Sistem
1. Memilih menu konfirmasi pembayaran	2. Sistem mengambil data pemesanan sesuai dengan nomor faktur yang dimasukan oleh konsumen untuk ditampilkan.
3. Mengisi form konfirmasi pembayaran	4. Menyimpan data konfirmasi pembayaran ke dalam *database*
Skenario alternatif – Data Konfirmasi Sudah ada	
1. Mengisi form konfirmasi pembayaran	2. Sistem menampilkan pesan data konfirmasi sudah ada
Kondisi Akhir	Tabel sudah di backup dalam bentuk *file*

Sequence diagram dari konfirmasi pembayaran dilihat pada gambar 3.21



Gambar 3.21 *Sequence* diagram konfirmasi pembayaran

17. Kontak

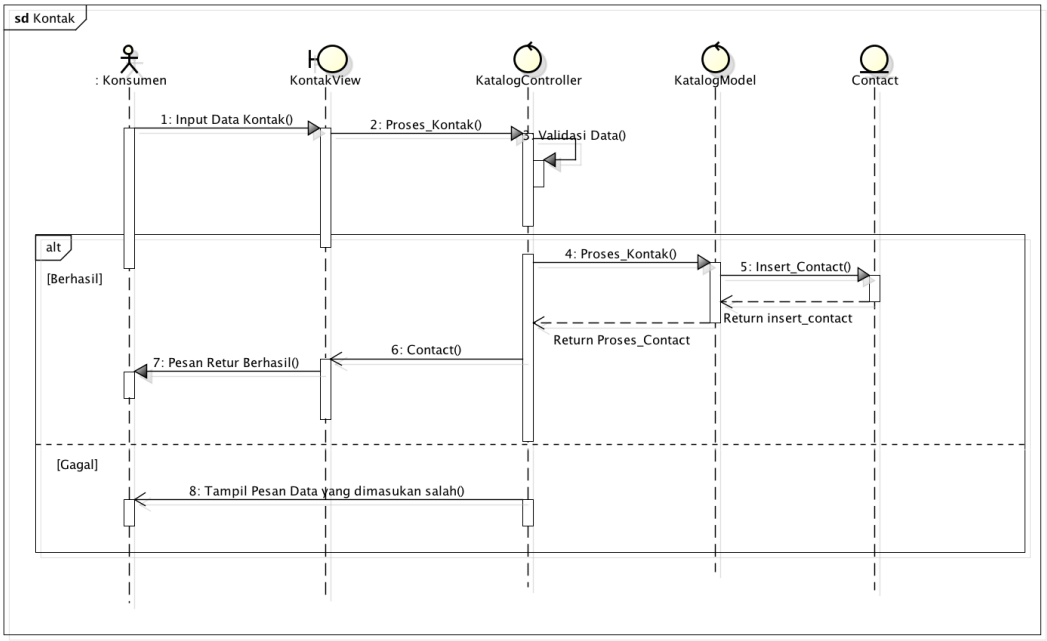
Interaksi antara aktor dengan *use case* kontak dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.22

Tabel 3.22 Skenario *use case backup database*

Nama *Use Case*	Kontak
Aktor	Konsumen
Tujuan	Mengirimkan pertanyaan kepada admin

Deskripsi		*Proses* untuk mengirimkan pertanyaan kepada admin
Keadaan Awal		Konsumen berada pada halaman kontak
Aktor		Sistem
1. Memasukan data kontak		2. Sistem Memeriksa apakah data yang dimasukan benar, apabla benar maka sistem akan menyimpan data kontak ke *database,* apabila salah maka sistem akan menampilkan pesan kesalahan.
Kondisi Akhir	Konsumen berhasil melakukan proses kontak	

Sequence diagram proses kontak dapat dilihat pada gambar 3.22



Gambar 3.22 *Sequence* diagram kontak

18. Balas Kontak

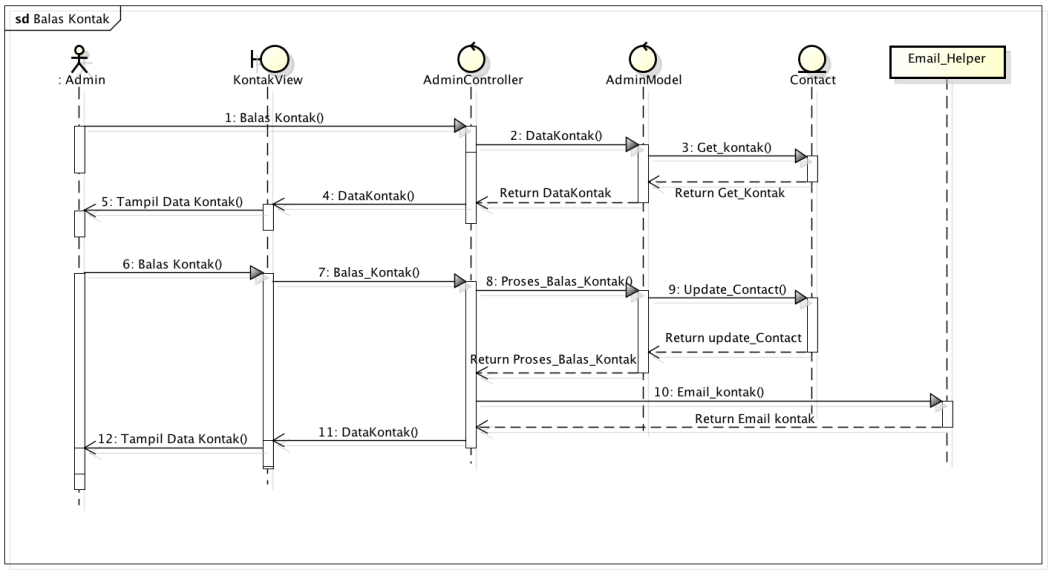
Interaksi antara aktor dengan *use case* kontak dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.23

Tabel 3.23 Skenario *use case backup database*

Nama *Use Case*	Balas Kontak
Aktor	Admin, petugas
Tujuan	Admin, petugas membalas pertanyaan yang diberikan oleh konsumen
Deskripsi	*Proses* untuk mengirimkan pertanyaan kepada admin
Keadaan Awal	Konsumen berada pada halaman kontak
Aktor	Sistem
1. Memilih menu balas kontak	2. Sistem mengambil data kontak dari *database* untuk ditampilkan.

3. Membalas kontak	4. Sistem menyimpan data kontak dan mengirimkan email balasan kepada konsumen
Kondisi Akhir	Admin, petugas berhasil membalas kontak email dari konsumen

Sequence diagram balas kontak dapat dilihat pada gambar 3.23



Gambar 3.23 *Sequence* diagram balas kontak

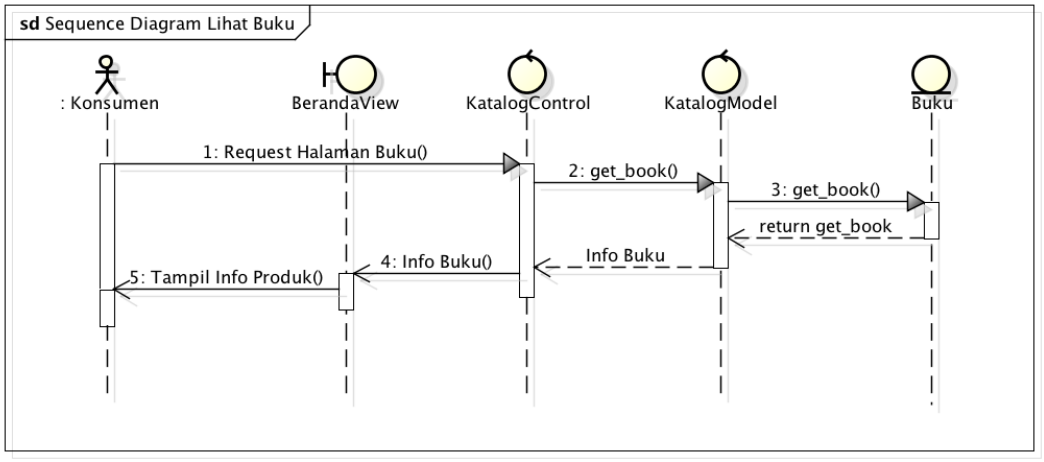
19. Lihat Buku

Interaksi antara aktor dengan *use case* lihat buku dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.24

Tabel 3.24 Skenario *use case* lihat buku

Nama *Use Case*	Lihat Buku
Aktor	Konsumen
Tujuan	Konsumen melihat data buku
Deskripsi	*Proses* untuk melihat data buku
Keadaan Awal	Konsumen berapa pada halaman katalog
Aktor	Sistem
1. Melihat buku pada katalog	2. Sistem mengambil data buku pada database untuk ditampilkan
Kondisi Akhir	Halaman buku berhasil ditampilkan

Sequence diagram lihat buku dapat dilihat pada gambar 3.24



Gambar 3.24 *Sequence* diagram balas kontak

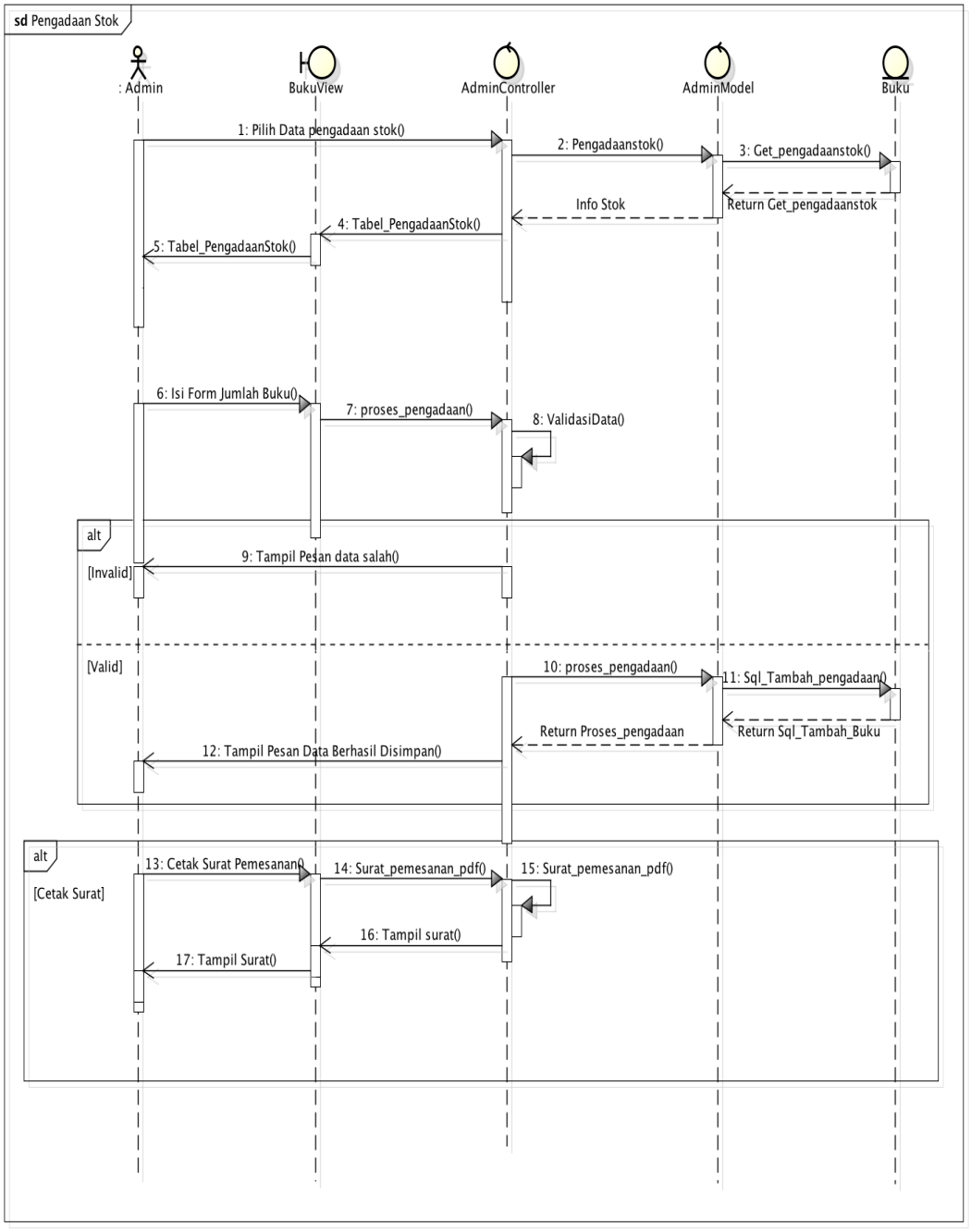
20. Pengadaan Stok

Interaksi antara aktor dengan *use case* pengadaan stok dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.25

Tabel 3.25 Skenario *use case* pengadaan stok

Nama *Use Case*	Pengadaan Stok
Aktor	Admin, Petugas
Tujuan	Admin, petugas dapat menambah stok buku
Deskripsi	*Proses* untuk menambah stok buku dan memesan ke *supplier*
Keadaan Awal	Admin, petugas sedang berada pada halaman pengadaan stok
Aktor	Sistem
1. Memilih menu pengadaan stok	2. Sistem mengambil data buku pada database yang jumlahnya kurang dari stok minimal untuk ditambah jumlahnya dan mencetak surat pemesanan
3. Memasukan jumlah buku yang akan dipesan	4. Sistem menyimpan data pengadaan buku dan menyimpan ke dalam database
5. Memilih menu cetak surat pemesanan	6. Sistem akan mencetak surat pemesanan sesuai dengan id surat yang diinginkan
Kondisi Akhir	Admin, petugas berhasil melakukan pemesanan stok ke *supplier*

Sequence diagram pengadaan stok dapat dilihat pada gambar 3.25



Gambar 3.25 *Sequence* diagram pengadaan stok

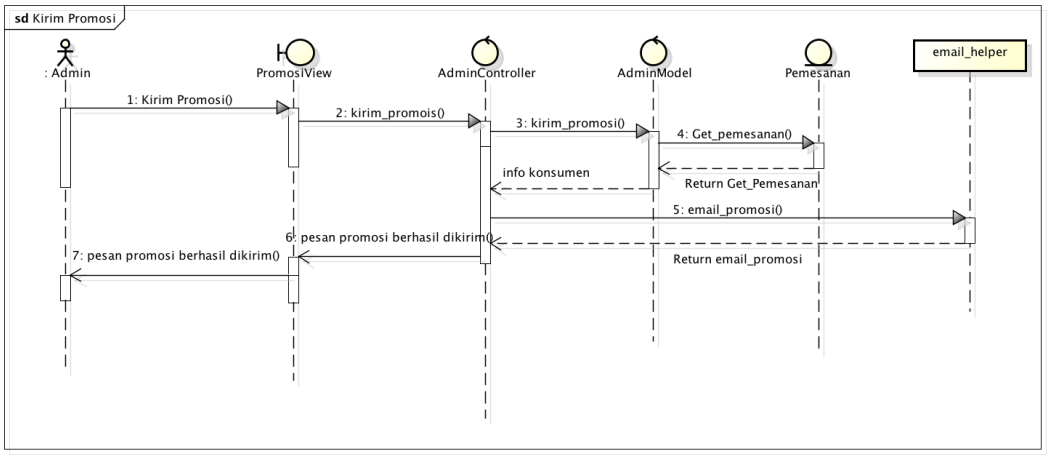
21. Promosi

Interaksi antara aktor dengan *use case* promosi dijelaskan dalam *use case scenario* yang dapat dilihat pada tabel 3.26

Tabel 3.26 Skenario *use case* promosi

Nama *Use Case*		Promosi
Aktor		Admin, Petugas
Tujuan		Admin, petugas dapat mempromosika produk terbaru
Deskripsi		*Proses* untuk mengirimkan promosi buku terbaru kepada konsumen
Keadaan Awal		Admin, petugas sedang berada pada halaman promosi
Aktor		Sistem
1. Memasukan gambar promosi yang akan dikirimkan ke konsumen		2. Sistem akan mengecek ke database konsumen yang sudah 5 kali membeli yang akan mendapat email promosi dari admin
Kondisi Akhir	Admin, petugas berhasil mengirimkan email promosi kepada konsumen	

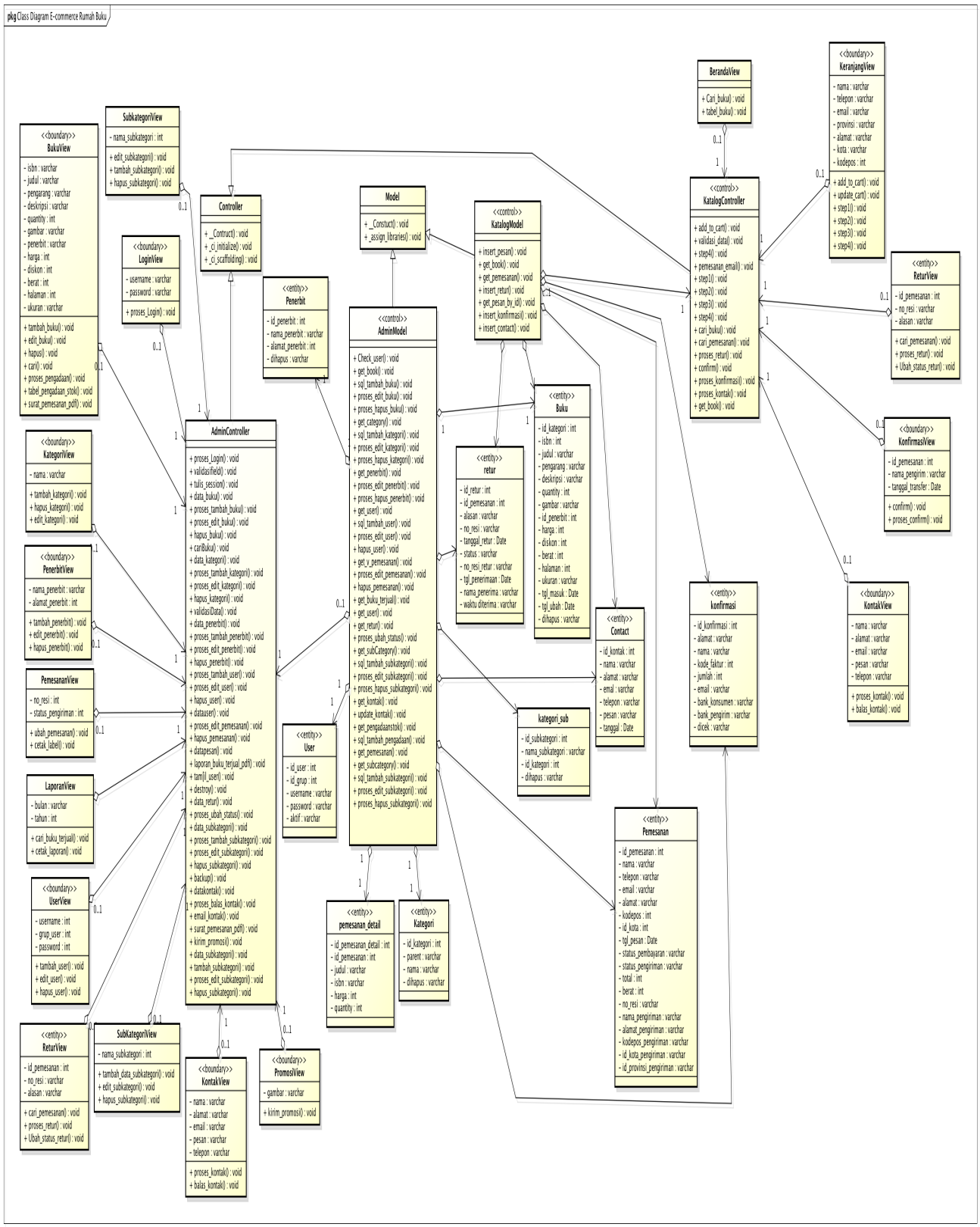
Sequence diagram kirim promosi dapat dilihat pada gambar 3.26



Gambar 3.26 *Sequence* diagram kirim promosi

3.1.8.6 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (fungsi). *Class* diagram dari *e-commerce* pada Rumah Buku dapat dilihat pada gambar 3.27



Gambar 3.27 *Class Diagram* aplikasi *e-commerce* Rumah Buku Bandung

Setiap kelas pada gambar 3.27 memiliki jenis kelasnya masing-masing seperti yang terlihat pada tabel 3.27

Tabel 3.27 Jenis class diagram aplikasi *e-commerce* pada toko Rumah Buku Bandung

Nama	Jenis
AdminController	***Control***
KatalogController	
KatalogModel	
AdminModel	
KategoriView	***Boundary***
BukuView	
LaporanView	
PemesananView	
LaporanView	
UserView	
ReturView	
LoginView	
PenerbitView	
KontakView	
SubKategoriView	
PromosiView	
KonfirmasiView	
BerandaView	
Kategori	***Entity***
Buku	
Pemesanan	
Penerbit	
User	
pemesanan_detail	
Kategori	
Retur	
Contact	
Konfirmasi	
kategori_sub	

Deskripsi dari setiap jenis *class* yang ada pada tabel 3.27 dapat dilihat pada penjelasan berikut ini :

1. *Class* AdminController

Class AdminController merupakan *class* yang digunakan untuk mengatur semua proses pengolahan data pada bagian *Admin*(*back end*) dimana nilai variabel dari *class view* akan terlebih dahulu di cek pada *class control* sebelum akhirnya masuk ke *class model* untuk diolah.

a. Deskripsi Atribut

Class AdminController tidak mempunyai atribut.

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada *class* AdminController dapat dilihat pada tabel 3.28

Tabel 3.28 Operasi pada *class* AdminController

Nama	Keterangan
ValidasiData() : void	Prosedur yang dipanggil untuk memvalidasi data pesanan yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
backup() : void	Prosedur yang dipanggil untuk backup semua tabel yang ada di sitem. Prosedur ini akan memamnggil fungsi backup pada framework untuk *backup database*.
Kategori	
proses_tambah_kategori() : void	Prosedur untuk melakukan penambahan data kategori dan akan memanggil class AdminModel untuk melakukan proses tambah data kategori dan menyimpannya pada *database*.
DataKategori() : void	Prosedur yang dipanggil ketika *user* meminta untuk menampilkan data kategori, prosedur ini akan memberi perintah kepada *class* AdminModel untuk mengambil data kategori pada *database*.
proses_edit_Kategori() : void	Prosedur untuk melakukan perubahan data kategori, akan memanggil class AdminModel untuk mengambil data, setelah terisi akan memanggil kembali KategoriModel untuk menyimpan data yang sudah diubah.
Hapus_Kategori() : void	Prosedur untuk menghapus data kategori, akan menampilkan pesan konfirmasi apakah data kategori akan dihapus, jika ya maka akan memanggil AdminModel untuk melakukan penghapusan data pada *database*.
Buku	
proses_tambah_buku() : void	Prosedur untuk melakukan penambahan data buku dan akan memanggil class AdminModel untuk melakukan proses tambah data buku dan menyimpannya pada *database*.
DataBuku() : void	Prosedur yang dipanggil ketika *user* meminta untuk menampilkan data buku, prosedur ini akan memberi perintah kepada *class* AdminModel untuk mengambil data buku pada *database*.
proses_edit_Buku() : void	Prosedur untuk melakukan perubahan data buku, akan memanggil class AdminModel untuk mengambil data, setelah terisi akan

	memanggil kembali AdminModel untuk menyimpan data yang sudah diubah.
Hapus_Buku() : void	Prosedur untuk menghapus data buku, akan menampilkan pesan konfirmasi apakah data buku akan dihapus, jika ya maka akan memanggil AdminModel untuk melakukan penghapusan data pada *database*.
Cari_Buku() : void	Prosedur yang dipanggil untuk menampilkan data buku sesuai dengan kata kunci sql yang diberikan, prosedur ini akan memberi perintah kepada *class* BukuModel untuk mengambil data pada *database*.
Pemesanan	
proses_edit_Pemesanan() : void	Prosedur untuk melakukan perubahan data pesanan, akan memanggil *class* AdminModel untuk mengambil data, setelah terisi akan memanggil kembali AdminModel untuk menyimpan data yang sudah diubah.
DataPesan() : void	Prosedur yang dipanggil ketika *user* meminta untuk menampilkan data pemesanan, prosedur ini akan memberi perintah kepada *class* AdminModel untuk mengambil data pemesanan pada *database*.
Hapus_Pemesanan() : void	Prosedur untuk menghapus data pesanan, akan menampilkan pesan konfirmasi apakah data pesanan akan dihapus, jika ya maka akan memanggil AdminModel untuk melakukan penghapusan data pada database.
Penerbit	
proses_tambah_penerbit () : void	Prosedur untuk melakukan penambahan data penerbit dan akan memanggil *class* AdminModel untuk melakukan proses tambah data penerbit dan menyimpannya pada *database*.
DataKategori() : void	Prosedur yang dipanggil ketika *user* meminta untuk menampilkan data penerbit, prosedur ini akan memberi perintah kepada *class* AdminModel untuk mengambil data penerbit pada *database*.
proses_edit_penerbit () : void	Prosedur untuk melakukan perubahan data penerbit, akan memanggil class AdminModel untuk mengambil data, setelah terisi akan memanggil kembali AdminModel untuk menyimpan data yang sudah diubah.
Hapus_Penerbit () : void	Prosedur untuk menghapus data penerbit, akan menampilkan pesan konfirmasi apakah data penerbit akan dihapus, jika ya maka akan memanggil AdminModel untuk melakukan penghapusan data pada *database*.
User	
DataUser() : void	Prosedur yang dipanggil ketika *user* meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* UserModel untuk mengambil data pada *database*.
Proses_Tambah_User() : void	Prosedur untuk melakukan penambahan data *user* dan akan memanggil class AdminModel untuk melakukan proses tambah data *user* dan menyimpannya pada *database*.
Proses_edit_User() : void	Prosedur untuk melakukan perubahan data *user*, akan memanggil class AdminModel untuk mengambil data, setelah terisi akan memanggil kembali AdminModel untuk menyimpan data yang sudah diubah.
ValidasiDataUser() : void	Prosedur yang dipanggil untuk memvalidasi data *user* yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
Hapus_User() : void	Prosedur untuk menghapus data *user*, akan menampilkan pesan konfirmasi apakah data user akan dihapus, jika ya maka akan memanggil AdminModel untuk melakukan penghapusan data pada database.

Login	
Proses_Login() : void	Prosedur yang dipanggil ketika *user* melakukan proses *login*, memvalidasi dan memanggil class AdminModel yang untuk mencari data *user* yang ada pada *database*.
ValidasiField () : void	Prosedur yang dipanggil untuk memvalidasi data yang dimasukan pada field.
Logout	
Destroy() : void	Prosedur untuk menghapus sesi yang ada untuk admin, petugas yang akan keluar dari sistem.
Laporan	
Cari_Buku_Terjual() : void	Prosedur untuk mencari laporan yang dicari dan memanggil class AdminModel untuk mengambil data buku terjual yang dicari pada *database*.
laporan_buku_terjual_pdf : void	Prosedur yang memanggil AdminModel untuk menampilkan laporan hasil pencarian dan mencetak dalam bentuk pdf.
Retur	
DataRetur() : void	Prosedur yang dipanggil ketika *user* meminta untuk menampilkan data retur, prosedur ini akan memberi perintah kepada *class* AdminModel untuk mengambil data retur pada *database*.
proses_ubah_status	Prosedur yang memanggil AdminModel untuk mengubah status retur dan menyimpannya pada *database*.
Subkategori	
Data_Subkategori() : void	Prosedur yang dipanggil ketika *user* meminta untuk menampilkan data subkategori, prosedur ini akan memberi perintah kepada *class* AdminModel untuk mengambil data subkategori pada *database*.
proses_tambah_subkategori() : void	Prosedur untuk melakukan penambahan data subkategori dan akan memanggil class AdminModel untuk melakukan proses tambah data subkategori dan menyimpannya pada *database*.
proses_edit_ subkategori () : void	Prosedur untuk melakukan perubahan data subkategori, akan memanggil class AdminModel untuk mengambil data, setelah terisi akan memanggil kembali KategoriModel untuk menyimpan data yang sudah diubah.
Hapus_ subkategori () : void	Prosedur untuk menghapus data kategori, akan menampilkan pesan konfirmasi apakah data kategori akan dihapus, jika ya maka akan memanggil AdminModel untuk melakukan penghapusan data subkategori pada *database*.
Kontak	
DataKontak () : void	Prosedur yang dipanggil ketika *user* meminta untuk menampilkan data kontak, prosedur ini akan memberi perintah kepada *class* AdminModel untuk mengambil data kontak pada *database*.
proses_balas_kontak() : void	Prosedur untuk membalas isian kontak dari konsumen dan prosedur ini akan memanggil AdminModel untuk menyimpan data balasan ke dalam *database*.
email_kontak(): void	Berisi email balasan yang akan dikirimkan kepada konsumen.
surat_pemesanan_pdf	Prosedur yang memanggil AdminModel untuk mencetak surat pengadaan barang dan mencetak dalam bentuk pdf.
kirim_promosi	prosedur yang dijalankan ketika admin akan mengirimkan email promosi.

2. *Class* KatalogController

Class KatalogController merupakan *class* yang digunakan untuk mengatur semua proses pengolahan data pada bagian katalog(*front end*) dimana nilai variabel dari *class view* akan terlebih dahulu di cek pada *class control* sebelum akhirnya masuk ke *class model* untuk diolah.

a. Deskripsi Atribut

Class katalog controller tidak memiliki attribut.

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada *class* KatalogController dapat dilihat pada tabel 3.29

Tabel 3.29 Operasi pada *class* KatalogController

Nama	Keterangan
ValidasiData() : void	Prosedur yang dipanggil untuk memvalidasi data pesanan yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
get_book : void	Prosedur ini akan memberi perintah *class* KatalogModel untuk menampilkan data buku dari database dan menampilkanya.
add_to_cart() : void	Prosedur yang dipanggil ketika *user* meminta untuk memasukan data buku ke keranjang, prosedur ini akan memberi perintah *class* katalogController untuk menyimpan data buku ke keranjang.
update_cart() : void	Prosedur yang dipanggil ketika *user* meminta untuk memasukan data buku ke keranjang, prosedur ini akan memberi perintah *class* katalogController untuk mengubah jumlah buku dan menyimpan data buku ke keranjang.
delete_cart() : void	Prosedur yang dipanggil ketika *user* meminta untuk menghapus data buku ke keranjang, prosedur ini akan memberi perintah *class* katalogController untuk menghapus data buku dari keranjang.
confirm() : void	Prosedur yang dipanggil ketika *user* ingin menampilkan halaman web pada katalog, prosedur ini akan memberi perintah katalogController untuk menampilkan menu konfirmasi pembayaran
step1() : void	Prosedur yang dipanggil ketika *user* ingin menampilkan halaman web pada katalog, prosedur ini akan memberi perintah katalogController untuk menampilkan menu isi data pengiriman.
step2() : void	Prosedur yang dipanggil ketika *user* ingin menampilkan halaman web pada katalog, prosedur ini akan memberi perintah katalogController untuk menampilkan halaman konfirmasi pemesanan.
step3() : void	Prosedur yang dipanggil ketika *user* ingin menampilkan halaman web pada katalog, prosedur ini akan memberi perintah

	katalogController untuk menampilkan halaman rangkuman pemesanan.
step4() : void	Prosedur yang dipanggil ketika *user* ingin menampilkan halaman web pada katalog, prosedur ini akan memberi perintah katalogController untuk menampilkan halaman ucapan terima kasih.
proses_contact	Prosedur yang dipanggil ketika *user* meminta mengirimkan data kontak, prosedur ini akan memberi perintah *class* katalogController memanggil class katalogModel untuk mengimpanya ke *database*.
proses_kofirmasi	Prosedur yang dipanggil ketika *user* meminta mengirimkan data konfirmasi, prosedur ini akan memberi perintah *class* katalogController memanggil class katalogModel untuk mengimpanya ke *database*.

3. *Class* AdminModel

Class AdminModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah semua data yang ada pada aplikasi yang dibangun. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

a. Deskripsi Atribut

Class AdminModel tidak memiliki atribut.

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class AdminModel dapat dilihat pada tabel 3.30

Tabel 3.30 Operasi pada *class* AdminModel

Nama	Keterangan
Kategori	
Get_Category() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data kategori.
get_sub_category	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data subkategori.
sql_tambah_kategori() : void	Prosedur yang dipanggil ketika proses tambah data kategori, melakukan perintah simpan data pada *database*.
sql_tambah_subkategori() : void	Prosedur yang dipanggil ketika proses tambah data subkategori, melakukan perintah simpan data pada *database*.
proses_edit_kategori() : void	Prosedur yang dipanggil ketika proses ubah data kategori, melakukan perintah update data pada *database*.
proses_edit_subkategori() : void	Prosedur yang dipanggil ketika proses ubah data subkategori, melakukan perintah update data pada *database*.

Delete_Kategori() : void	Prosedur yang dipanggil ketika hapus data, menghapus data kategori pada *database*.
Buku	
Get_Buku() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data produk.
sql_tambah_buku() : void	Prosedur yang dipanggil ketika proses tambah data produk, melakukan perintah simpan data pada *database*.
proses_edit_buku() : void	Prosedur yang dipanggil ketika proses ubah data produk, melakukan perintah update data pada *database*.
Delete_Buku() : void	Prosedur yang dipanggil ketika hapus data, menghapus data produk pada *database*.
Laporan dan Pemesanan	
Get_pemesanan() : void	Prosedur mengambil data pesanan dari database dengan memanggil class pemesananModel dan digunakan untuk menampilkan data pesanan yang akan dibuatkan laporannya.
User	
Check_user	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data *user*.
sql_tambah_user() : void	Prosedur yang dipanggil ketika proses tambah data *user*, melakukan perintah simpan data pada *database*.
proses_edit_User() : void	Prosedur yang dipanggil ketika proses ubah data *user*, melakukan perintah update data pada *database*.
Delete_User() : void	Prosedur yang dipanggil ketika hapus data, menghapus data *user* pada *database*.
Penerbit	
Get_Penerbit() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data penerbit.
sql_tambah_penerbit() : void	Prosedur yang dipanggil ketika proses tambah data penerbit, melakukan perintah simpan data pada *database*.
proses_edit_penerbit() : void	Prosedur yang dipanggil ketika proses ubah data penerbit, melakukan perintah update data pada *database*.
Delete_Penerbit() : void	Prosedur yang dipanggil ketika hapus data, menghapus data penerbit pada *database*.
Pemesanan	
get_pemesanan	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data pemesanan.
get_detail_pemesanan	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data detail pemesanan.
proses_edit_pemesanan	Prosedur yang dipanggil ketika proses ubah data pemesanan, melakukan perintah update data pada *database*.

4. *Class* KatalogModel

   Class AdminModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah semua data yang ada pada aplikasi yang dibangun. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

c. Deskripsi Atribut

Class AdminModel tidak memiliki atribut.

d. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class AdminModel dapat dilihat pada tabel 3.31

Tabel 3.31 Operasi pada *class* KatalogModel

Nama	Keterangan
Kategori	
Get_Category() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data kategori.
get_sub_category() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data subkategori.
Buku	
Get_Book() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data buku.
Laporan dan Pemesanan	
Get_pemesanan() : void	Prosedur mengambil data pesanan dari database dengan memanggil class pemesananModel dan digunakan untuk menampilkan data pesanan yang akan dibuatkan laporannya.
Pemesanan	
get_detail_pemesanan() : void	Prosedur mengambil data dari *database*, digunakan untuk menampilkan data detail pemesanan.
get_pemesanan_by_id() : void	Prosedur mengambil data pemesanan berdasarkan id_pemesanan.
proses_edit_pemesanan() : void	Prosedur yang dipanggil ketika proses ubah data pemesanan, melakukan perintah update data pada *database*.
insert_pesan() : void	Prosedur menyiman data pemesanan ke *database*.
insert_retur() : void	Prosedur yang digunakan untuk menyimpan data retur kedalam database.
insert_konfirmasi() : void	Prosedur yang digunakan untuk menyimpan konfirmasi pembayaran ke dalam database.

5. *Class* KategoriView

Class Kategori merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* KategoriView dapat dilihat pada tabel 3.32

Tabel 3.32 Deskripsi atribut class KategoriView

Nama	Tipe	Panjang	Range
nama	varchar	32	'a'..'Z'

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class KategoriView dapat dilihat pada tabel 3.33

Tabel 3.33 Operasi pada class kategoriView

Nama	Keterangan
Tambah_Kategori() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data pada class AdminControl.
edit_kategori() : void	Prosedur untuk menampilkan *form* ketika memanggil prosedur ubah data kategori, ketika ditampilkan *form* sudah terisi dengan data kategori yang sebelumnya sudah ada pada *database*.
Hapus_kategori() : void	Metode yang berfungsi untuk memanggil metode Hapus_Kategori() pada class AdminController

6. *Class* Buku*View*

Class Buku merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* BukuView dapat dilihat pada tabel 3.34

Tabel 3.34 Deskripsi atribut class BukuView

Nama	Tipe	Panjang	Range
isbn	varchar	17	'0'..'9'
quantity	Int	4	'0'..'9'
judul	varchar	128	'a'..'Z', '0'..'9'
gambar	Varchar	255	'a'..'Z', '0'..'9'
Id_penerbit	Int	3	'0'..'9'
id_kategori	int	3	'0'..'9'
id_subkategori	int	3	'0'..'9'
harga	int	11	'0'..'9'
diskon	float	3,2	'0'..'9'
berat	decimal	5.2	'0'..'9'
tgl_masuk	datetime	0000-00-00 00:00:00	
Tgl_ubah	datetime	0000-00-00 00:00:00	
Pengarang	varchar	32	'a'..'Z'
halaman	int	4	'0'..'9'

ukuran	varchar	10	'a'..'Z', '0'..'9'
deskripsi	text		'a'..'Z', '0'..'9'
dihapus	varchar	1	'Y', 'N'

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class Bukuview dapat dilihat pada tabel 3.35

Tabel 3.35 Operasi pada class Buku*View*

Nama	Keterangan
Tambah_Buku() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data pada class BukuControl.
Edit_Buku() : void	Prosedur untuk menampilkan form ketika memanggil prosedur ubah data, ketika ditampilkan form sudah terisi dengan data produk yang sebelumnya sudah ada pada *database*.
proses_pengadaan() : void	Metode yang berfungsi untuk memanggil metode pengadaanstok() pada class AdminController
proses_pengadaan() : void	Metode yang berfungsi untuk memanggil metode pengadaanstok() pada class AdminController
surat_pemesanan_pdf() : void	Metode yang berfungsi untuk memanggil metode surat_pemesanan_pdf() pada class AdminController.
Hapus () : void	Metode yang berfungsi untuk memanggil metode Hapus_Buku() pada class AdminController

7. *Class* LaporanView

Class Laporan merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

a. Deskripsi Atribut

Class LaporanView tidak memiliki atribut.

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class laporan*View* dapat dilihat pada tabel 3.36

Tabel 3.36 Operasi pada class laporan*View*

Nama	Keterangan
Tombol Cetak_laporan()	Metode yang berfungsi untuk mencetak laporan yang tampil

Tombol Cari_buku_terjual ()	Metode yang berfungsi untuk memanggil metode cari_buku_terjual() pada class AdminController

8. *Class* PemesananView

Class pemesanan merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan konsumen.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* PemesananView dapat dilihat pada tabel 3.37

Tabel 3.37 Deskripsi atribut class PemesananView

Nama Field	Tipe	Panjang	Range
status_pengiriman	varchar	20	'a'..'Z'
no resi	varchar	13	

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada *class* PemesananView dapat dilihat pada tabel 3.38

Tabel 3.38 Operasi pada *class* PemesananView

Nama	Keterangan
Tombol Ubah() : void	Metode yang berfungsi untuk memanggil metode UbahDataPemesanan() pada class AdminController.
Tombol Cetak_label() : void	Metode yang berfungsi untuk memanggil metode cetak_label() pada class AdminController

9. *Class* UserView

Class User merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* UserView dapat dilihat pada tabel 3.39

Tabel 3.39 Deskripsi atribut *class* UserView

Nama Field	Tipe	Panjang	Range
id_user	int	2	’0’..’9’
id_grup	int	2	’a’..’Z’
username	varchar	20	’a’..’Z’, ’0’..’9’
password	varchar	32	’a’..’Z’, ’0’..’9’

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada *class* UserView dapat dilihat pada tabel 3.40

Tabel 3.40 Operasi pada *class* UserView

Nama	Keterangan
Edit_User() : void	Prosedur untuk menampilkan *form* ketika memanggil prosedur ubah data, ketika ditampilkan form sudah terisi dengan data *user* yang sebelumnya sudah ada pada database.
Tambah_User() : void	Prosedur untuk menampilkan form ketika memanggil prosedur tambah data pada *class* AdminController.
Hapus_User () : void	Metode yang berfungsi untuk memanggil metode Hapus_User() pada class AdminController

10. *Class* LoginView

Class Login merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* LoginView dapat dilihat pada tabel 3.41

Tabel 3.41 Deskripsi atribut *class* LoginView

Nama Field	Tipe	Panjang	Range
username	varchar	20	’a’..’Z’, ’0’..’9’
password	varchar	32	’a’..’Z’, ’0’..’9’

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada *class* LoginView dapat dilihat pada tabel 3.42

Tabel 3.42 Operasi pada *class* LoginView

Nama	Keterangan
proses_login () : void	Metode yang digunakan untuk memanggil CekLogin() pada LoginController

11. Class PenerbitView

Class Penerbit merupakan class yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* PenerbitView dapat dilihat pada tabel 3.43

Tabel 3.43 Deskripsi atribut *class* PenerbitView

Nama Field	Tipe	Panjang	Range
nama	varchar	64	’a’..’Z’, ’0’..’9’
alamat_penerbit	Varchar	128	’a’..’Z’, ’0’..’9’

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class penerbitView dapat dilihat pada tabel 3.44

Tabel 3.44 Operasi pada *class* PenerbitView

Nama	Keterangan
Tambah_Penerbit() : void	Metode yang berfungsi untuk memanggil metode Tambah_Penerbit() pada class AdminController
Edit_Penerbit() : void	Metode yang berfungsi untuk memanggil metode edit_Penerbit() pada class AdminController.
Tombol Hapus_penerbit() : void	Metode yang berfungsi untuk memanggil metode Hapus_Penerbit() pada class AdminController

12. *Class* ReturView

Class returview merupakan class yang berisi prosedur-prosedur untuk mengatur tampilan retur yang berinteraksi langsung dengan user.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* ReturView dapat dilihat pada tabel 3.45

Tabel 3.45 Deskripsi atribut *class* ReturView

Nama Field	Tipe	Panjang	Range
no_resi	varchar	13	’0’..’9’
alasan	Varchar	text	’a’..’Z’, ’0’..’9’

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class penerbitView dapat dilihat pada tabel 3.46

Tabel 3.46 Operasi pada *class* ReturView

Nama	Keterangan
proses_retur() : void	Metode yang berfungsi untuk memanggil metode proses_retur() pada class KatalogController
ubah_status_retur () : void	Metode yang berfungsi untuk memanggil metode ubah_status_retur() pada class KatalogController.

13. SubkategoriView

Class subkategoriView merupakan class yang berisi prosedur-prosedur untuk mengatur tampilan subkategori yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* SubkategoriView dapat dilihat pada tabel 3.47

Tabel 3.47 Deskripsi atribut *class* SubkategoriView

Nama Field	Tipe	Panjang	Range
Nama Subkategori	Varchar	text	’a’..’Z’, ’0’..’9’

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class penerbitView dapat dilihat pada tabel 3.48

Tabel 3.48 Operasi pada *class* SubkategoriView

Nama	Keterangan
tambah_subkategori() : void	Prosedur yang digunakan untuk memanggil proses_tambah_subkategori pada adminController.
edit_ subkategori () : void	Prosedur yang digunakan untuk memanggil proses_edit_subkategori pada adminController.

14. *Class* KontakView

Class kontakview merupakan class yang berisi prosedur-prosedur untuk mengatur tampilan kontak yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* KontakView dapat dilihat pada tabel 3.49

Tabel 3.49 Deskripsi atribut *class* KontakView

Nama Field	Tipe	Panjang	Range
Nama Subkategori	Varchar	text	’a’..’Z’, ’0’..’9’
alamat	varchar	128	’a’..’Z’, ’0’..’9’
email	varchar	32	’a’..’Z’, ’0’..’9’
pesan	text		’a’..’Z’, ’0’..’9’

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class kontakView dapat dilihat pada tabel 3.50

Tabel 3.50 Operasi pada *class* KontakView

Nama	Keterangan
proses_kontak() : void	Prosedur yang digunakan untuk memanggil proses_kontak pada KatalogController.
balas_kontak () : void	Prosedur yang digunakan untuk memanggil balas_kontak pada AdminController.

15. *Class* PromosiView

Class promosiview merupakan class yang berisi prosedur-prosedur untuk mengatur tampilan promosi yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* promosiview dapat dilihat pada tabel 3.51

Tabel 3.51 Deskripsi atribut *class* PromosiView

Nama Field	Tipe	Panjang	Range
gambar	varchar	255	'a'..'Z', '0'..'9'

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class promosiView dapat dilihat pada tabel 3.52

Tabel 3.52 Operasi pada *class* PromosiView

Nama	Keterangan
kirim_promosi() : void	Prosedur yang digunakan untuk memanggil kirim_promosi pada AdminController.

16. *Class* KonfirmasiView

Class konfirmasiview merupakan class yang berisi prosedur-prosedur untuk mengatur tampilan konfirmasi yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* konfirmasiview dapat dilihat pada tabel 3.53

Tabel 3.53 Deskripsi atribut *class* KonfirmasiView

Nama Field	Tipe	Panjang	Range
id_pemesanan	int	5	'a'..'Z', '0'..'9'
nama_pengirim	varchar	32	'a'..'Z'
tanggal_transfer	date		'0'..'9'

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class KonfirmasiView dapat dilihat pada tabel 3.54

Tabel 3.54 Operasi pada *class* KonfirmasiView

Nama	Keterangan
Confirm () : void	Prosedur yang digunakan untuk memanggil confirm pada AdminController.
proses_confirm() : void	Prosedur yang digunakan untuk memanggil metode proses_confirm pada AdminController.

17. *Class* KeranjangView

Class keranjangview merupakan class yang berisi prosedur-prosedur untuk mengatur tampilan keranjang yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Deskripsi dari atribut yang terdapat pada *class* keranjang dapat dilihat pada tabel 3.55

Tabel 3.55 Deskripsi atribut *class* keranjangview

Nama Field	Tipe	Panjang	Range
nama	varchar	32	’a’..’Z’, ’0’..’9’
telepon	varchar	13	’a’..’Z’
email	varchar	32	’0’..’9’
provinsi	varchar	100	’a’..’Z’, ’0’..’9’
alamat	varchar	128	’a’..’Z’, ’0’..’9’
kota	varchar	50	’a’..’Z’
kodepos	int	5	’0’..’9’

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class KeranjangView dapat dilihat pada tabel 3.56

Tabel 3.56 Operasi pada *class* KeranjangView

Nama	Keterangan
add_to_cart : void	Prosedur yang digunakan untuk memanggil fungsi add_to_cart() pada AdminController.
Update_cart () : void	Prosedur yang digunakan untuk memanggil fungsi update_cart() pada AdminController.
Step1() : void	Prosedur yang digunakan utuk memanggil fungsi step1() pada katalogController yaitu fungsi untuk menampilkan halaman pemesanan langkah pertama.
Step2() : void	Prosedur yang digunakan utuk memanggil fungsi step2() pada katalogController yaitu fungsi untuk menampilkan halaman pemesanan langkah kedua.
Step3() : void	Prosedur yang digunakan utuk memanggil fungsi step3()pada

	katalogController yaitu fungsi untuk menampilkan halaman pemesanan langkah ketiga.
Step4() : void	Prosedur yang digunakan utuk memanggil fungsi step4()pada katalogController yaitu fungsi untuk menampilkan halaman pemesanan langkah keempat.

18. *Class* BerandaView

Class berandaview merupakan class yang berisi prosedur-prosedur untuk mengatur tampilan beranda yang berinteraksi langsung dengan *user*.

a. Deskripsi Atribut

Class berandaView tidak meiliki attribut.

b. Deskripsi Operasi

Penjelasan dari operasi yang ada pada class KeranjangView dapat dilihat pada tabel 3.57

Tabel 3.57 Operasi pada *class* KeranjangView

Nama	Keterangan
cari_buku : void	Prosedur yang digunakan untuk memanggil fungsi cari_buku() pada AdminController.
tabel_buku () : void	Prosedur yang digunakan untuk menampilkan halaman buku.

19. *Class* Penerbit

Class penerbit merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel penerbit.

a. Deskripsi Atribut

Deskripsi atribut dari *class* penerbit dapat dilihat pada tabel 3.58

Tabel 3.58 Deskripsi atribut *class* penerbit

Nama Field	Tipe	Panjang	Range
Id_penerbit	Int	11	’0’..’9’
nama_penerbit	Varchar	32	’a’..’Z’
alamat_penerbit	Varchar	128	’a’..’Z’, ’0’..’9’
Dihapus	Varchar	1	‘Y’, ‘N’

b. Deskripsi Operasi

Class Penerbit tidak memiliki Operasi

20. *Class* Kategori

Class Kategori merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel kategori.

a. Deskripsi Atribut

Deskripsi atribut dari *class* kategori dapat dilihat pada tabel 3.59

Tabel 3.59 Deskripsi atribut *class* kategori

Nama Field	Tipe	Panjang	Range
Id_kategori	Int	3	'0'..'9'
parent	varchar	20	'a'..'Z',
Nama	varchar	32	'a'..'Z',
Dihapus	varchar	1	'Y','N'

b. Deskripsi Operasi

Class kategori tidak memiliki Operasi

21. *Class* Buku

Class buku merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel buku.

a. Deskripsi Atribut

Deskripsi atribut dari *class* buku dapat dilihat pada tabel 3.60

Tabel 3..60 Deskripsi atribut *class* buku

Nama Field	Tipe	Panjang	Range
isbn	varchar	17	'0'..'9'
quantity	Int	4	'0'..'9'
judul	varchar	128	'a'..'Z', '0'..'9'
gambar	Varchar	255	'a'..'Z', '0'..'9'
Id_penerbit	Int	3	'0'..'9'
id_kategori	int	3	'0'..'9'
id_subkategori	int	3	'0'..'9'
harga	int	11	'0'..'9'
diskon	float	3,2	'0'..'9'

berat	decimal	5.2	’0’..’9’
tgl_masuk	datetime	0000-00-00 00:00:00	
Tgl_ubah	datetime	0000-00-00 00:00:00	
Pengarang	varchar	32	’a’..’Z’
halaman	int	4	’0’..’9’
ukuran	varchar	10	’0’..’9’
deskripsi	text		’a’..’Z’, ’0’..’9’
dihapus	varchar	1	‘Y', ‘N’

b. Deskripsi Operasi

Class buku tidak memiliki Operasi

22. *Class* Pemesanan

Class pemesanan merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel pemesanan.

a. Deskripsi Atribut

Deskripsi atribut dari *class* pemesanan dapat dilihat pada tabel 3.61

Tabel 3.61 Deskripsi atribut *class* pemesanan

Nama Field	Tipe	Panjang	Range
nama	varchar	20	’a’..’Z’
id_pemesanan	int	5	’0’..’9’
id_user	int	2	’0’..’9’
telepon	varchar	13	’0’..’9’
email	varchar	32	’a’..’Z’, ’0’..’9’
alamat	varchar	32	’a’..’Z’, ’0’..’9’
kodepos	varchar	5	’0’..’9’
id_kabkota	int	4	’0’..’9’
id_provinsi	int	3	’0’..’9’
status_pembayaran	varchar	20	’a’..’Z’
status_pengiriman	varchar	20	’a’..’Z’
nama_pengiriman	varchar	32	’a’..’Z’
telepon_pengiriman	varchar	13	’0’..’9’
total	decimal	15,2	’0’..’9’
email_pengiriman	varchar	32	’a’..’Z’, ’0’..’9’
alamat pengiriman	varchar	128	’a’..’Z’, ’0’..’9’
kodepos_pengiriman	int	5	’0’..’9’
id_kota_pengiriman	int	4	’0’..’9’
id_provinsi_pengiriman	int	3	’0’..’9’
total	int	11	’0’..’9’
berat	int	11	’0’..’9’
no resi	varchar	13	’0’..’9’

b. Deskripsi Operasi

Class pemesanan tidak memiliki Operasi

23. *Class User*

Class user merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel *user*.

a. Deskripsi Atribut

Deskripsi atribut dari *class user* dapat dilihat pada tabel 3.62

Tabel 3.62 Deskripsi atribut *class user*

Nama Field	**Tipe**	**Panjang**	**Range**
id_user	int	2	’0’..’9’
id_grup	int	2	’0’..’9’
aktif	varchar	1	‘Y’, ‘N’
username	varchar	20	’a’..’Z’, ’0’..’9’
password	varchar	32	’a’..’Z’, ’0’..’9’

b. Deskripsi Operasi

Class user tidak memiliki Operasi

24. *Class* Pemesanan detail

Class pemesanan detail merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel pemesanan detail.

a. Deskripsi Atribut

Deskripsi atribut dari *class* pemesanan detail dapat dilihat pada tabel 3.63

Tabel 3.63 Deskripsi atribut *class* pemesanan detail

Nama Field	Tipe	Panjang	Keterangan
id_pemesanan_detail	int	5	’0’..’9’
id_pemesanan	int	5	’0’..’9’
isbn	varchar	17	’0’..’9’
judul	varchar	128	’a’..’Z’, ’0’..’9’
isbn	varchar	24	’0’..’9’
harga	int	11	’0’..’9’
quantity	int	4	’0’..’9’

b. Deskripsi Operasi

Class pemesanan detail tidak memiliki Operasi

25. *Class* kontak

Class kontak merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel kontak.

a. Deskripsi Atribut

Deskripsi atribut dari *class* kontak dapat dilihat pada tabel 3.64

Tabel 3.64 Deskripsi atribut *class* kontak

Nama Field	Tipe	Panjang	Keterangan
Id_kontak	Int	5	'0'..'9'
Nama	Varchar	32	'a'..'Z'
alamat	varchar	128	'a'..'Z', '0'..'9'
email	varchar	32	'a'..'Z', '0'..'9'
telepon	varchar	13	'0'..'9'
pesan	text		'a'..'Z', '0'..'9'
tanggal	datetime		

b. Deskripsi Operasi

Class kontak tidak memiliki Operasi

26. *Class* Retur

Class Retur merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel retur.

a. Deskripsi Atribut

Deskripsi atribut dari *class* retur dapat dilihat pada tabel 3.65

Tabel 3.65 Deskripsi atribut *class* retur

Nama Field	Tipe	Panjang	Keterangan
id_retur	Int	3	'0'..'9'
id_pemesanan	int	5	'0'..'9'
no_resi	varchar	13	'0'..'9'
alasan	text		'a'..'Z', '0'..'9'
tanggal_retur	date		
status	varchar	20	'Y', 'N'
no_resi_retur	varchar	13	'0'..'9'

tgl_penerimaan	Date		
nama_penerima	Varchar	20	’a’..’Z’
waktu_diterima	Varchar	6	’0’..’9’

b. Deskripsi Operasi

Class retur tidak memiliki Operasi

27. *Class* Konfirmasi

Class konfirmasi merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel konfirmasi.

a. Deskripsi Atribut

Deskripsi atribut dari *class* konfirmasi dapat dilihat pada tabel 3.66

Tabel 3.66 Deskripsi atribut *class* konfirmasi

Nama Field	Tipe	Panjang	Keterangan
id_konfirmasi	Int	5	’0’..’9’
kode_faktur	int	5	’0’..’9’
nama	Varchar	32	’a’..’Z’
email	varchar	32	’a’..’Z’, ’0’..’9’
jumlah	int	11	’0’..’9’
bank	Varchar	32	’a’..’Z’
bank_konsumen	varchar	32	’a’..’Z’
nama_pengirim	int	32	’a’..’Z’
dicek	varchar1	1	‘Y’, ‘T’

b. Deskripsi Operasi

Class konfirmasi tidak memiliki Operasi

28. *Class* Subkategori

Class subkategori merupakan kelas entitas yang berfungsi untuk melakukan *mapping* ke tabel subkategori.

a. Deskripsi Atribut

Deskripsi atribut dari *class* subkategori dapat dilihat pada tabel 3.67

Tabel 3.67 Deskripsi atribut *class* subkategori

Nama Field	Tipe	Panjang	Keterangan
Id_sub_kategori	Int	3	’0’..’9’

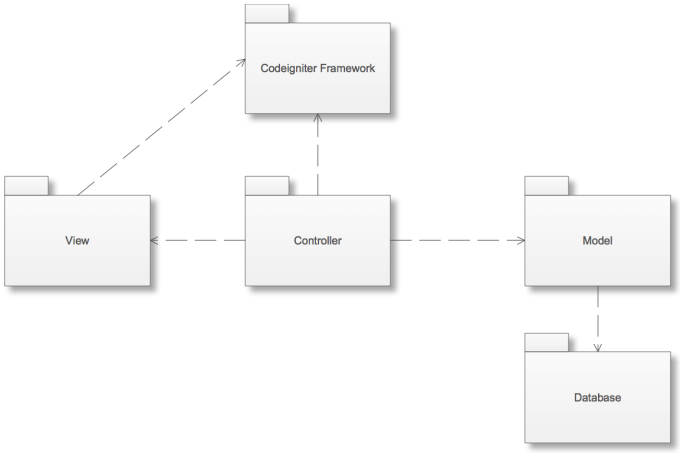
nama_sub_kategori	Varchar	32	'a'..'Z'
id_kategori	Int	3	'0'..'9'
dihapus	varchar	1	'Y', 'N'

b. Deskripsi Operasi

Class subkategori tidak memiliki Operasi

3.1.8.7 *Package* Diagram

Package diagram adalah sebuah bentuk pengelompokkan yang memungkinkan untuk mengambil setiap bentuk di UML dan mengelompokkan elemen-elemennya dalam tingkatan unit yang lebih tinggi. *Package* diagram untuk pengelompokkan *class* aplikasi *e-commerce* pada toko Rumah Buku Bandung dapat dilihat pada gambar 3.28

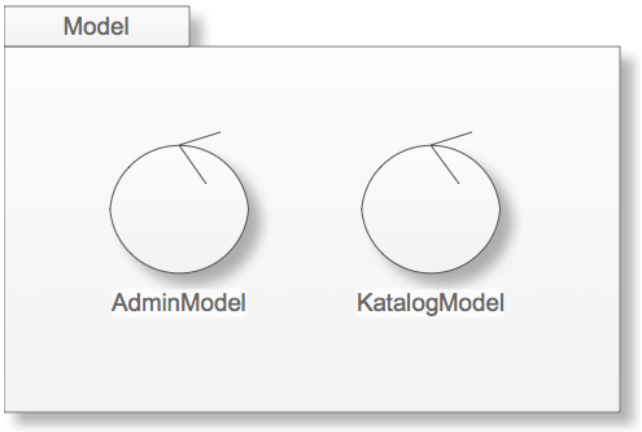


Gambar 3.28 *Package diagram e-commerce* toko Rumah Buku Bandung

Deskripsi untuk *package diagram* pada gambar 3.28 adalah sebagai berikut :

1. *Package Model*

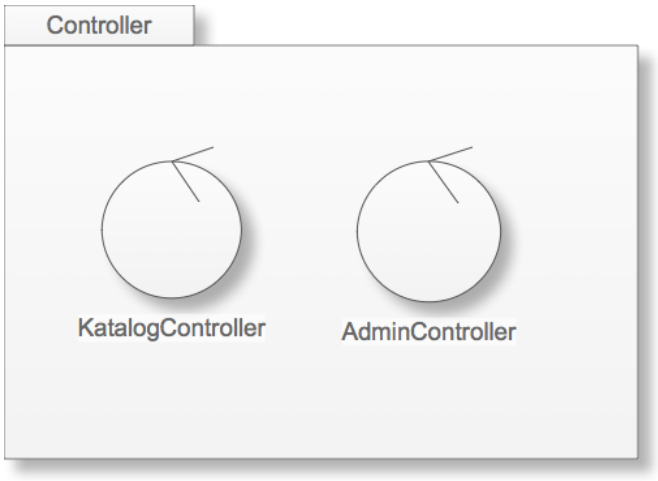
   Package Model berisi *class-class* model yang mengakses langsung pada *database* untuk dapat mengolah data. Isi dari *package* Model dapat dilihat pada gambar 3.29



Gambar 3.29 Komponen *package* Model

2. *Package Controllers*

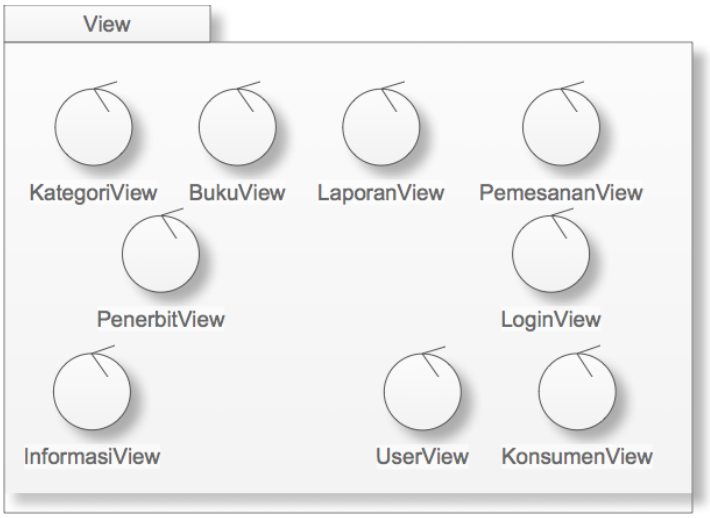
   Package Controllers berisi *class-class control* yang merupakan pengatur dalam setiap proses yang ada. Isi dari *package Controllers* dapat dilihat pada gambar 3.30



Gambar 3.30 Komponen *package controller*

3. *Package View*

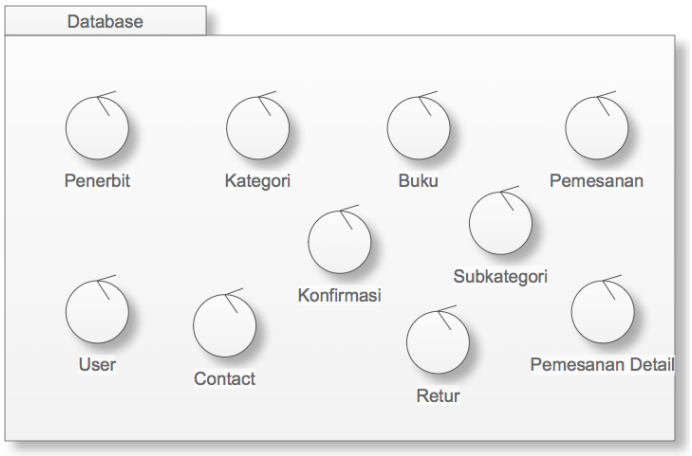
   Package View berisi *class-class view* yang digunakan untuk menampilkan interface, dapat berupa tabel atau *form*. Isi dari *Package View* dapat dilihat pada gambar 3.31



Gambar 3.31 Komponen *package view*

4. *Package Database*

   Package *database* berisi *class-class view* yang digunakan untuk menyimpan data. Isi dari *package database* dapat dilihat pada gambar 3.32

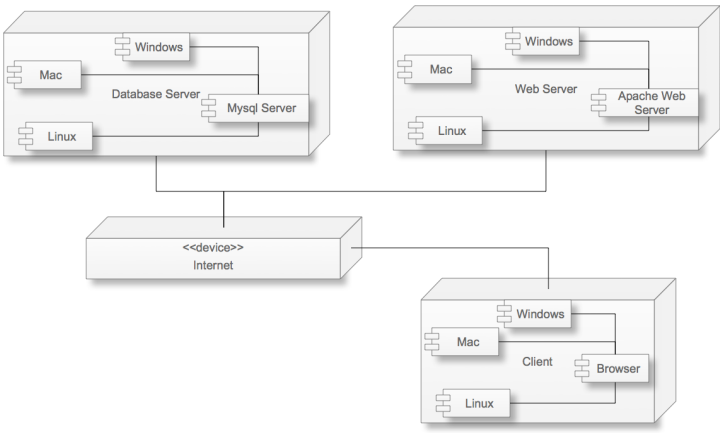


Gambar 3.32 Komponen *package database*

3.1.8.8 *Deployment Diagram*

Menggambarkan sumber fisik dalam sistem, termasuk *node*, komponen dan koneksi (model implementasi sistem yang statistik). Dalam hal ini meliputi

topologi *hardware* yang dipakai sistem. Deployment dari aplikasi *e-commerce* pada toko rumah buku dapat dilihat pada gambar 3.33



Gambar 3.33 *Deployment Diagram*

3.2 Perancangan Sistem

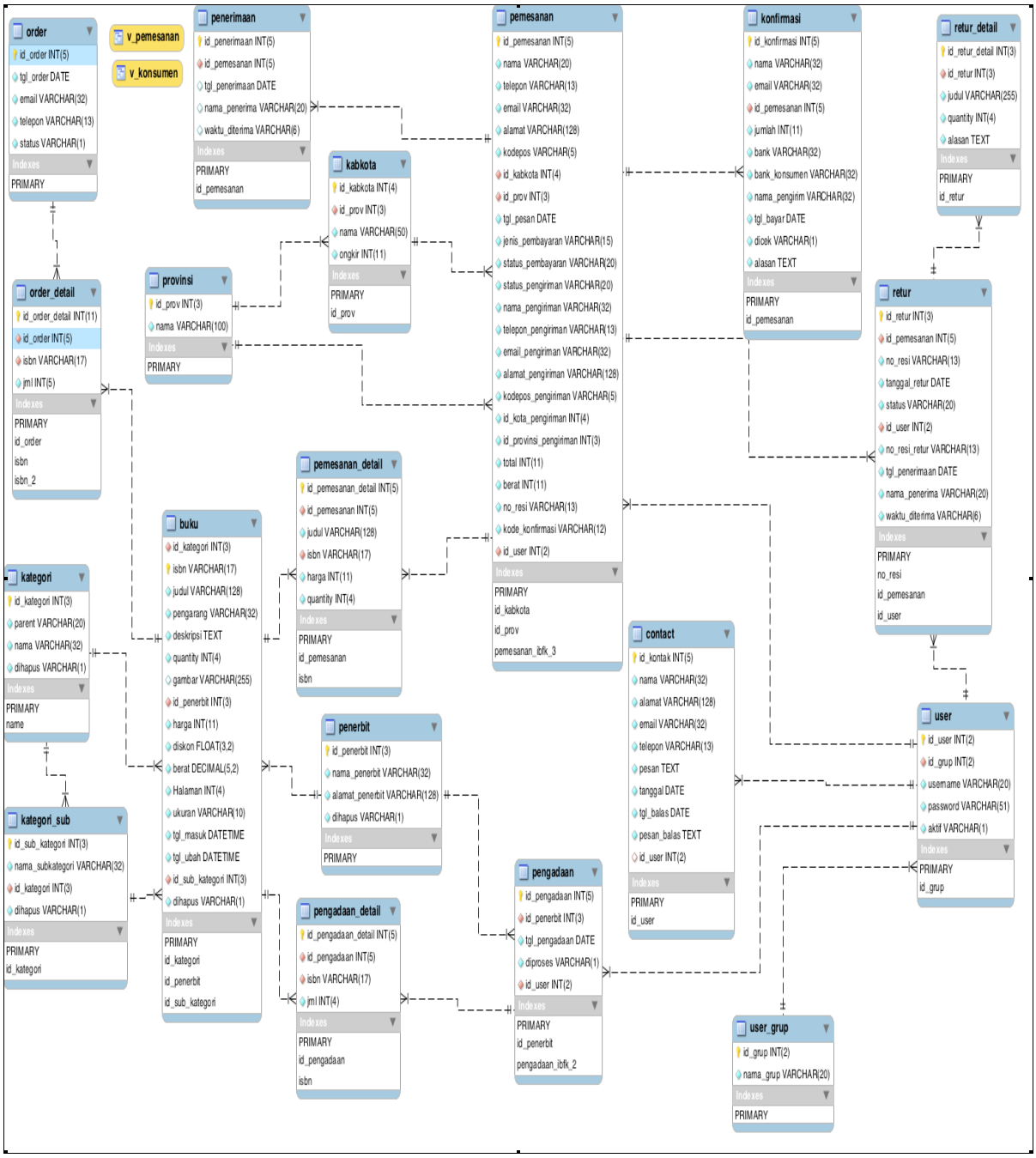
Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi. Tahapan ini meliputi mengkonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi sistem akan benar-benar memuaskan berdasarkan rancang bangun yang telah ditetapkan pada tahap akhir analisis sistem.

3.2.1 Perancangan Data

Perancangan data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perancangan data terbagi menjadi dua yaitu diagram relasi dan perancangan struktur tabel.

3.2.1.1 Diagram Relasi

Diagram relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Penjelasan rangkaian basis data aplikasi *e-commerce* pada toko rumah buku dapat dilihat pada gambar 3.34



Gambar 3.34 Diagram relasi aplikasi *e-commerce* Rumah Buku Bandung

3.2.1.2 Struktur Tabel

Tabel-tabel yang digunakan dalam *database* aplikasi *e-commerce* pada toko Rumah Buku adalah :

1. Tabel *User*

   Tabel *user* digunakan untuk menyimpan data *admin* dan data petugas seperti *username* dan *password*. Struktur tabel *user* dapat dilihat pada tabel 3.68

Tabel 3.68 Struktur tabel *user*

Nama Field	Tipe	Panjang	Keterangan
id_user	int	2	*Primary Key*
id_grup	int	2	FK1 *reference* user_grup (id_grup)
username	varchar	20	
password	varchar	32	
aktif	varchar	1	

2. Tabel buku

   Tabel buku digunakan untuk menyimpan data buku dan jumlah ketersediaan buku. Struktur tabel admin_grup dapat dilihat pada tabel 3.69

Tabel 3.69 Struktur tabel buku

Nama Field	Tipe	Panjang	Keterangan
isbn	varchar	17	*Primary Key*
quantity	Int	4	
judul	varchar	128	
gambar	Varchar	255	
Id_penerbit	Int	3	FK2 *reference* penerbit (id_penerbit)
id_kategori	int	3	FK1 *reference* kategori (id_kategori)
id_subkategori	int	3	FK1 *reference* kategori (id_subkategori)
harga	int	11	
diskon	float	3,2	
berat	decimal	5.2	
tgl_masuk	datetime	0000-00-00 00:00:00	
Tgl_ubah	datetime	0000-00-00 00:00:00	
Pengarang	varchar	32	
halaman	int	4	
ukuran	varchar	10	
deskripsi	text		
dihapus	varchar	1	

3. Tabel kabkota

Tabel kabkota digunakan untuk menyimpan data kabupaten dan kota. Struktur tabel kabkota dapat dilihat pada tabel 3.70

Tabel 3.70 Struktur tabel kabkota

Nama Field	Tipe	Panjang	Keterangan
Id_kabkota	Int	4	*Primary Key*
id_prov	int	3	FK1 *reference* provinsi (id_prov)
nama	varchar	50	
kabkota	varchar	20	
ongkir	Int	11	

4. Tabel kategori

Tabel kategori digunakan untuk menyimpan semua data kategori buku. Struktur tabel kategori dapat dilihat pada tabel 3.71

Tabel 3.71 Struktur tabel kategori

Nama Field	Tipe	Panjang	Keterangan
Id_kategori	Int	3	*Primary Key*
parent	varchar	20	
Nama	varchar	32	
Dihapus	varchar	1	

5. Tabel pemesanan

Tabel pemesanan digunakan untuk menyimpan data pemesanan. Struktur tabel pemesanan dapat dilihat pada tabel 3.72

Tabel 3.72 Struktur tabel pemesanan

Nama Field	Tipe	Panjang	Keterangan
nama	varchar	20	Primary Key
id_pemesanan	int	5	
id_user	int	2	FK1 *reference* user (id_user)
telepon	varchar	13	
email	varchar	32	
alamat	varchar	32	
kodepos	varchar	5	
id_kabkota	int	4	FK2 *reference* kabkota (id_kabkota
id_provinsi	int	3	FK3 *reference* provinsi (id_prov)
status_pembayaran	varchar	20	
status_pengiriman	varchar	20	
nama_pengiriman	varchar	32	

telepon_pengiriman	varchar	13	
total	decimal	15,2	
email_pengiriman	varchar	32	
alamat pengiriman	varchar	128	
kodepos_pengiriman	int	5	
id_kota_pengiriman	int	4	
id_provinsi_pengiriman	int	3	
total	int	11	
berat	int	11	
no resi	varchar	13	

6. Tabel pemesanan_detail

   Tabel pemesanan_detail digunakan untuk menyimpan data pemesanan detail. Struktur tabel pemesanan_detail dapat dilihat pada tabel 3.73

Tabel 3.73 Struktur tabel pemesanan_detail

Nama Field	Tipe	Panjang	Keterangan
id_pemesanan_detail	int	5	*Primary Key*
id_pemesanan	int	5	FK2 *reference* pemesanan (id_pemesanan)
isbn	varchar	17	FK1 *reference* buku (isbn)
judul	varchar	128	
isbn	varchar	24	
harga	int	11	
quantity	int	4	

7. Tabel penerbit

   Tabel penerbit digunakan untuk menyimpan data penerbit. Struktur tabel penerbit dapat dilihat pada tabel 3.74

Tabel 3.74 Struktur tabel penerbit

Nama Field	Tipe	Panjang	Keterangan
Id_penerbit	Int	11	Primary Key
nama_penerbit	Varchar	32	
alamat_penerbit	Varchar	128	
Dihapus	Varchar	1	

8. Tabel provinsi

   Tabel provinsi digunakan untuk menyimpan data provinsi. Struktur tabel provinsi dapat dilihat pada tabel 3.75

Tabel 3.75 Struktur tabel provinsi

Nama Field	Tipe	Panjang	Keterangan
Id_prov	Int	3	Primary Key
Nama	Varchar	100	

9. Tabel *contact*

Tabel *contact* digunakan untuk menyimpan data pertanyaan dari halaman kontak kami. Struktur tabel *contact* dapat dilihat pada tabel 3.76

Tabel 3.76 Struktur tabel *contact*

Nama Field	Tipe	Panjang	Keterangan
Id_kontak	Int	5	Primary Key
Nama	Varchar	32	
alamat	varchar	128	
email	varchar	32	
telepon	varchar	13	
pesan	text		
tanggal	datetime		

10. Tabel kategori_sub

Tabel kategori_sub digunakan untuk menyimpan data sub kategori. Struktur tabel kategori_sub dapat dilihat pada tabel 3.77

Tabel 3.77 Struktur tabel kategori_sub

Nama Field	Tipe	Panjang	Keterangan
Id_sub_kategori	Int	3	Primary Key
nama_sub_kategori	Varchar	32	
id_kategori	Int	3	FK1 *reference* kategori (id_kategori)
dihapus	varchar	1	

11. Tabel konfirmasi

Tabel konfirmasi digunakan untuk menyimpan data dari hasil konfirmasi pembayaran yang dilakukan oleh. Struktur tabel konfirmasi dapat dilihat pada tabel 3.78

Tabel 3.78 Struktur tabel konfirmasi

Nama Field	Tipe	Panjang	Keterangan
id_konfirmasi	Int	5	Primary Key
id_pemesanan	int	5	FK1 *reference* pemesanan

			(id_pemesanan)
nama	Varchar	32	
email	varchar	32	
jumlah	int	11	
bank	Varchar	32	
bank_konsumen	varchar	32	
nama_pengirim	int	32	
dicek	varchar1	1	

12. Tabel User_Grup

Tabel user_grup digunakan untuk menyimpan data dari user_grup. Struktur tabel konfirmasi dapat dilihat pada tabel 3.79

Tabel 3.79 Struktur tabel user_grup

Nama Field	Tipe	Panjang	Keterangan
id_grup	Int	2	Primary Key
nama_grup	Varchar	20	

13. Tabel Kategori

Tabel kategori digunakan untuk menyimpan data kategori. Tabel kategori dapat dilihat pada tabel 3.80

Tabel 3.80 Struktur tabel kategori

Nama Field	Tipe	Panjang	Keterangan
id_kategori	Int	3	Primary Key
parent	varchar	20	
nama	varchar	32	
dihapus	varchar	1	

14. Tabel *Order*

Tabel *order* digunakan untuk menyimpan data *order* buku dimana konsumen akan memesan lebih dari jumlah stok buku yang ada. Tabel *order* dapat dilihat pada tabel 3.81

Tabel 3.81 Struktur tabel *order*

Nama Field	Tipe	Panjang	Keterangan
id_order	Int	5	Primary Key
tgl_order	date		
email	varchar	32	
telepon	varchar	13	

status	varchar	1	

15. Tabel *Order_detail*

Tabel *order_detail* digunakan untuk menyimpan data order buku dimana konsumen akan memesan lebih dari jumlah stok buku yang ada. Tabel *order* dapat dilihat pada tabel 3.82

Tabel 3.82 Struktur tabel *order_detail*

Nama Field	Tipe	Panjang	Keterangan
id_order_detail	Int	11	Primary Key
id_order	int	5	FK1 *reference* order (id_order)
isbn	varchar	17	FK2 *reference* buku (isbn)
jml	int	5	

16. Tabel pengadaan

Tabel pengadaan digunakan untuk menyimpan data pengadaan buku. Tabel pengadaan dapat dilihat pada tabel 3.83

Tabel 3.83 Struktur tabel pengadaan

Nama Field	Tipe	Panjang	Keterangan
id_pengadaan	Int	5	Primary Key
id_penerbit	int	3	FK1 *reference* penerbit (id_penerbit)
id_user	int	2	FK2 *reference* user (id_user)
tgl_pengadaan	date		
diproses	varchar	1	

17. Tabel pengadaan detail

Tabel pengadaan detail digunakan untuk menyimpan detail dari data pengadaan buku. Tabel pengadaan detail dapat dilihat pada tabel 3.84

Tabel 3.84 Struktur tabel pengadaan detail

Nama Field	Tipe	Panjang	Keterangan
id_pengadaan_detail	Int	5	Primary Key
id_pengadaan	int	5	FK1 *reference* pengadaan (id_pengadaan)
isbn	varchar	17	FK2 *reference* buku (isbn)
jml	int	4	

18. Tabel Retur

Tabel retur digunakan untuk menyimpan data retur buku. Tabel retur dapat dilihat pada tabel 3.85

Tabel 3.85 Struktur tabel retur

Nama Field	Tipe	Panjang	Keterangan
id_retur	Int	3	Primary Key
id_pemesanan	int	5	FK1 *reference* pemesanan (id_pemesanan)
id_user	int	2	FK1 *references* user (id_user)
no_resi	varchar	13	FK2 *reference* pemesanan (no_resi)
alasan	text		
tanggal_retur	date		
status	varchar	20	
no_resi_retur	varchar	13	
tgl_penerimaan	date		
nama_penerima	varchar	20	
waktu_diterima	varchar	6	

19. Tabel retur_detail

Tabel retur digunakan untuk menyimpan detail dari data retur buku. Tabel retur detail dapat dilihat pada tabel 3.86

Tabel 3.86 Struktur tabel retur_detail

Nama Field	Tipe	Panjang	Keterangan
id_retur_detail	Int	3	Primary Key
id_retur	int	3	FK1 *reference* retur (id_retur)
judul	varchar	255	
quantity	int	4	
Alasan	Text		

20. Tabel penerimaan

Tabel retur digunakan untuk menyimpan detail dari data retur buku. Tabel retur detail dapat dilihat pada tabel 3.87

Tabel 3.87 Struktur tabel retur_detail

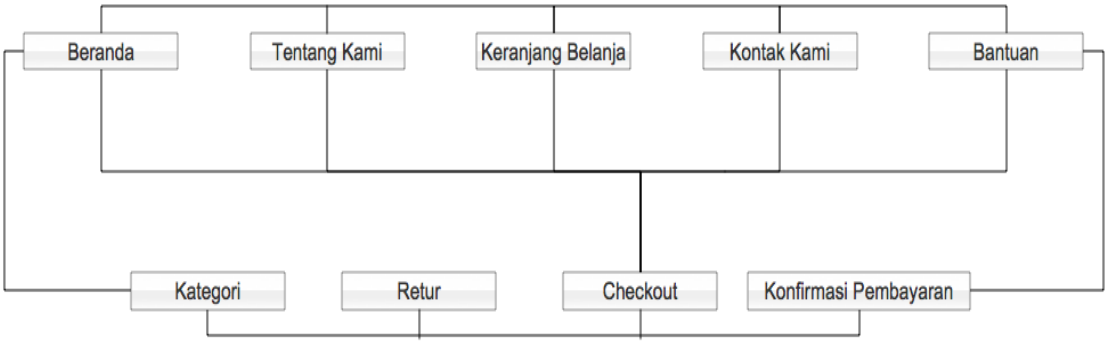
Nama Field	Tipe	Panjang	Keterangan
id_penerimaan	Int	5	Primary Key
id_pemesanan	int	5	FK1 *reference* pemesanan (id_pemesanan)
tgl_penerimaan	date		
nama_penerima	varchar	20	
waktu_diterima	varchar	6	

3.2.2 Arsitektur Menu

Menu dirancang untuk menggambarkan segala macam bentuk kegiatan yang bisa dilakukan dalam program aplikasi, seperti *input* data, cari data dan membuat laporan berdasarkan data *input*. Struktur menu yang dirancang sesuai dengan level pengguna sistem. Terdapat 3 pengguna aplikasi *e-commerce* pada toko Rumah Buku yaitu administrator dan konsumen. Adapun struktur menu dati setiap pengguna adalah sebagai berikut :

a. Perancangan struktur menu konsumen

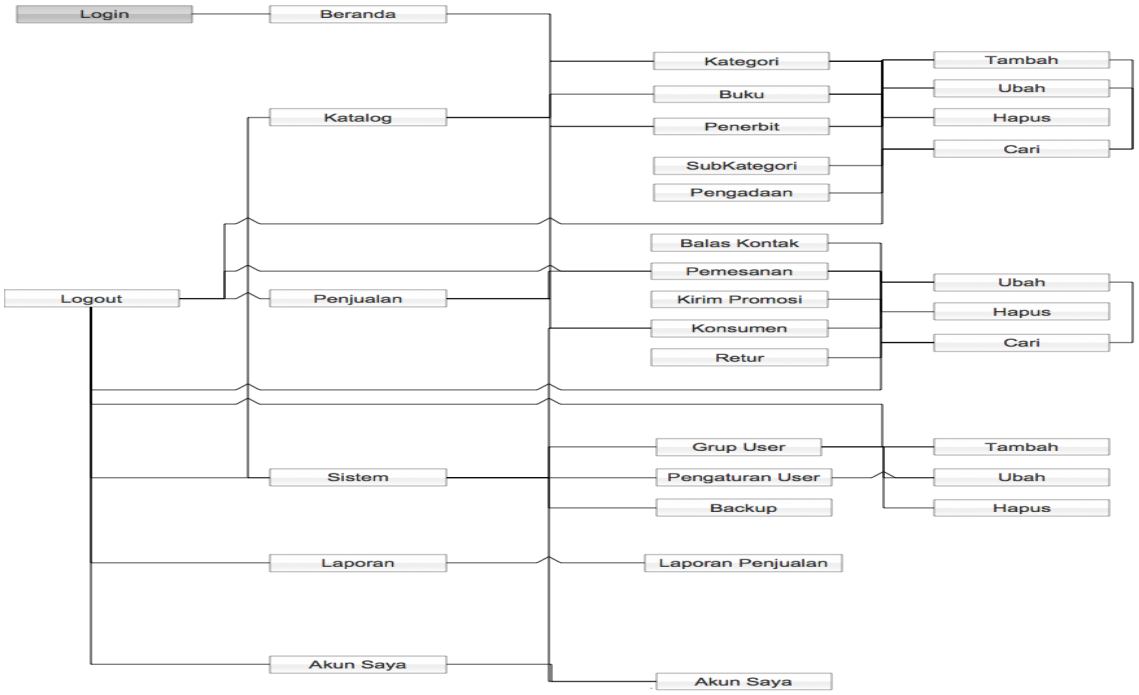
Perancangan struktur menu konsumen dapat dilihat pada gambar 3.35



Gambar 3.35 Struktur menu konsumen

b. Perancangan struktur menu *admin*

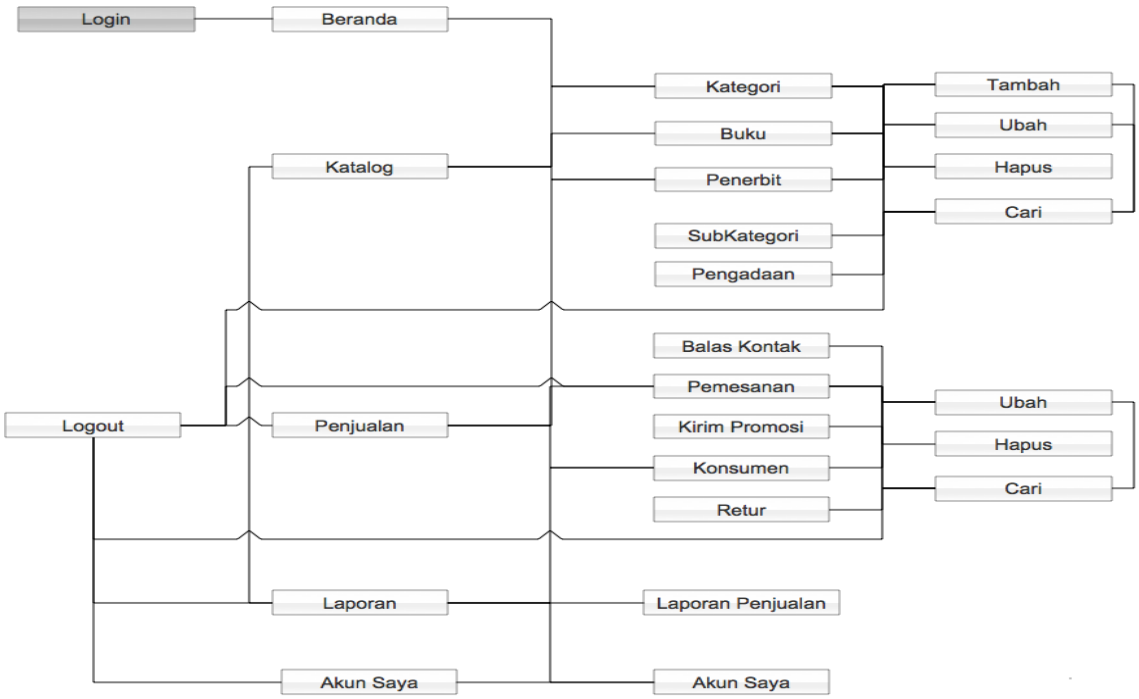
Perancangan struktur menu admin dapat dilihat pada gambar 3.36



Gambar 3.36 Struktur menu *admin*

a. Perancangan struktur menu petugas

Perancangan struktur menu admin dapat dilihat pada gambar 3.37



Gambar 3.37 Struktur menu petugas

3.2.3 Perancangan Antarmuka

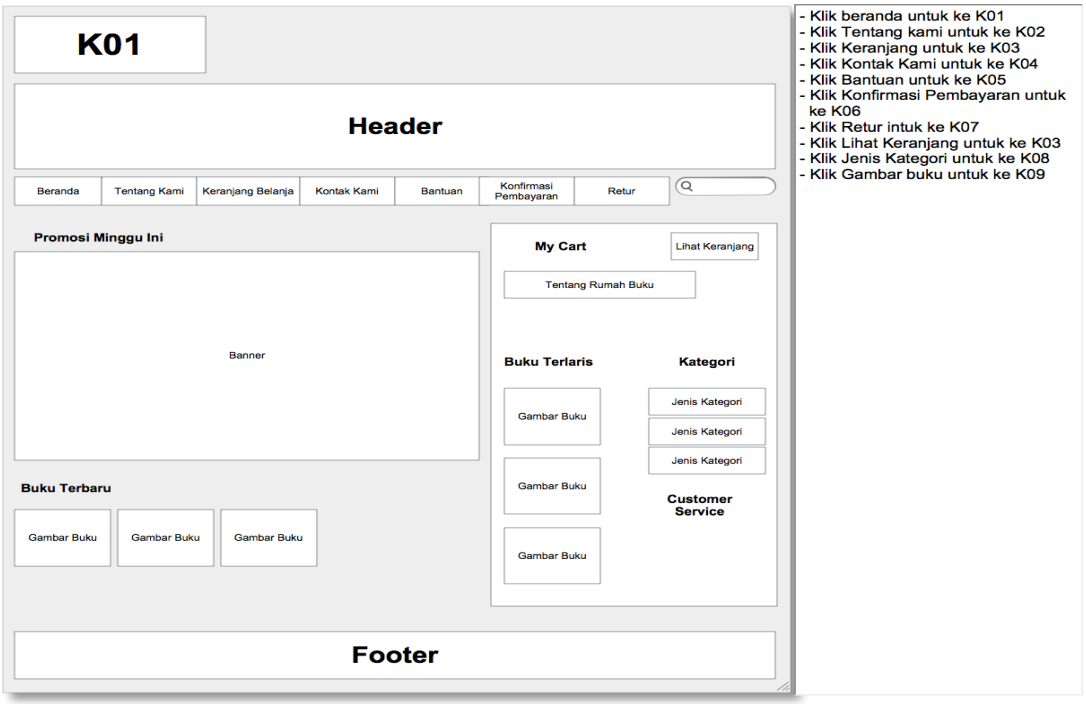
Spesifikasi antarmuka merupakan suatu bentuk tampilan dari program yang akan dibuat untuk kebutuhan *interface* dengan *user*. Spesifikasi antarmuka terdiri dari perancangan tampilan menu, tampilan form, tampilan pesan, keluaran dan jaringan semantik.

3.2.3.1 Perancangan Antarmuka Katalog (*Front End)*

Perancangan antarmuka *front end* terdiri dari perancangan halaman-halaman yang terhihat oleh pengunjung. Perancangan antarmuka *front end* terdiri dari :

1. Perancangan antarmuka beranda

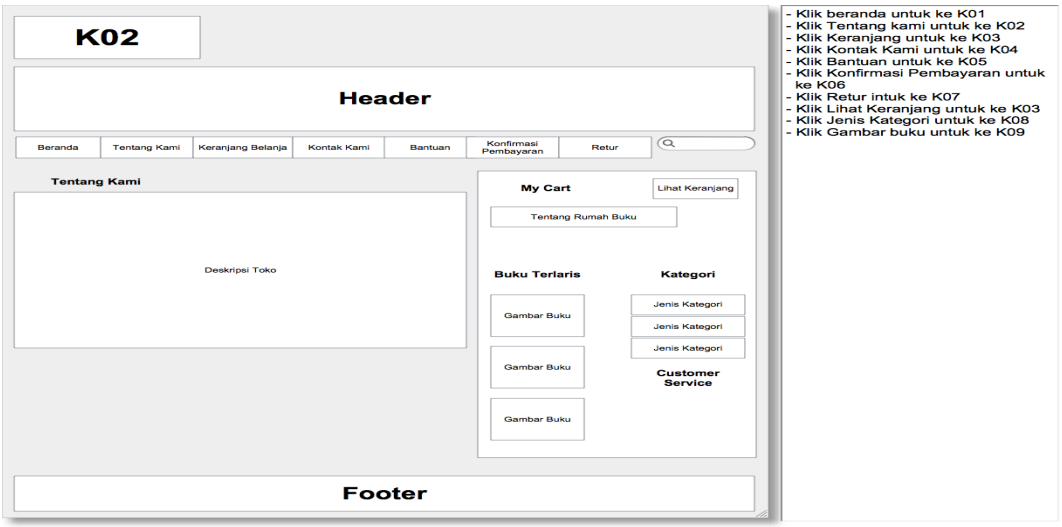
   Perancangan antarmuka beranda pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.38



Gambar 3.38 Perancangan antarmuka beranda

2. Perancangan antarmuka tentang kami

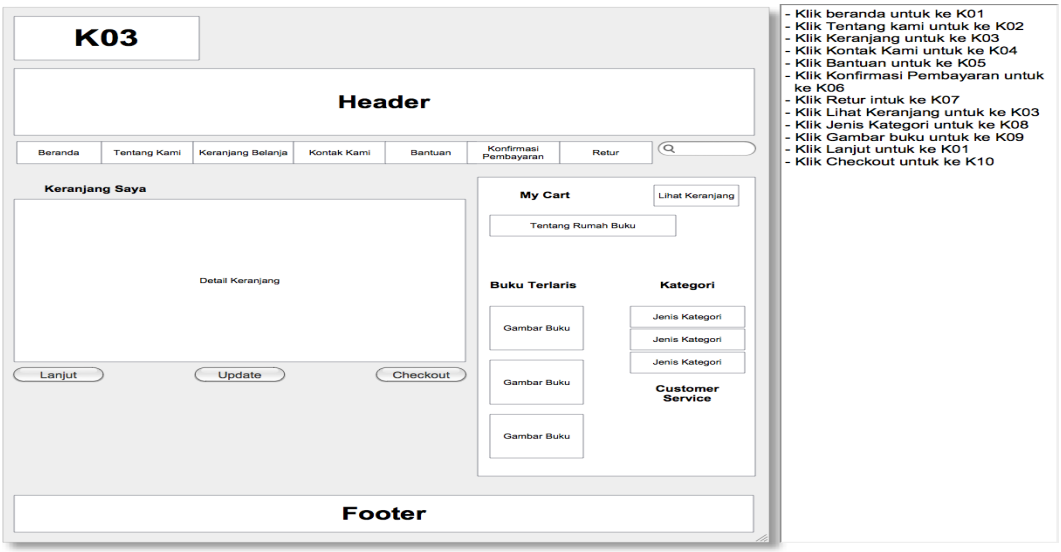
   Perancangan antarmuka tentang kami pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.39



Gambar 3.39 Perancangan antarmuka tentang kami

3. Perancangan antarmuka keranjang

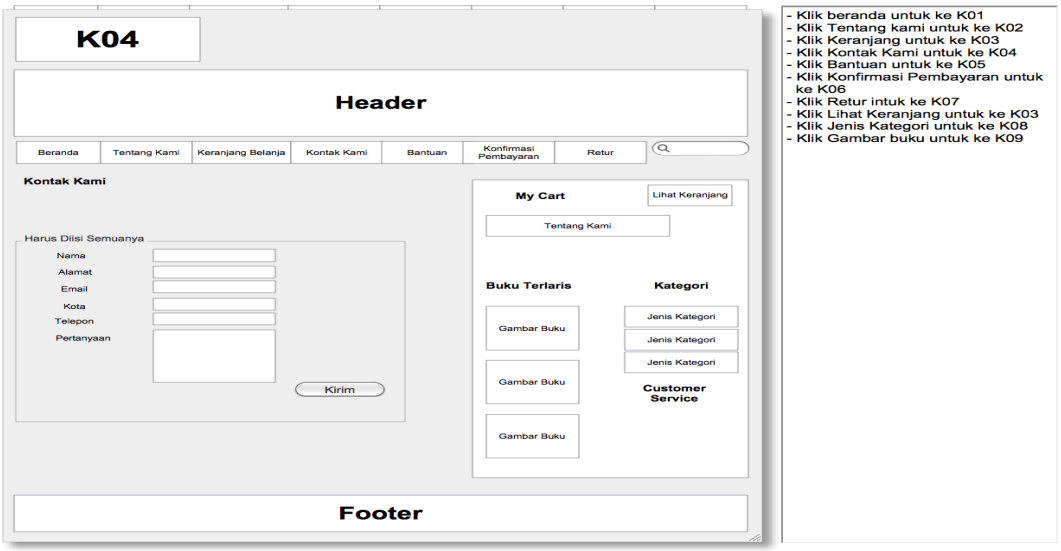
   Perancangan antarmuka kategori pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.40



Gambar 3.40 Perancangan antarmuka keranjang

4. Perancangan antarmuka kontak kami

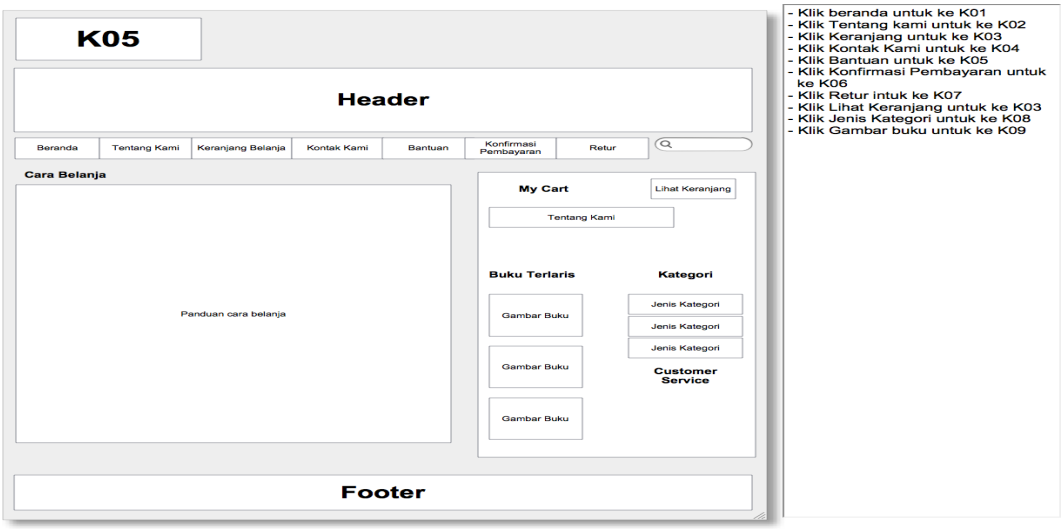
   Perancangan antamuka kontak kami pada aplikasi e-commerce Rumah Buku dapat dilihat pada gambar 3.41



Gambar 3.41 Perancangan antarmuka kontak kami

5. Perancangan antarmuka bantuan

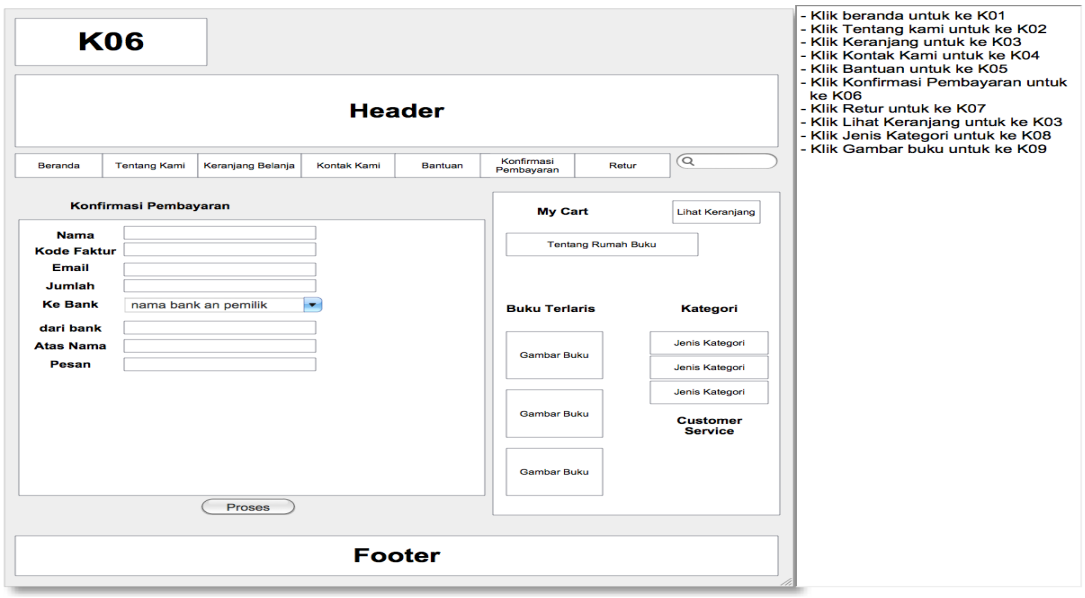
   Perancangan antarmuka bantuan pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.42



Gambar 3.42 Perancangan antarmuka bantuan

6. Perancangan antarmuka konfirmasi pembayaran

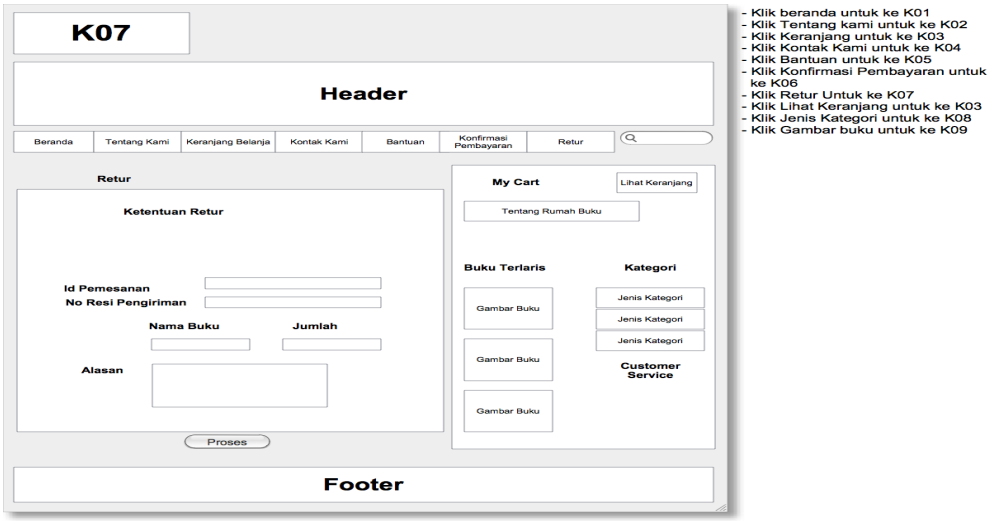
   Perancangan antarmuka konfirmasi pembayaran pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.43



Gambar 3.43 Perancangan antarmuka konfirmasi pembayaran

7. Perancangan antarmuka Retur

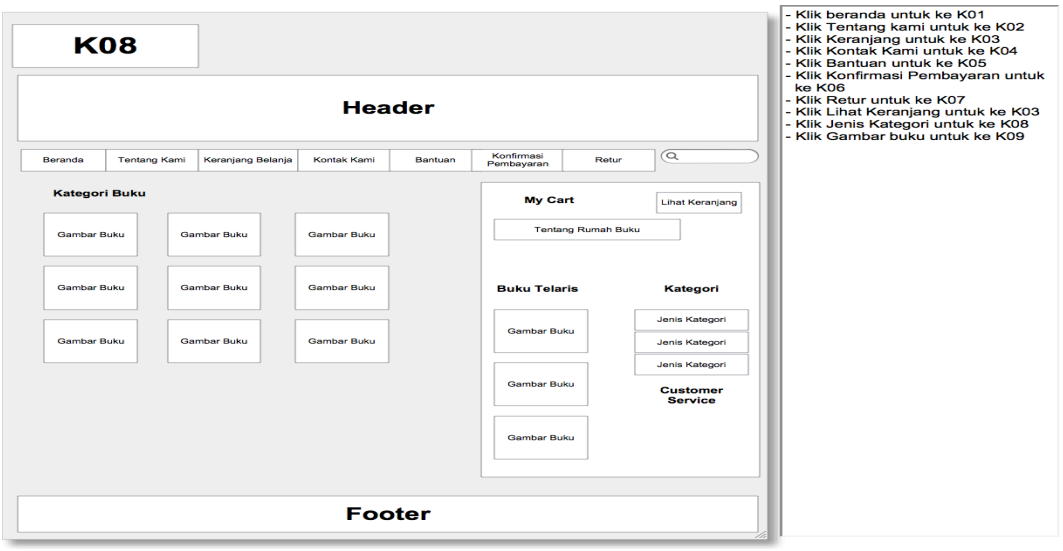
   Perancangan antarmuka retur buku pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.44



Gambar 3.44 Perancangan antarmuka retur

8. Perancangan antarmuka kategori buku

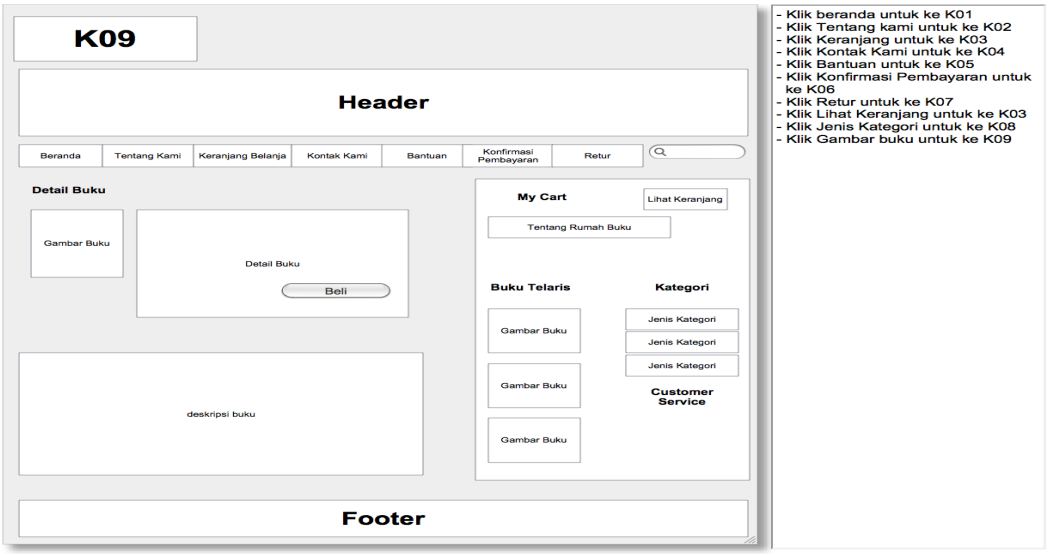
   Perancnagan antar muka kategori buku pada e-commerce Rumah Buku dapat dilihat pada gambar 3.45



Gambar 3.45 Peracangan antarmuka kategori buku

9. Perancangan antarmuka detail buku

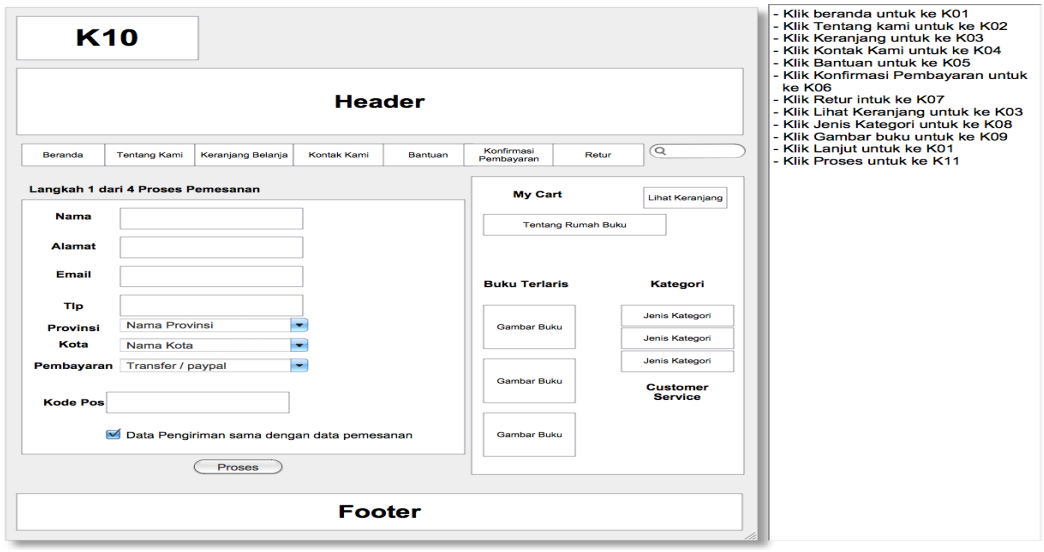
   Perancangan antarmuka detail buku pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.46



Gambar 3.46 Perancangan antarmuka detail buku

10. Perancangan antarmuka *Checkout* (langkah 1)

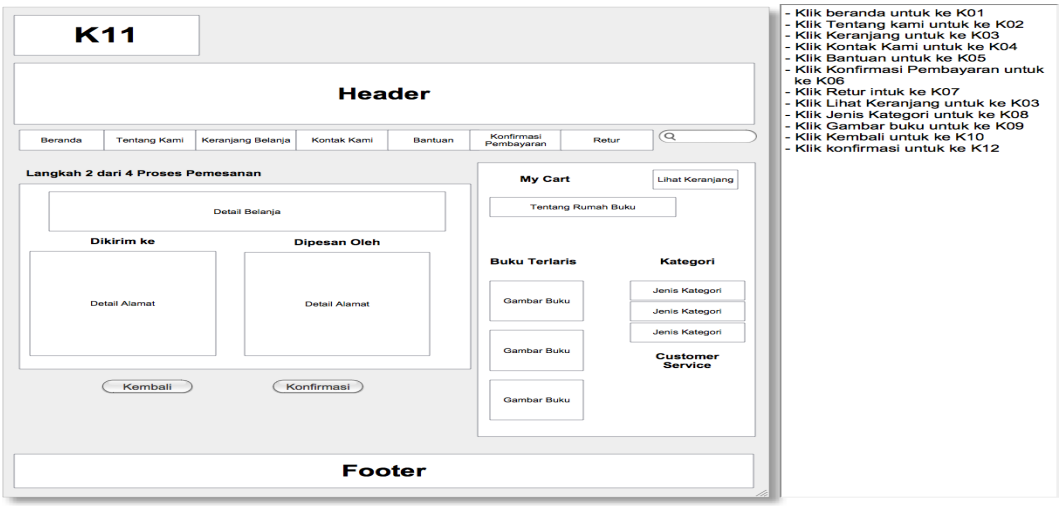
    Perancangan antarmuka *checkout* (langkah 1) pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.47



Gambar 3.47 Perancangan antarmuka *checkout* (langkah 1)

11. Perancangan antarmuka *Checkout* (langkah 2)

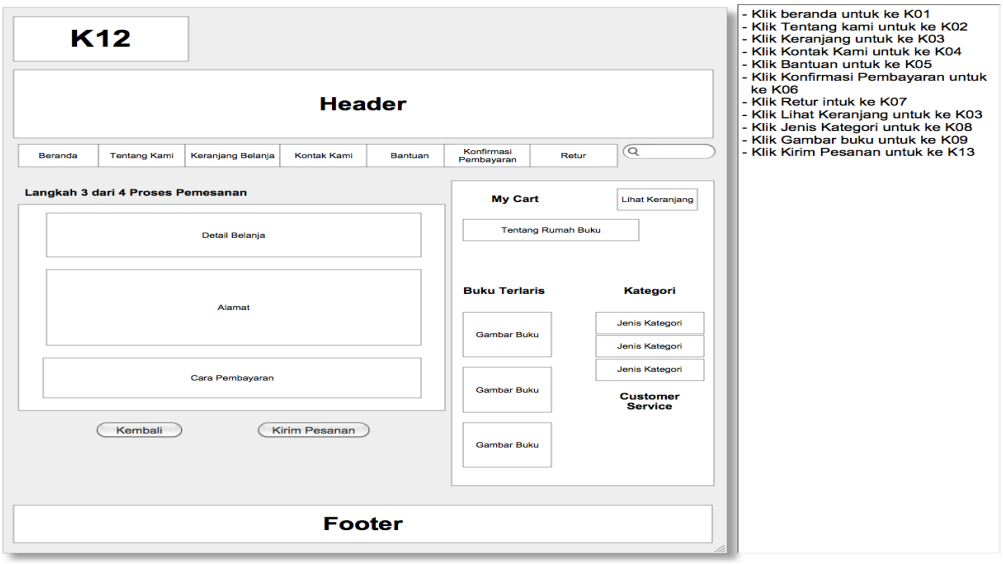
    Perancangan antarmuka *checkout* (langkah 2) pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.48



Gambar 3.48 Perancangan antarmuka *checkout* (langkah 2)

12. Perancangan antarmuka *Checkout* (langkah 3)

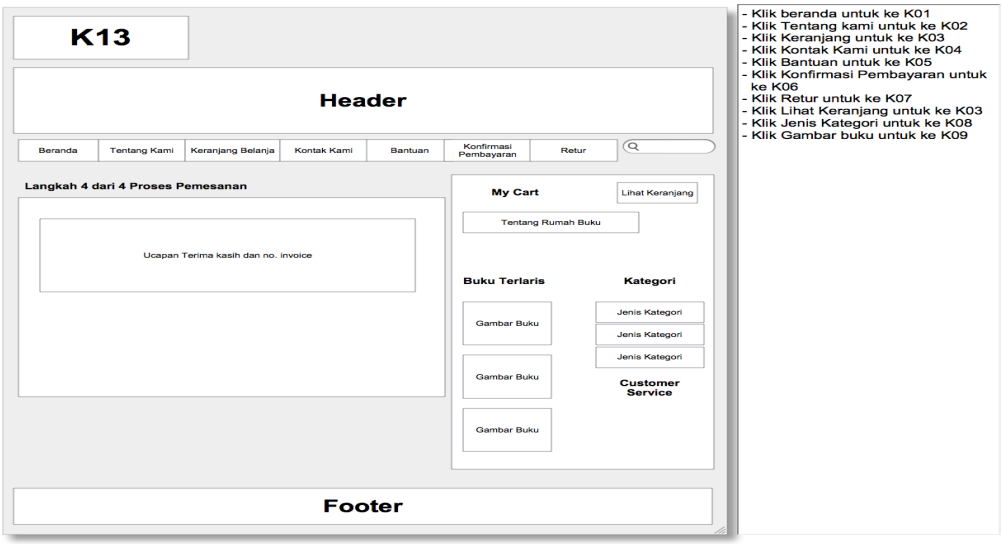
Perancangan antarmuka *checkout* (langkah 3) pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.49



Gambar 3.49 Perancangan antarmuka *checkout* (langkah 3)

13. Perancangan antarmuka *Checkout* (langkah 4)

Perancangan antarmuka *checkout* (langkah 4) pada *e-commerce* Rumah Buku dapat dilihat pada gambar 3.50



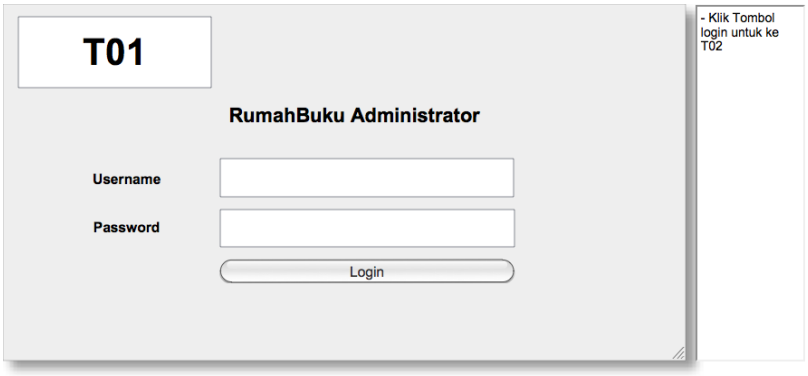
Gambar 3.50 Perancangan antarmuka *checkout* (langkah 4)

3.2.3.2 Perancangan Antarmuka *Back End (Admin)*

Perancangan antarmuka *admin* terdiri dari perancangan halaman-halaman yang terhihat oleh admin. Perancangan antarmuka *admin* terdiri dari :

1. Perancangan antarmuka *login* admin

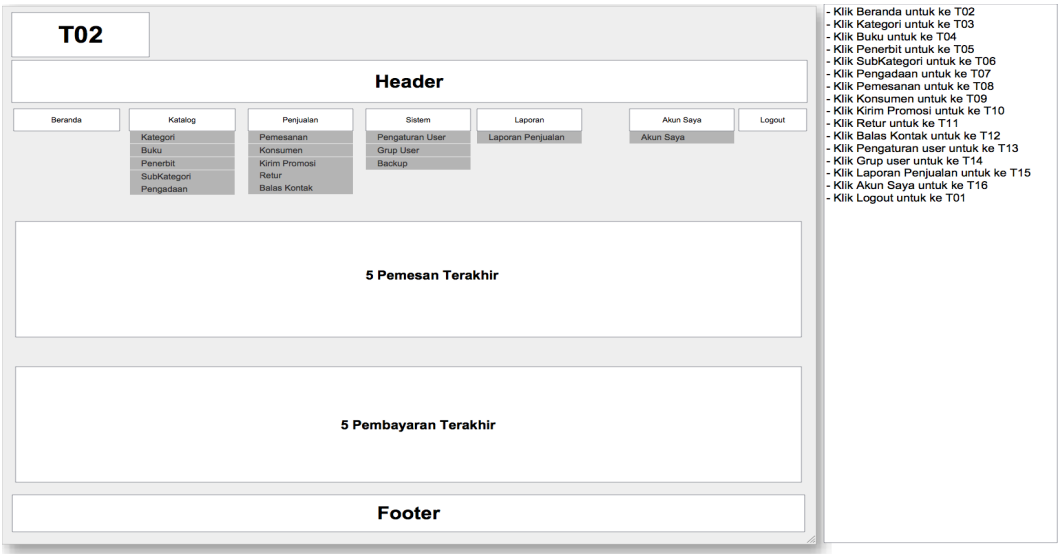
   Perancangan antarmuka login *admin admin* dapat dilihat pada gambar 3.51



Gambar 3.51 Perancangan antarmuka *login admin*

2. Perancangan antarmuka beranda

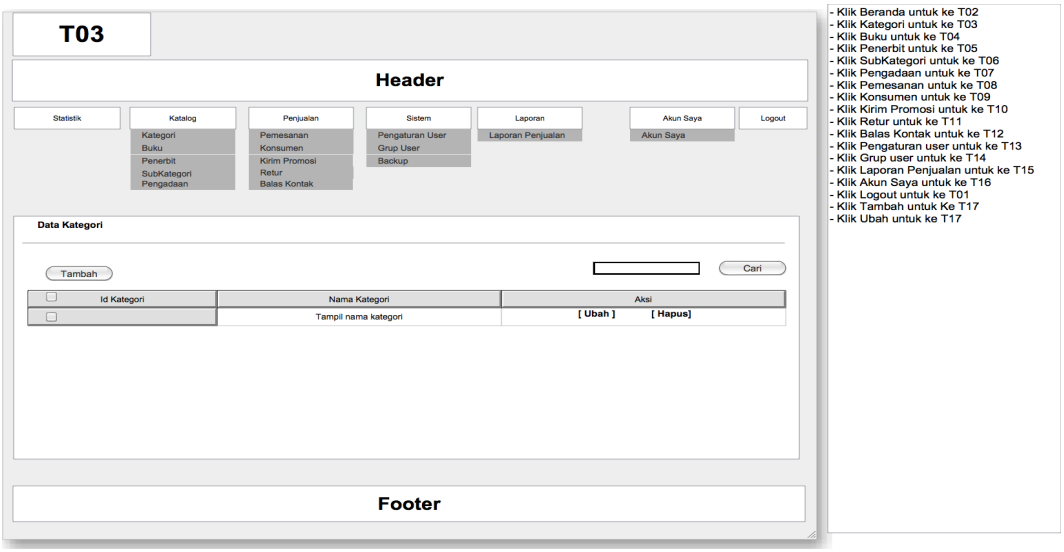
   Perancangan antarmuka statistik pada aplikasi *e-commerce* Rumah Buku dapat dilihta pada gambar 3.52



Gambar 3.52 Perancangan antarmuka beranda

3. Perancangan antarmuka kategori

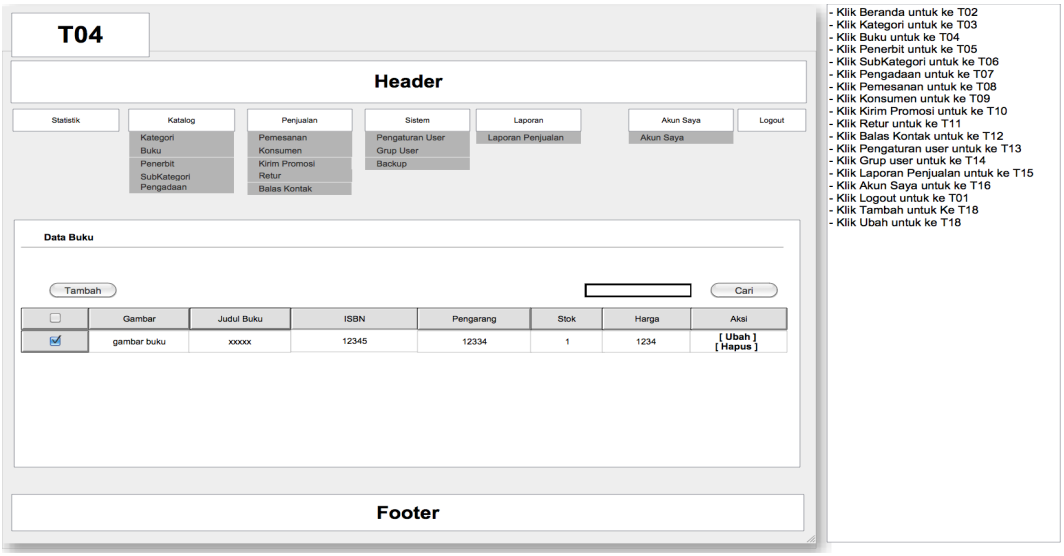
   Perancangan antarmuka kategori pada aplikasi *e-commerce* pada toko Rumah Buku dapat dilihat pada gambar 3.53



Gambar 3.53 Perancangan antarmuka kategori

4. Perancangan antarmuka buku

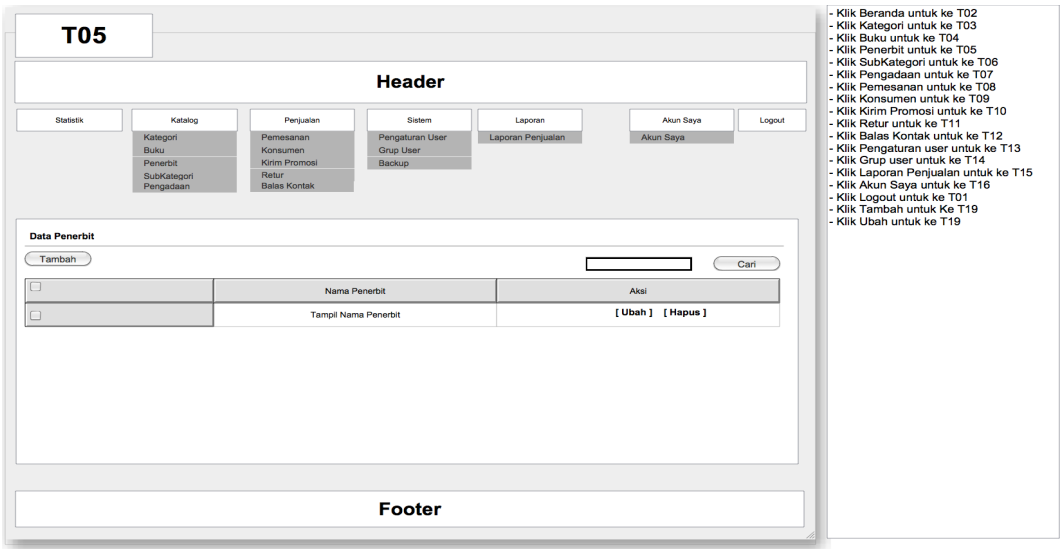
   Perancangan antarmuka buku pada aplikasi *e-commerce* toko Rumah Buku padat dilihat pada gambar 3.54



Gambar 3.54 Perancangan antarmuka buku

5. Perancangan antarmuka penerbit

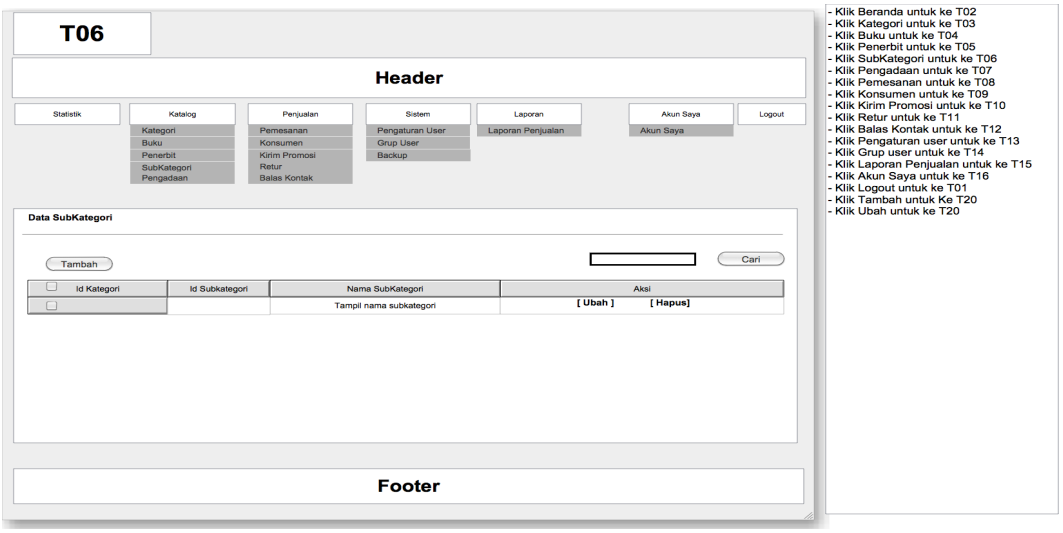
Perancangan antarmuka penerbit pada aplikasi *e-commerce* toko Rumah Buku dapat dilihat pada gambar 3.55



Gambar 3.55 Perancangan antarmuka penerbit

6. Perancangan antarmuka subkategori buku

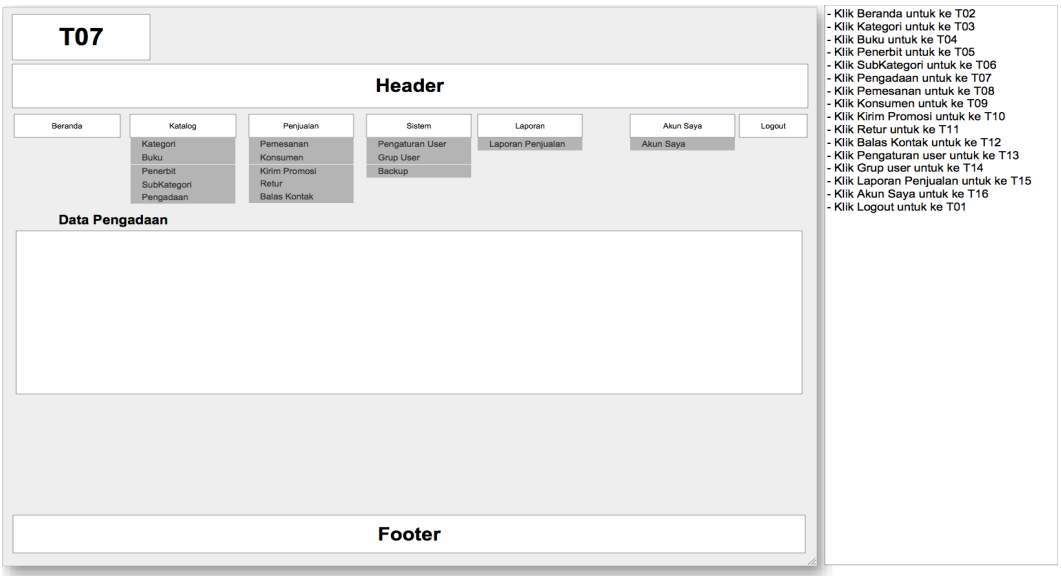
Perancangan antarmuka subkategori buku pada aplikasi *e-commerce* toko Rumah Buku dapat dilihat pada gambar 3.56



Gambar 3.56 Perancangan antarmuka subkategori buku

7. Perancangan antarmuka pengadaan buku

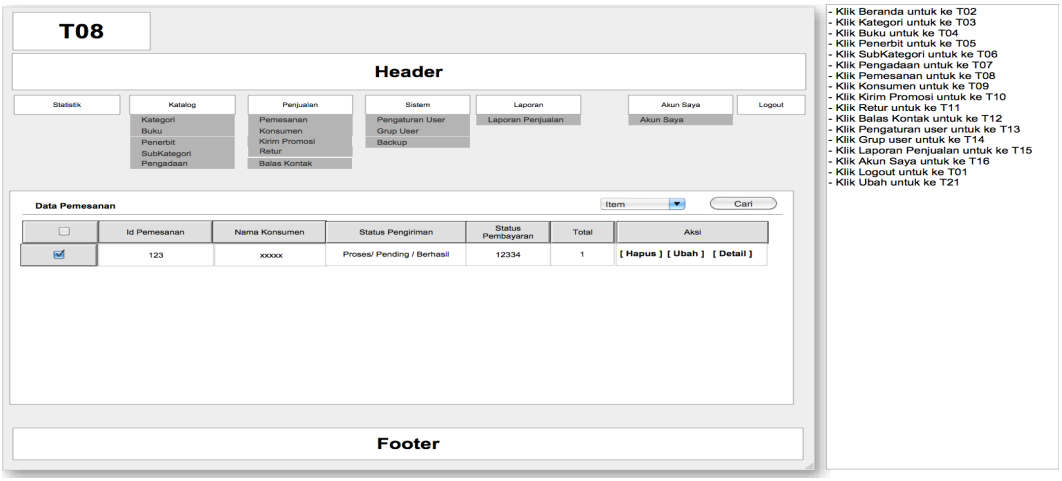
   Perancangan antarmuka pengadaan buku pada aplikasi *e-commerce* toko Rumah Buku dapat dilihat pada gambar 3.57



Gambar 3.57 Perancangan antarmuka pengadaan buku

8. Perancangan antarmuka pemesanan

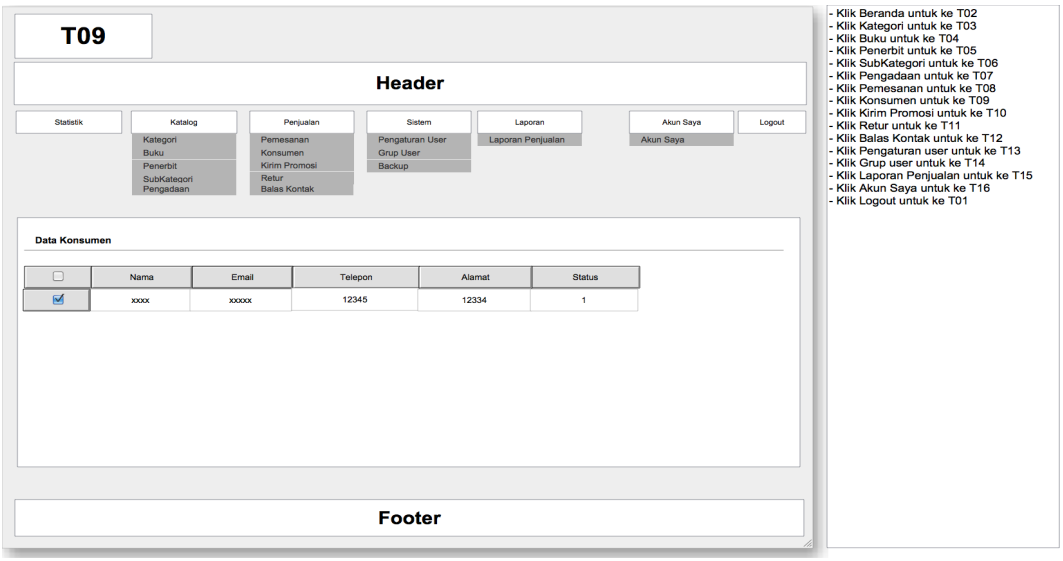
   Perancangan antarmuka pemesanan pada aplikasi *e-commerce* toko Rumah Buku dapat dilihat pada gambar 3.58



Gambar 3.58 Perancangan antarmuka pemesanan

9. Perancangan antarmuka konsumen

Perancangan antarmuka konsumen pada aplikasi *e-commerce* toko Rumah Buku dapat dilihat pada gambar 3.59



Gambar 3.59 Perancangan antarmuka konsumen

10. Perancangan antarmuka kirim promosi

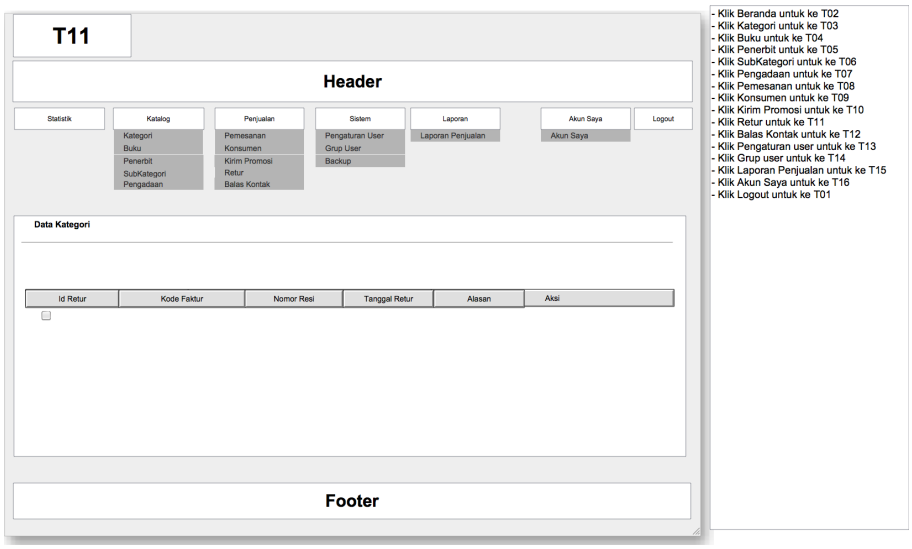
Perancangan antarmuka konfirmasi pembayaran pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.60



Gambar 3.60 Perancangan antarmuka kirim promosi

11. Perancangan antarmuka Retur

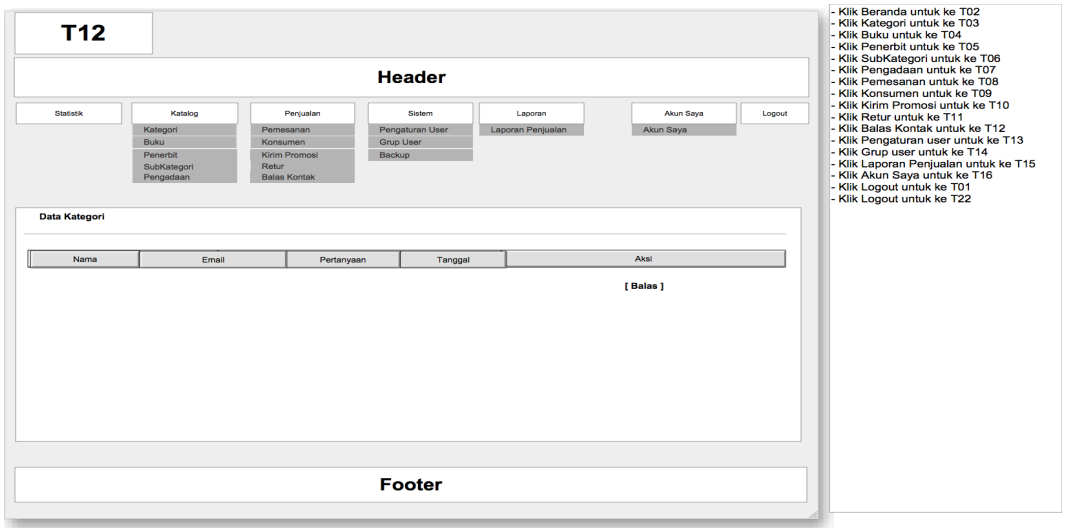
Perancangan antarmuka retur pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.61



Gambar 3.61 Perancangan antarmuka retur

12. Perancangan antarmuka balas kontak

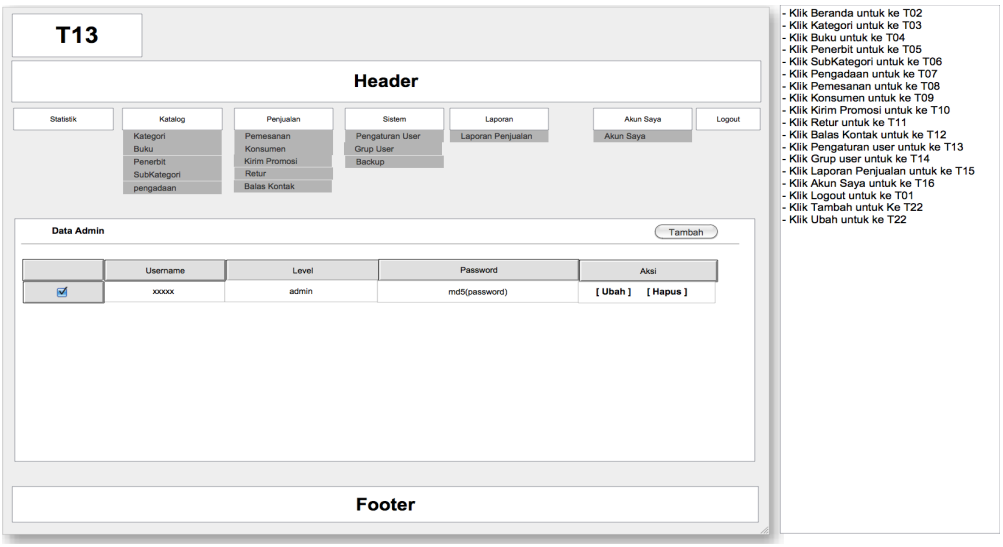
Perancangan antarmuka balas kontak pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.62



Gambar 3.62 Perancangan antarmuka balas kontak

13. Perancangan antarmuka pengaturan user

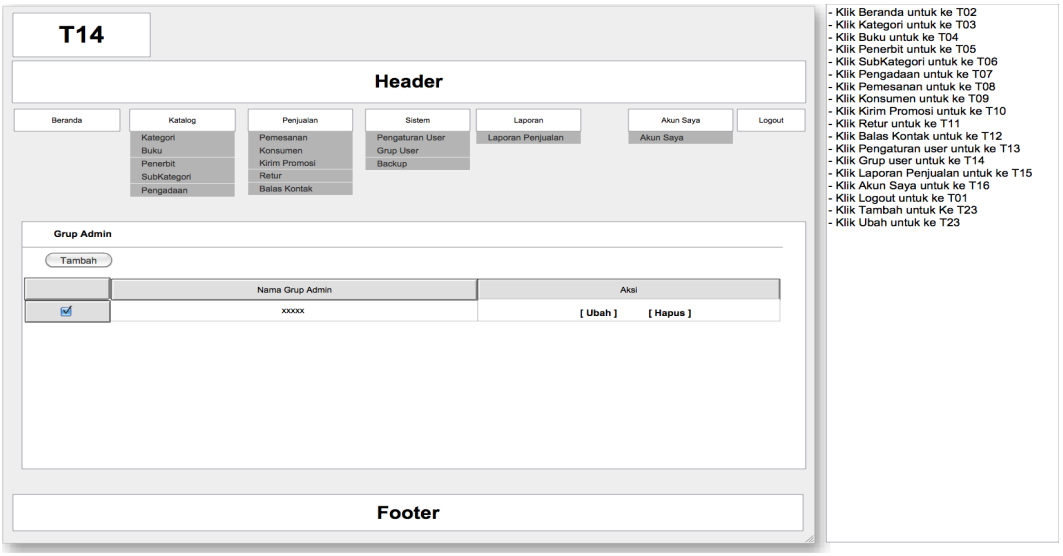
Perancangan antarmuka pengaturan admin pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.63



Gambar 3.63 Perancangan antarmuka pengaturan admin

14. Perancangan antarmuka grup admin

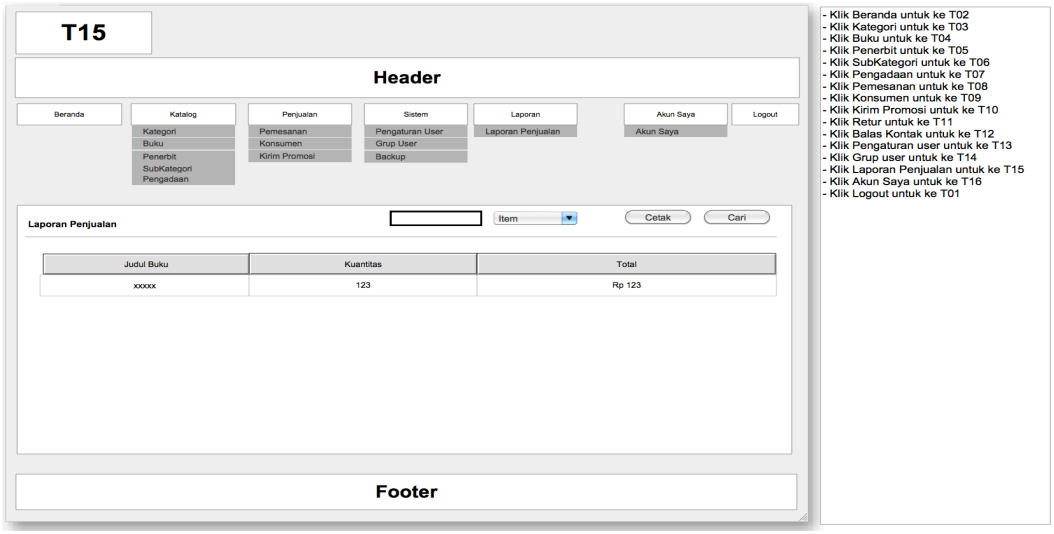
Perancangan antarmuka grup admin pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.64



Gambar 3.64 Perancangan antarmuka grup admin

15. Perancangan antarmuka laporan penjualan

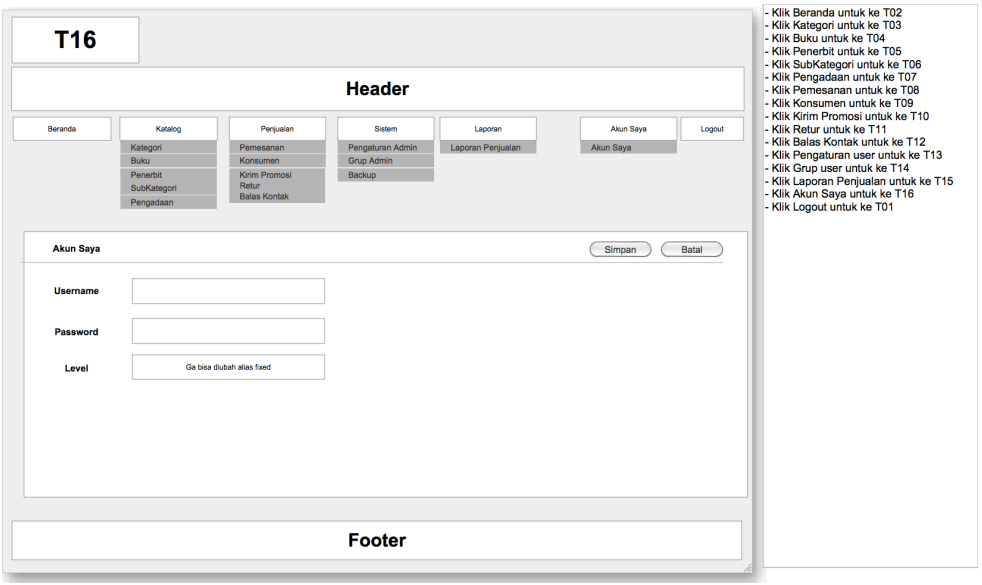
Perancangan antarmuka laporan penjualan pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.65



Gambar 3.65 Perancangan antarmuka laporan penjualan

16. Perancangan antarmuka akun saya

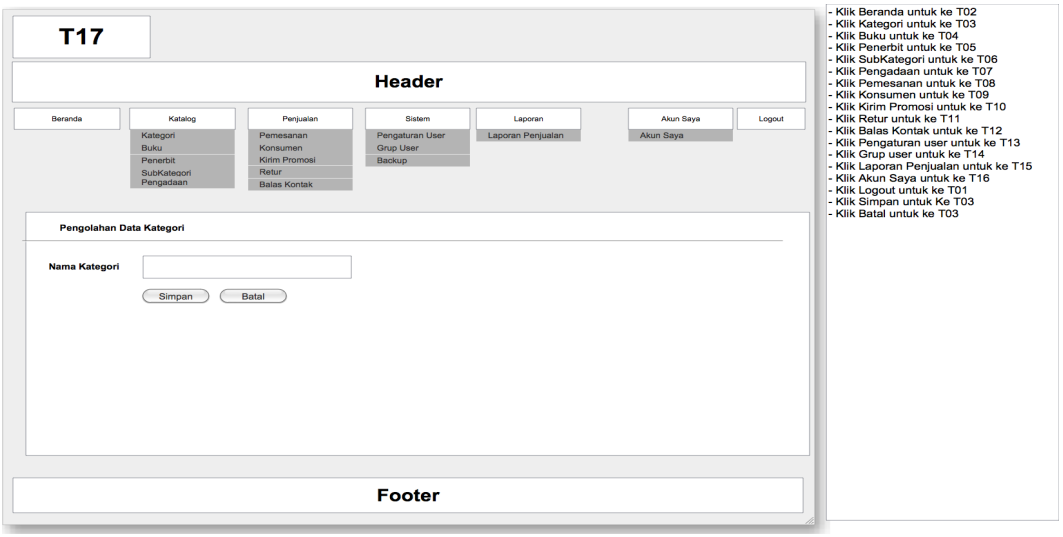
Perancangan antarmuka akun saya pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.66



Gambar 3.66 Perancangan antarmuka akun saya

17. Perancangan antarmuka pengolahan data kategori

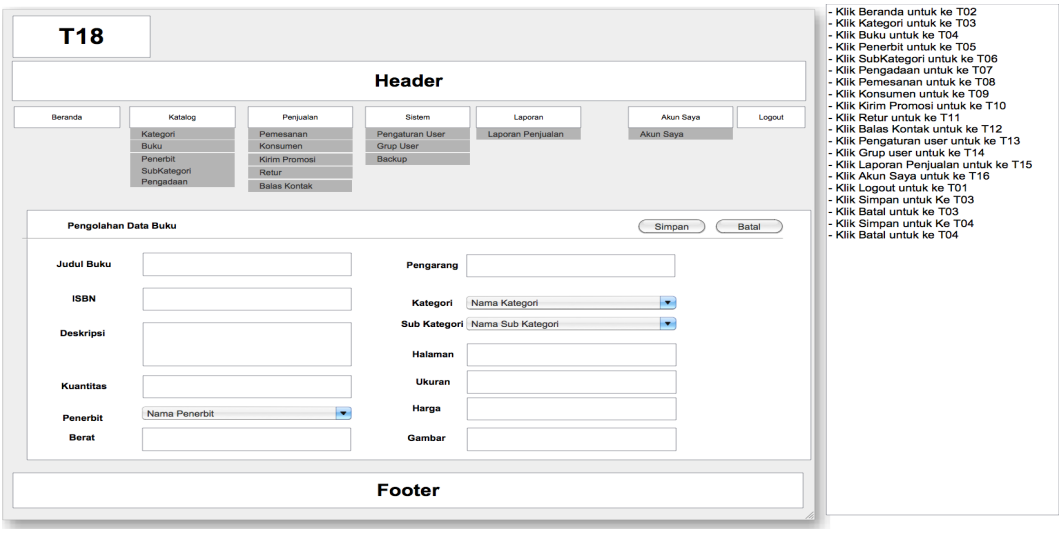
Perancangan antarmuka pengolahan data kategori pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.67



Gambar 3.67 Perancangan antaramuka pengolahan data kategori

18. Perancangan antarmuka pengolahan data buku

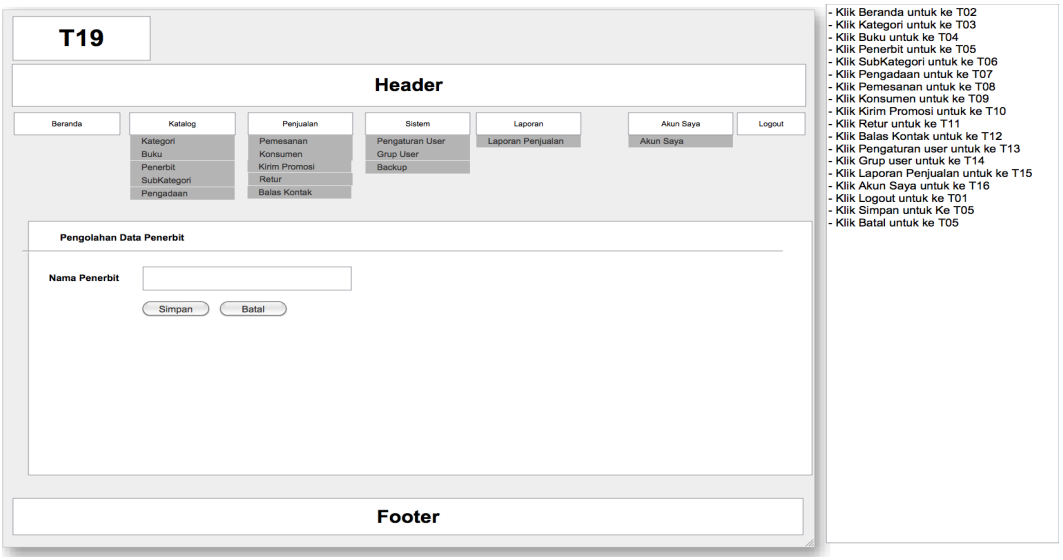
Perancangan antarmuka pengolahan data buku pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.68



Gambar 3.68 Perancangan antarmuka pengolahan data buku

19. Perancangan antarmuka pengolahan data penerbit

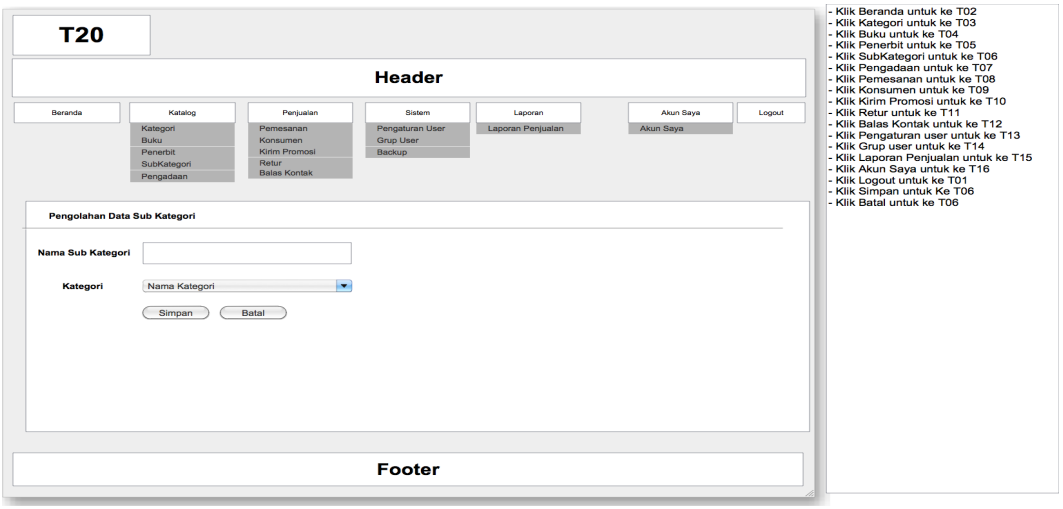
Perancangan antarmuka pengolahan data penerbit pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.69



Gambar 3.69 Perancangan antarmuka pengolahan data penerbit

20. Perancangan antarmuka pengolahan data subkategori

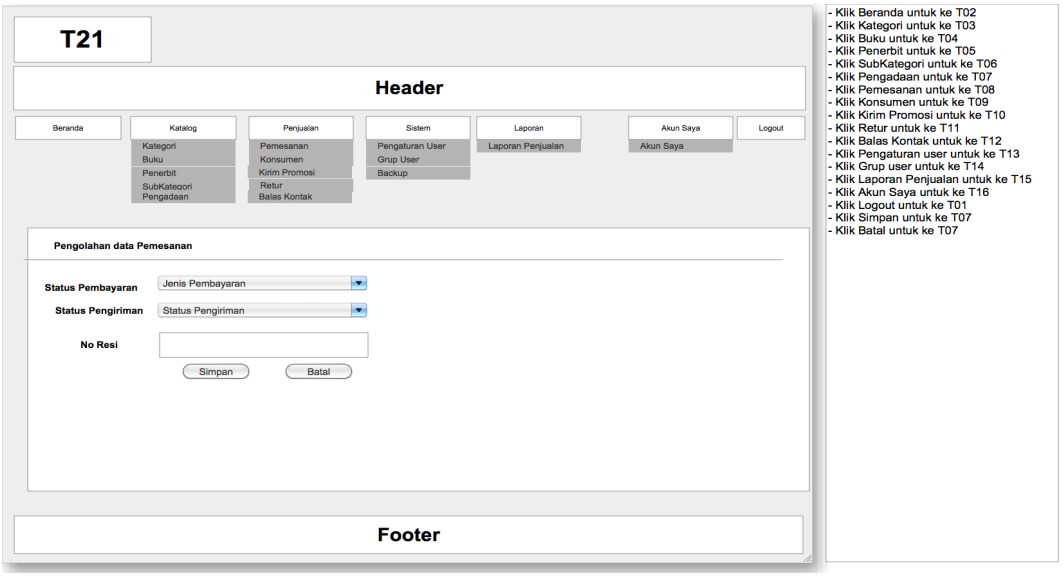
Perancangan antarmuka pengolahan data informasi pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.70



Gambar 3.70 Perancangan antarmuka pengolahan data subkategori

21. Perancangan antarmuka pengolahan data pemesanan

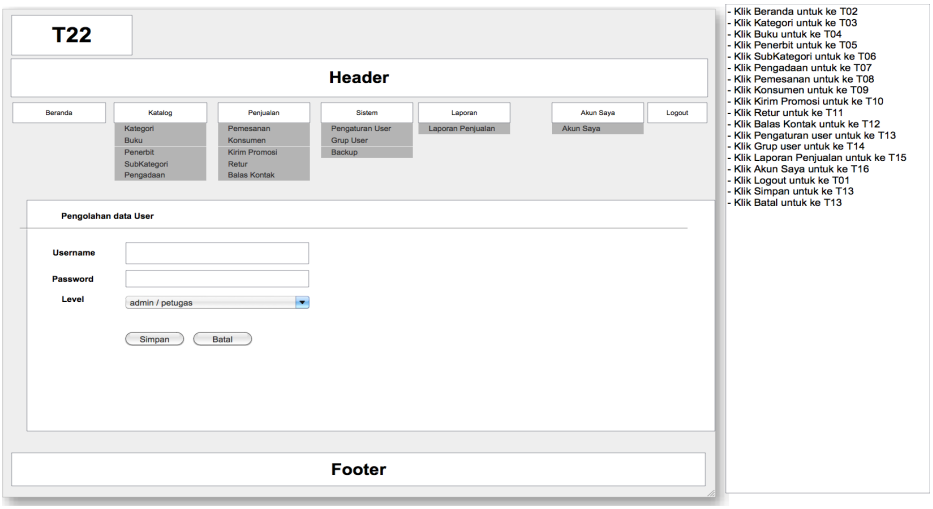
Perancangan antarmuka pengolahan data pesanan pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.71



Gambar 3.71 Perancangan antarmuka pengolahan data pemesanan

22. Perancangan antarmuka pengolahan data *user*

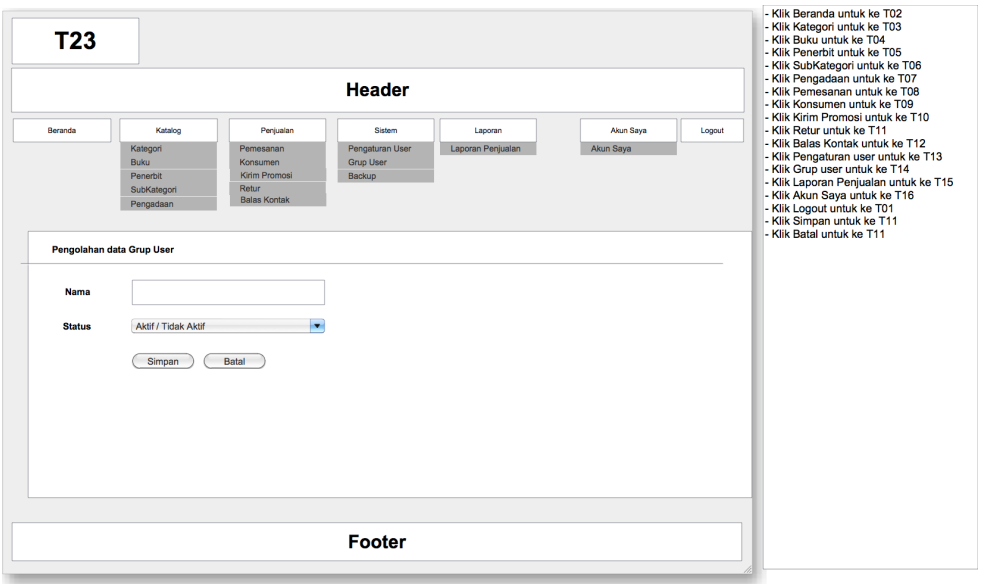
Perancangan antarmuka pengolahan data admin pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.72



Gambar 3.72 Perancangan antarmuka pengolahan data *user*

23. Perancangan antarmuka pengolahan data grup admin

Perancangan antarmuka pengolahan data grup admin pada aplikasi *e-commerce* Rumah buku dapat dilihat pada gambar 3.73



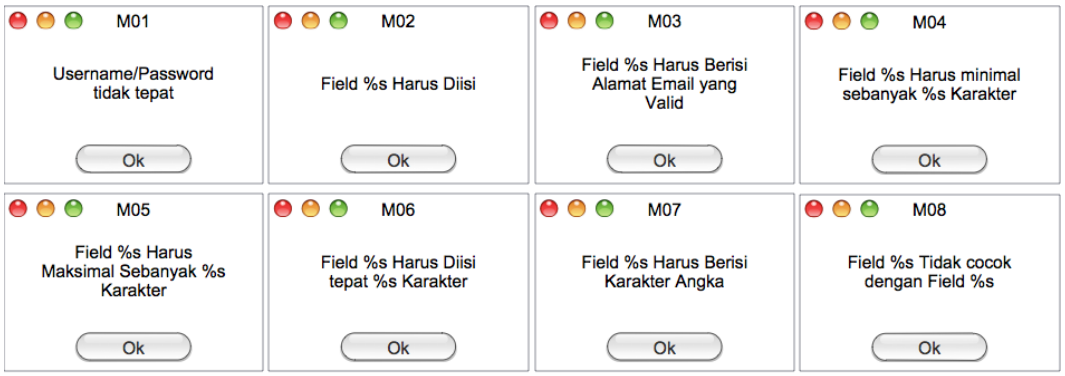
Gambar 3.73 Perancangan antarmuka pengolahan data grup *admin*

3.2.3.3 Perancangan Pesan

Perancangan pesan merupakan bahasa yang disampaikan oleh sistem kepada pengguna untuk menyampaikan informasi sesuai yang dilakukan oleh *user*. Tampilan pesan yang ada pada aplikasi *e-commerce* pada toko Rumah buku adalah sebagai berikut :

1. Perancangan pesan halaman *admin*

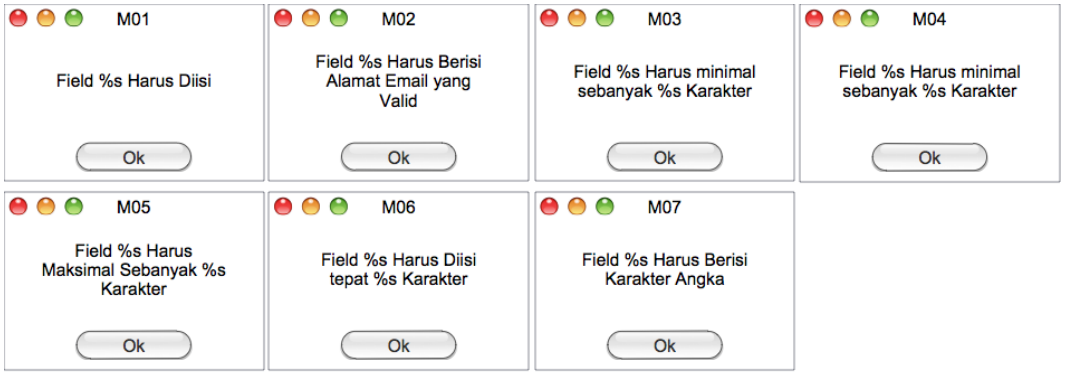
Perancangan pesan pada halaman *admin* dapat dilihat pada gambar pada gambar 3.74



Gambar 3.74 Perancangan pesan halaman *admin*

2. Perancangan pesan halaman katalog

   Perancangan pesan pada halaman katalog dapat dilihat pada gambar 3.75



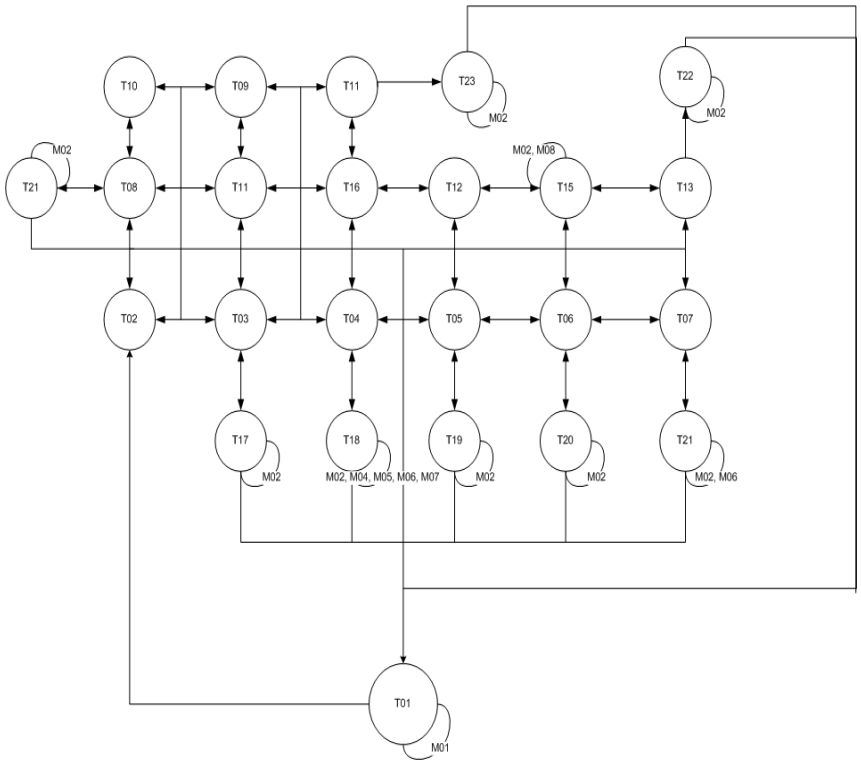
Gambar 3.75 Perancangan pesan halaman katalog

3.2.3.4 Jaringan Semantik

Jaringan semantik merupakan aliran dari menu-menu yang ada pada aplikasi yang dibangun. Jaringan semantik aplikasi *e-commerce* pada toko rumah buku adalah :

1. Jaringan semantik tampilan *admin*

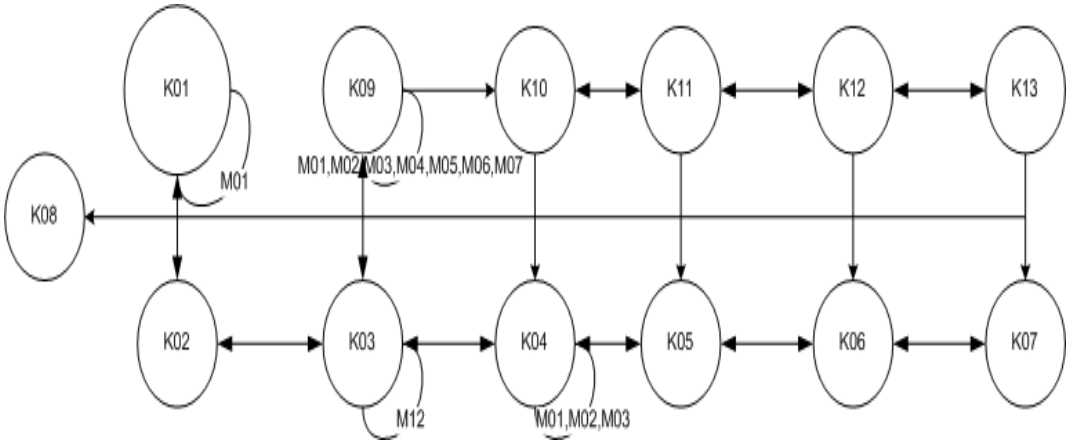
   Jaringan semantik tampilan admin menggambarkan jaringan tampilan pada halaman admin. Jaringan semantik tampilan admin dapat dilihat pada gambar 3.76



Gambar 3.76 Jaringan semantik tampilan *admin*

2. Jaringan semantik tampilan katalog

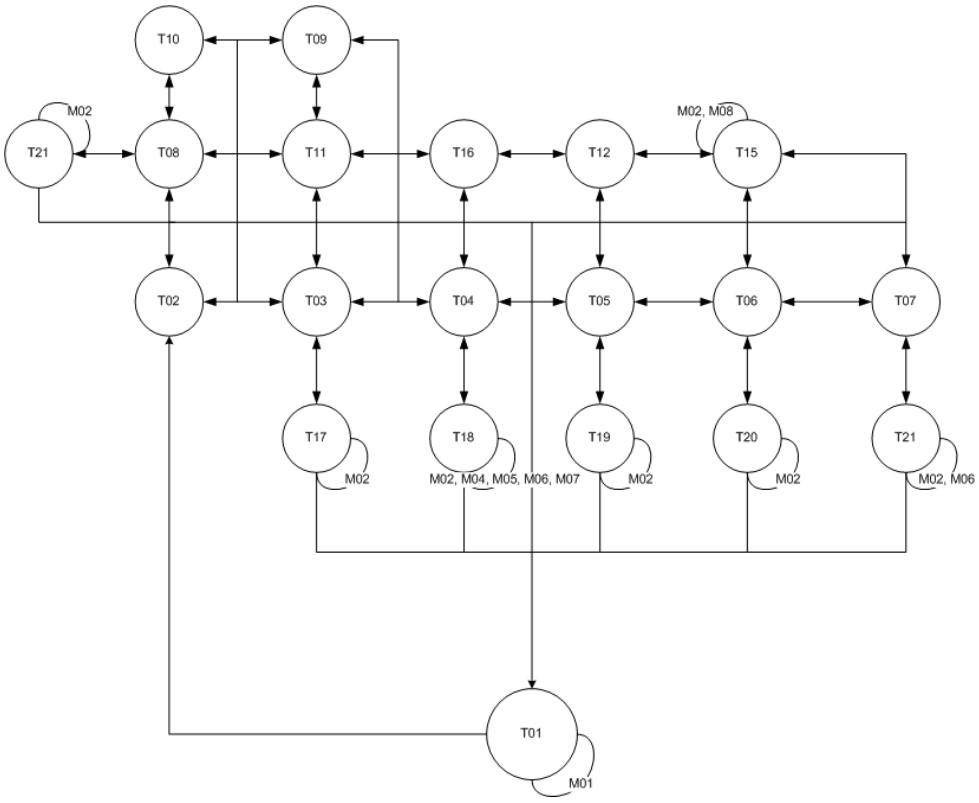
   Jaringan semantik tampilan katalog menggambarkan jaringan tampilan pada halaman katalog. Jaringan semantik tampilan katalog dapat dilihat pada gambar 3.77



Gambar 3.77 Jaringan semantik tampilan katalog

3. Jaringan semantik tampilan petugas

   Jaringan semantik tampilan petugas menggambarkan jaringan tampilan pada halaman petugas. Jaringan semantik tampilan petugas dapat dilihat pada gambar 3.78



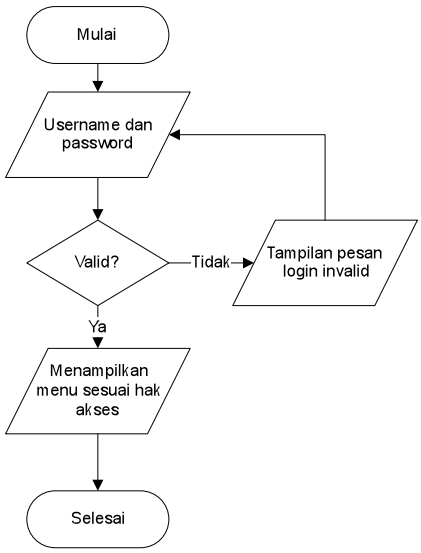
Gambar 3.78 Jaringan semantik tampilan petugas

3.2.4 Perancangan Prosedural

Perancangan prosedural adalah tata cara atau urutan langkah-langkah untuk melakukan suatu proses. Prosedural ini akan digunakan sebagai algoritma dasar dalam mengkodekan prosedur yang ada. *Tools* yang digunakan adalah flowchart program. Adapun perancangan prosedural yang ada pada aplikasi e-commerce pada toko Rumah Buku adalah sebagai berikut :

1. Flowchart *Login*

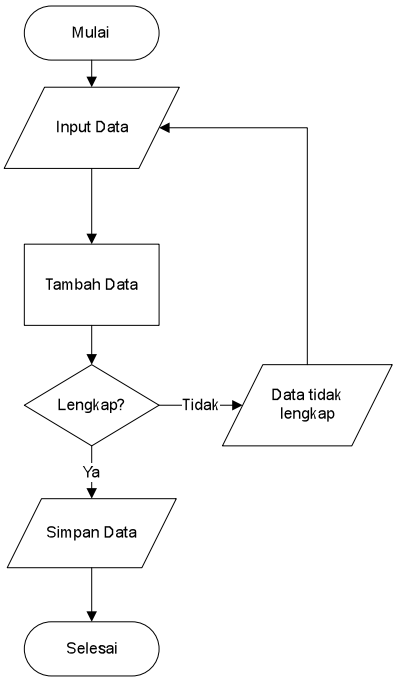
   Diagram alir pada prosedur *login* dapat dilihat pada gambar 3.79



Gambar 3.79 *Flowchart Login*

2. *Flowchart* tambah data

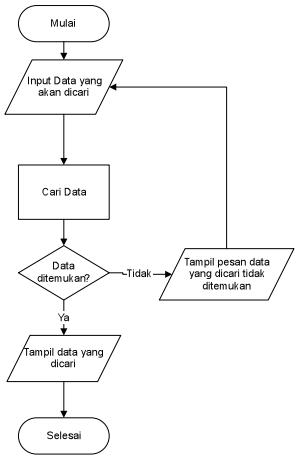
   Diagram alir pada prosedur tambah data dapat dilihat pada gambar 3.80



Gambar 3.80 Flowchart Tambah Data

3. *Flowchart* cari data

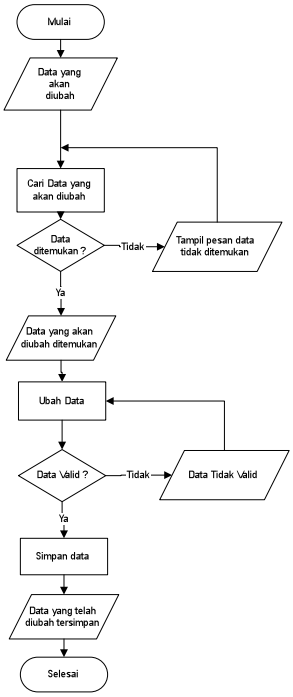
Diagram alir pada prosedur cari data dapat dilihat pada gambar 3.81



Gambar 3.81 *Flowchart* Cari Data

4. *Flowchart* ubah data

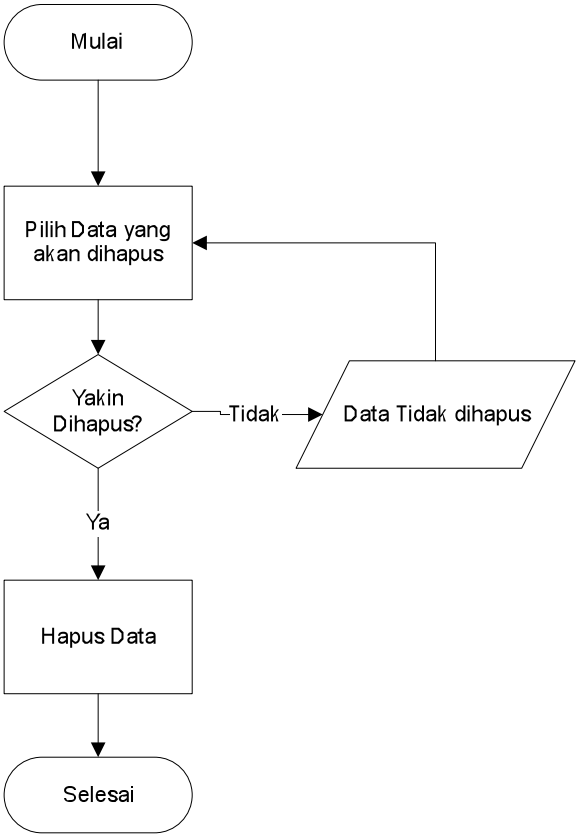
Diagram alir pada prosedur ubah data dapat dilihat pada gambar 3.82



Gambar 3.82 Flowchart Ubah Data

5. *Flowchart* hapus data

Diagram alir pada prosedur hapus data dapat dilihat pada gambar 3.83



Gambar 3.83 *Flowchart* Hapus Data