



SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

SITB

(Sistem Informasi Toko Buku)

Untuk :


Toko Buku XYZ

Dipersiapkan oleh:

Yohanes Yanuar Adi N / 110706607

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL SITB		1/81
		Revisi	-	

Program Studi Teknik Informatika	SKPL – SITB	1/81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	YYAN							
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	PENDAHULUAN.....	6
1.1	TUJUAN.....	6
1.2	LINGKUP MASALAH.....	6
1.3	DEFINISI, AKRONIM DAN SINGKATAN.....	7
1.4	REFERENSI.....	8
1.5	DESKRIPSI UMUM (OVERVIEW).....	8
2	DESKRIPSI KEBUTUHAN.....	9
2.1	PERSPEKTIF PRODUK.....	9
2.2	FUNGSI PRODUK.....	10
2.3	KARAKTERISTIK PENGGUNA.....	25
2.4	BATASAN-BATASAN.....	25
2.5	ASUMSI DAN KETERGANTUNGAN.....	26
3	KEBUTUHAN KHUSUS.....	26
3.1	KEBUTUHAN ANTARMUKA EKSTERNAL.....	26
3.1.1	<i>Antarmuka pemakai.....</i>	<i>26</i>
3.1.2	<i>Antarmuka perangkat keras.....</i>	<i>26</i>
3.1.3	<i>Antarmuka perangkat lunak.....</i>	<i>27</i>
3.1.4	<i>Antarmuka Komunikasi.....</i>	<i>28</i>
4	SPEKIFIKASI RINCI KEBUTUHAN.....	30
4.1	SPEKIFIKASI KEBUTUHAN FUNGSIONALITAS.....	30
5	ERD.....	80

Daftar Gambar

Gambar 1	Arsitektur Perangkat Lunak SITB.....	9
Gambar 2	Use Case Diagram SITB.....	29
Gambar 5.1	Entity Relationship Diagram.....	80
Gambar 5.2	Entity Relationship Diagram (lanjutan)....	81



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SITB (Sistem Informasi Toko Buku) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna) dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SITB ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak SITB dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan inventory dari beberapa gudang.
2. Menangani pengelolaan transaksi pembelian, retur pembelian, hutang, penjualan, retur penjualan, order, dan piutang yang terjadi pada Toko Buku XYZ.
3. Menangani pencatatan proses keluar dan masuk barang dari dan atau menuju gudang.
4. Menangani reporting.

Dan berjalan pada lingkungan dekstop Windows dan menggunakan Microsoft SQL Server 2005 sebagai DBMS.

Program Studi Teknik Informatika	SKPL – SITB	6/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-SITB-XXX	Kode merepresentasikan kebutuhan SITB (Sistem Informasi Toko Buku) dimana XXX merupakan nomor fungsi produk.
SITB	Perangkat lunak yang dikembangkan untuk menangani pengelolaan proses transaksi Toko Buku XYZ.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
MULAI	Status pengerjaan order dimana order baru diterima oleh toko buku XYZ.
PROSES	Status pengerjaan order dimana order sedang dikerjakan oleh toko buku XYZ.
SELESAI	Status pengerjaan order dimana order telah selesai dikerjakan toko buku XYZ.
Barcode	Kode yang berupa kombinasi garis hitam dan putih dengan berbagai kombinasi jarak untuk mengkodekan angka atau huruf.
Barcode scanner	Alat yang digunakan untuk membaca data yang ada didalam kode barcode tersebut dan mengirimkan data tersebut ke komputer.

1.4 Referensi

Referensi yang digunakan pada SKPL ini adalah:

1. Damanik, Rudolfo Rizki. *Spesifikasi Kebutuhan Perangkat Lunak (SKPL) Automatic Report System (ARS)*, Universitas Atma Jaya Yogyakarta, 2012.
2. Michael Aristian S, *Spesifikasi Kebutuhan Perangkat Lunak (SKPL) LaSerS*, Universitas Atma Jaya Yogyakarta, 2012.
3. Nugroho, Yohanes Yanuar Adi. *Spesifikasi Kebutuhan Perangkat Lunak (SKPL) Sistem Informasi Kana Jaya (SIKAJA)*, Universitas Atma Jaya Yogyakarta, 2013.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian pertama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SITB yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SITB tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SITB yang akan dikembangkan.

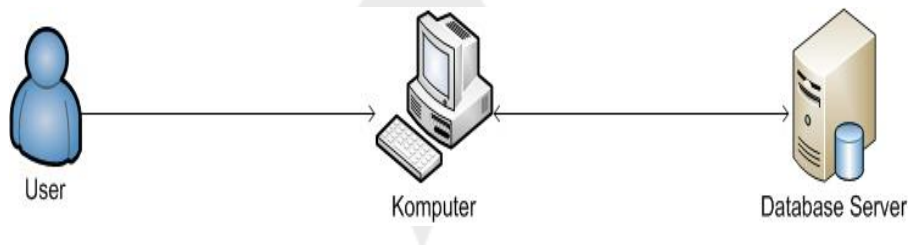
2 Deskripsi Kebutuhan

2.1 Perspektif produk

SITB merupakan perangkat lunak yang dikembangkan untuk membantu pengelolaan sistem informasi pada Toko Buku XYZ. Sistem akan digunakan oleh pihak internal toko buku saja. Secara umum sistem ini menangani pengelolaan data transaksional yang terjadi pada Toko Buku XYZ. Selain itu juga menangani pencatatan keluar atau masuk barang dari dan atau ke gudang.

Perangkat lunak SITB ini merupakan aplikasi yang berjalan pada desktop dengan platform Windows 7 dan dibuat menggunakan bahasa pemrograman Microsoft Visual C#. Sedangkan untuk lingkungan pemrogramannya menggunakan Microsoft Visual Studio 2008 dan menggunakan Microsoft SQL Server 2005 sebagai basis data.

Arsitektur perangkat lunak SITB ditunjukkan seperti pada gambar 1. Perangkat lunak ini digunakan oleh user melalui antarmuka pada komputer. Semua data yang ada disimpan di dalam server. User dapat mengakses data yang ada di server secara localhost.



Gambar 1 Arsitektur Perangkat Lunak SITB

2.2 Fungsi Produk

Fungsi produk perangkat lunak SITB adalah sebagai berikut :

APLIKASI DEKSTOP

1. Fungsi *Login* (**SKPL-SITB-001**).

Merupakan fungsi yang digunakan oleh semua user dari Toko Buku XYZ untuk dapat masuk dalam sistem dan memperoleh hak akses sesuai dengan peran yang di miliki user.

2. Fungsi *Pengelolaan Data User* (**SKPL-SITB-002**).

Merupakan fungsi yang digunakan oleh owner untuk mengelola data pengguna dari sistem SITB. Fungsi *Pengelolaan Data User* mencakup :

a. Fungsi *Entri Data User* (**SKPL-SITB-002-01**).

Merupakan fungsi yang digunakan untuk menambahkan data user.

b. Fungsi *Edit Data User* (**SKPL-SITB-002-02**).

Merupakan fungsi yang digunakan untuk mengubah data user.

c. Fungsi *Delete Data User* (**SKPL-SITB-002-03**).

Merupakan fungsi yang digunakan untuk menghapus data user.

d. Fungsi *Display Data User* (**SKPL-SITB-002-04**).

Merupakan fungsi yang digunakan untuk menampilkan data user.

e. Fungsi *Cari Data User* (**SKPL-SITB-002-05**).

Merupakan fungsi yang digunakan administrator untuk mencari data user berdasarkan nama pengguna atau role yang di masukkan.

Program Studi Teknik Informatika	SKPL – SITB	10/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Fungsi *Pengelolaan Data Customer* (**SKPL-SITB-003**).

Merupakan fungsi yang digunakan owner dan atau manager untuk mengelola data customer dari Toko Buku XYZ. Fungsi *Pengelolaan Data Customer* mencakup :

a. Fungsi *Entri Data Customer* (**SKPL-SITB-003-01**).

Merupakan fungsi yang digunakan untuk menambahkan data customer yang baru.

b. Fungsi *Edit Data Customer* (**SKPL-SITB-003-02**).

Merupakan fungsi yang digunakan untuk mengubah data customer.

c. Fungsi *Delete Data Customer* (**SKPL-SITB-003-03**).

Merupakan fungsi yang digunakan untuk menghapus data customer.

d. Fungsi *Display Data Customer* (**SKPL-SITB-003-04**).

Merupakan fungsi yang digunakan untuk menampilkan data customer.

e. Fungsi *Cari Data Customer* (**SKPL-SITB-003-05**).

Merupakan fungsi yang digunakan untuk mencari data customer berdasarkan kode customer, nama customer, atau jenis customer.

4. Fungsi *Pengelolaan Data Supplier* (**SKPL-SITB-004**).

Merupakan fungsi yang digunakan owner untuk mengelola data jasa pelayanan yang ditawarkan bengkel Kana Jaya. Fungsi *Pengelolaan Data Supplier* meliputi:

Program Studi Teknik Informatika	SKPL – SITB	11/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

a. Fungsi *Entri Data Supplier* (**SKPL-SITB-004-01**).

Merupakan fungsi yang digunakan untuk menambah data supplier dari Toko Buku XYZ.

b. Fungsi *Edit Data Supplier* (**SKPL-SITB-004-02**).

Merupakan fungsi untuk mengubah atau mengganti data supplier.

c. Fungsi *Delete Data Supplier* (**SKPL-SITB-004-03**).

Merupakan fungsi untuk menghapus data supplier.

d. Fungsi *Display Data Supplier* (**SKPL-SITB-004-04**).

Merupakan fungsi yang digunakan untuk menampilkan data supplier.

e. Fungsi *Cari Data Supplier* (**SKPL-SITB-004-05**).

Merupakan fungsi yang digunakan untuk mencari data supplier berdasarkan kode supplier dan nama supplier.

5. Fungsi *Pengelolaan Jenis Customer* (**SKPL-SITB-005**).

Merupakan fungsi yang digunakan owner dan atau manager untuk mengelola jenis customer. Fungsi *Pengelolaan Jenis Customer* meliputi:

a. Fungsi *Entri Data Jenis Customer* (**SKPL-SITB-005-01**).

Merupakan fungsi yang digunakan untuk menambah data jenis customer.

Program Studi Teknik Informatika	SKPL – SITB	12/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

b. Fungsi *Edit Data Jenis Customer* (**SKPL-SITB-005-02**) .

Merupakan fungsi yang digunakan untuk mengganti/ mengubah data jenis customer.

c. Fungsi *Delete Data Jenis Customer* (**SKPL-SITB-005-03**) .

Merupakan fungsi yang digunakan untuk menghapus data jenis customer.

d. Fungsi *Display Data Jenis Customer* (**SKPL-SITB-005-04**) .

Merupakan fungsi yang digunakan untuk menampilkan data jenis customer.

6. Fungsi *Pengelolaan Data Kategori Barang* (**SKPL-SITB-006**) .

Merupakan fungsi yang digunakan owner untuk mengelola data kategori barang. Fungsi *Pengelolaan Data Kategori Barang* meliputi:

a. Fungsi *Entri Data Kategori Barang* (**SKPL-SITB-006-01**) .

Merupakan fungsi yang digunakan untuk menambah data kategori barang.

b. Fungsi *Edit Data Kategori Barang* (**SKPL-SITB-006-02**) .

Merupakan fungsi untuk mengubah atau mengganti data kategori barang.

c. Fungsi *Delete Data Kategori Barang* (**SKPL-SITB-006-03**) .

Merupakan fungsi untuk menghapus data kategori barang.

d. Fungsi *Display Data Kategori Barang* (**SKPL-SITB-006-04**) .

Merupakan fungsi yang digunakan untuk menampilkan data kategori barang.

e. Fungsi *Cari Data Kategori Barang* (**SKPL-SITB-006-05**) .

Merupakan fungsi yang digunakan untuk mencari data kategori barang berdasarkan deskripsi kategori barang.

7. Fungsi *Pengelolaan Data Barang* (**SKPL-SITB-007**) .

Merupakan fungsi yang digunakan owner untuk mengelola data barang. Fungsi *Pengelolaan Data Barang* meliputi:

a. Fungsi *Entri Data Barang* (**SKPL-SITB-007-01**) .

Merupakan fungsi yang digunakan untuk menambah data barang.

b. Fungsi *Edit Data Barang* (**SKPL-SITB-007-02**) .

Merupakan fungsi untuk mengubah atau mengganti data barang.

c. Fungsi *Delete Data Barang* (**SKPL-SITB-007-03**) .

Merupakan fungsi untuk menghapus data barang.

d. Fungsi *Display Data Barang* (**SKPL-SITB-007-04**).

Merupakan fungsi yang digunakan untuk menampilkan data barang.

e. Fungsi *Cari Data Barang* (**SKPL-SITB-007-05**).

Merupakan fungsi yang digunakan untuk mencari data barang berdasarkan kode barang, nama barang, kategori barang, atau nama supplier barang.

f. Fungsi *Generate Barcode* (**SKPL-SITB-007-06**).

Merupakan fungsi yang digunakan untuk generate gambar barcode dari kode barang.

8. Fungsi *Pengelolaan Data Jasa* (**SKPL-SITB-008**).

Merupakan fungsi yang digunakan owner untuk mengelola data jasa. Fungsi *Pengelolaan Data Jasa* meliputi:

a. Fungsi *Entri Data Jasa* (**SKPL-SITB-008-01**).

Merupakan fungsi yang digunakan untuk menambah data jasa.

b. Fungsi *Edit Data Jasa* (**SKPL-SITB-008-02**).

Merupakan fungsi untuk mengubah atau mengganti data jasa.

c. Fungsi *Delete Data Jasa* (**SKPL-SITB-008-03**).

Merupakan fungsi untuk menghapus data jasa.

d. Fungsi *Display Data Jasa* (**SKPL-SITB-008-04**).

Merupakan fungsi yang digunakan untuk menampilkan data jasa.

Program Studi Teknik Informatika	SKPL – SITB	15/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

e. Fungsi *Cari Data Jasa* (**SKPL-SITB-008-05**).

Merupakan fungsi yang digunakan untuk mencari data jasa berdasarkan kode jasa atau nama jasa.

9. Fungsi *Pengelolaan Data Lokasi* (**SKPL-SITB-009**).

Merupakan fungsi yang digunakan owner dan atau manager untuk mengelola data lokasi gudang/ toko. Fungsi *Pengelolaan Data Lokasi* meliputi:

a. Fungsi *Entri Data Lokasi* (**SKPL-SITB-009-01**).

Merupakan fungsi yang digunakan untuk menambah data lokasi.

b. Fungsi *Edit Data Lokasi* (**SKPL-SITB-009-02**).

Merupakan fungsi untuk mengubah atau mengganti data lokasi.

c. Fungsi *Delete Data Lokasi* (**SKPL-SITB-009-03**).

Merupakan fungsi untuk menghapus data lokasi.

d. Fungsi *Display Data Lokasi* (**SKPL-SITB-009-04**).

Merupakan fungsi yang digunakan untuk menampilkan data lokasi.

e. Fungsi *Cari Data Lokasi* (**SKPL-SITB-009-05**).

Merupakan fungsi yang digunakan untuk mencari data lokasi berdasarkan kode lokasi, nama lokasi, alamat, atau jenis lokasi.

10. Fungsi *Pengelolaan Data Ketentuan Harga* (**SKPL-SITB-010**).

Merupakan fungsi yang digunakan owner untuk mengelola ketentuan harga jual barang. Dalam

Program Studi Teknik Informatika	SKPL – SITB	16/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

kasus ini harga jual barang selain ditentukan barangnya ditentukan juga berdasarkan jumlah barang yang dibeli dan jenis customer yang membeli. Fungsi *Pengelolaan Ketentuan Harga* meliputi:

a. Fungsi *Entri Data Ketentuan Harga* (**SKPL-SITB-010-01**).

Merupakan fungsi yang digunakan untuk menambah data ketentuan harga.

b. Fungsi *Edit Data Ketentuan Harga* (**SKPL-SITB-010-02**).

Merupakan fungsi untuk mengubah atau mengganti data ketentuan harga barang.

c. Fungsi *Delete Data Ketentuan Harga* (**SKPL-SITB-010-03**).

Merupakan fungsi untuk menghapus data ketentuan harga.

d. Fungsi *Display Data Ketentuan Harga* (**SKPL-SITB-010-04**).

Merupakan fungsi yang digunakan untuk menampilkan data ketentuan harga.

e. Fungsi *Cari Data Ketentuan Harga* (**SKPL-SITB-010-05**).

Merupakan fungsi yang digunakan untuk mencari data ketentuan harga berdasarkan nama barang atau jenis customer.

Program Studi Teknik Informatika	SKPL – SITB	17/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

11. Fungsi *Pencarian History Stock* (**SKPL-SITB-011**) .

Merupakan fungsi yang digunakan oleh owner untuk melakukan pencarian history dari stock barang.

12. Fungsi *Pencarian Stock Barang* (**SKPL-SITB-012**) .

Merupakan fungsi yang digunakan oleh semua user untuk melakukan pencarian stock barang yang tersedia.

13. Fungsi *Cetak Label Barang* (**SKPL-SITB-013**) .

Merupakan fungsi yang digunakan oleh owner untuk dapat mencetak label barang.

14. Fungsi *Transaksi Pembelian* (**SKPL-SITB-014**) .

Merupakan fungsi yang digunakan oleh owner untuk mencatat pembelian yang dilakukan. Transaksi Pembelian ini sudah termasuk didalamnya detil pembelian. Fungsi transaksi pembelian meliputi :

a. Fungsi *Mengecek Nomor Faktur Pembelian* (**SKPL-SITB-014-01**) .

Merupakan fungsi untuk mengecek nomor faktur pembelian sebelum data transaksi pembelian dibuat. Nomor faktur pembelian tidak boleh ada yang sama.

b. Fungsi *Menyimpan Data Transaksi Pembelian ke Tabel Pembelian* (**SKPL-SITB-014-02**) .

Merupakan fungsi untuk menyimpan data pembelian ke dalam tabel pembelian.

c. Fungsi *Insert Data Detil Pembelian* (**SKPL-SITB-014-03**) .

Merupakan fungsi yang digunakan untuk menambahkan data detil pembelian barang.

d. Fungsi *Hapus Data Detil Pembelian* **(SKPL-SITB-014-04)** .

Merupakan fungsi yang digunakan untuk menghapus data detil pembelian.

e. Fungsi *Display Data Detil Pembelian* **(SKPL-SITB-014-05)** .

Merupakan fungsi yang digunakan untuk menampilkan data detil pembelian.

15. Fungsi *Transaksi Retur Pembelian Keluar* **(SKPL-SITB-015)** .

Merupakan fungsi yang digunakan oleh owner untuk mencatat transaksi retur pembelian keluar, dimana barang yang sudah dibeli toko buku XYZ dikembalikan kepada supplier. Pengembalian barang ini tidak bergantung pada transaksi pembelian, namun hanya bergantung pada barang dan supplier. Fungsi transaksi retur pembelian keluar ini meliputi :

a. Fungsi *Menyimpan Data Transaksi Retur Pembelian Keluar ke Tabel Retur Pembelian* **(SKPL-SITB-015-01)** .

Merupakan fungsi untuk menyimpan data retur pembelian keluar ke dalam tabel retur pembelian.

b. Fungsi *Insert Data Detil Retur Pembelian Keluar* **(SKPL-SITB-015-02)** .

Program Studi Teknik Informatika	SKPL – SITB	19/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan untuk menambahkan data detail retur pembelian keluar barang.

c. Fungsi *Hapus Data Detil Retur Pembelian* Keluar **(SKPL-SITB-015-03)** .

Merupakan fungsi yang digunakan untuk menghapus data detail retur pembelian keluar.

d. Fungsi *Display Data Detil Retur Pembelian* Keluar **(SKPL-SITB-015-04)** .

Merupakan fungsi yang digunakan untuk menampilkan data detail retur pembelian keluar.

16. Fungsi *Transaksi Retur Pembelian Masuk* **(SKPL-SITB-016)** .

Merupakan fungsi yang digunakan owner untuk mencatat transaksi retur pembelian masuk. Transaksi yang dimaksud adalah saat barang yang diretur keluar sudah dikembalikan oleh supplier kepada toko buku XYZ. Fungsi transaksi retur pembelian masuk ini meliputi :

a. Fungsi *Menyimpan Data Transaksi Retur Pembelian Masuk ke Tabel Retur Pembelian* **(SKPL-SITB-016-01)** .

Merupakan fungsi untuk menyimpan data retur pembelian masuk ke dalam tabel retur pembelian.

b. Fungsi *Insert Data Detil Retur Pembelian Masuk* **(SKPL-SITB-016-02)** .

Merupakan fungsi yang digunakan untuk menambahkan data detail retur pembelian masuk barang.

Program Studi Teknik Informatika	SKPL – SITB	20/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

c. Fungsi *Hapus Data Detil Retur Pembelian Masuk* (**SKPL-SITB-016-03**) .

Merupakan fungsi yang digunakan untuk menghapus data detil retur pembelian masuk.

d. Fungsi *Display Data Detil Retur Pembelian Masuk* (**SKPL-SITB-016-04**) .

Merupakan fungsi yang digunakan untuk menampilkan data detil retur pembelian masuk.

17. Fungsi *Transaksi Pembayaran Hutang* (**SKPL-SITB-017**) .

Merupakan fungsi yang digunakan owner untuk mencatat data pembayaran hutang atas suatu transaksi pembelian yang dilakukan oleh toko buku XYZ. Pembayaran atas suatu hutang dapat dilakukan dalam beberapa kali pembayaran.

18. Fungsi *Transaksi Penjualan* (**SKPL-SITB-018**) .

Merupakan fungsi yang digunakan oleh semua user untuk mencatat transaksi penjualan yang terjadi. Fungsi transaksi penjualan meliputi :

a. Fungsi *Generate Nomor Faktur Penjualan* (**SKPL-SITB-018-01**) .

Merupakan fungsi yang digunakan untuk generate nomor faktur penjualan. Nomor faktur penjualan mempunyai format FJBXXXXXX/MM/YY dimana XXXXXX adalah 6 digit nomor urut transaksi, MM adalah 2 digit angka bulan terjadi transaksi dan YY adalah 2 digit angka tahun terjadi transaksi.

b. Fungsi *Menyimpan Data Transaksi Penjualan ke Tabel Penjualan* (**SKPL-SITB-018-02**) .

Program Studi Teknik Informatika	SKPL – SITB	21/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi untuk menyimpan data penjualan ke dalam tabel penjualan.

c. Fungsi *Insert Data Detil Penjualan* (**SKPL-SITB-018-03**).

Merupakan fungsi yang digunakan untuk menambahkan data detil penjualan barang.

d. Fungsi *Hapus Data Detil Penjualan* (**SKPL-SITB-018-04**).

Merupakan fungsi yang digunakan untuk menghapus data detil penjualan.

e. Fungsi *Display Data Detil Penjualan* (**SKPL-SITB-018-05**).

Merupakan fungsi yang digunakan untuk menampilkan data detil penjualan.

19. Fungsi *Transaksi Retur Penjualan* (**SKPL-SITB-019**).

Merupakan fungsi yang digunakan semua user untuk mencatat transaksi retur penjualan. Transaksi terjadi saat ada pembeli toko buku XYZ yang melakukan retur barang. Fungsi transaksi retur penjualan meliputi :

a. Fungsi *Menyimpan Data Transaksi Retur Penjualan ke Tabel Retur Penjualan* (**SKPL-SITB-019-01**).

Merupakan fungsi untuk menyimpan data retur penjualan ke dalam tabel retur penjualan.

b. Fungsi *Insert Data Detil Retur Penjualan* (**SKPL-SITB-019-02**).

Merupakan fungsi yang digunakan untuk menambahkan data detil retur penjualan barang.

c. Fungsi *Hapus Data Detil Retur Penjualan* (**SKPL-SITB-019-03**) .

Merupakan fungsi yang digunakan untuk menghapus data detil retur penjualan.

d. Fungsi *Display Data Detil Retur Penjualan* (**SKPL-SITB-019-04**) .

Merupakan fungsi yang digunakan untuk menampilkan data detil retur penjualan.

20. Fungsi *Transaksi Order* (**SKPL-SITB-020**) .

Merupakan fungsi yang digunakan oleh semua user untuk mencatat transaksi order yaitu transaksi penjualan jasa. Fungsi transaksi order meliputi :

a. Fungsi *Generate Nomor Faktur Order* (**SKPL-SITB-020-01**) .

Merupakan fungsi yang digunakan untuk generate nomor faktur order. Nomor faktur order mempunyai format FJJXXXXXX/MM/YY dimana XXXXXX adalah 6 digit nomor urut transaksi, MM adalah 2 digit angka bulan terjadi transaksi dan YY adalah 2 digit angka tahun terjadi transaksi.

b. Fungsi *Menyimpan Data Transaksi Order ke Tabel Pemesanan* (**SKPL-SITB-020-02**) .

Merupakan fungsi untuk menyimpan data order ke dalam tabel pemesanan.

c. Fungsi *Insert Data Detil Order* (**SKPL-SITB-020-03**) .

Merupakan fungsi yang digunakan untuk menambahkan data detil order jasa.

Program Studi Teknik Informatika	SKPL – SITB	23/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

d. Fungsi *Hapus Data Detil Order* **(SKPL-SITB-020-04)** .

Merupakan fungsi yang digunakan untuk menghapus data detil order.

e. Fungsi *Display Data Detil Order* **(SKPL-SITB-020-05)** .

Merupakan fungsi yang digunakan untuk menampilkan data detil order.

21. Fungsi *Transaksi Pemrosesan Order* **(SKPL-SITB-021)** .

Merupakan fungsi yang dapat digunakan semua user untuk melakukan update status order ketika suatu order mengalami progres dalam pengerjaannya. Status order hanya ada 3 jenis yaitu MULAI, PROSES dan SELESAI.

22. Fungsi *Transaksi Pembatalan Order* **(SKPL-SITB-022)** .

Merupakan fungsi yang dapat digunakan semua user untuk membatalkan suatu order. Suatu transaksi order yang dapat dibatalkan adalah yang mempunyai status order MULAI.

23. Fungsi *Transaksi Pembayaran Piutang* **(SKPL-SITB-023)** .

Merupakan fungsi yang dapat digunakan semua user untuk mencatat data pembayaran piutang atas suatu transaksi penjualan yang dilakukan oleh customer. Pembayaran atas suatu piutang dapat dilakukan dalam beberapa kali pembayaran.

Program Studi Teknik Informatika	SKPL – SITB	24/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

24. Fungsi *Transaksi Mutasi Gudang* (**SKPL-SITB-024**) .

Merupakan fungsi yang digunakan oleh owner untuk mencatat transaksi pemindahan barang dari satu lokasi ke lokasi yang lain.

25. Fungsi *Reporting* (**SKPL-SITB-025**) .

Merupakan fungsi yang digunakan oleh owner untuk melihat laporan dari semua transaksi yang terjadi pada toko buku XYZ.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SITB adalah sebagai berikut :

1. Memahami pengoperasian komputer.
2. Memahami pengoperasian Microsoft Windows.
3. Memahami penggunaan aplikasi SITB.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SITB tersebut adalah :

1. Kebijakan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SITB.

2. Keterbatasan perangkat keras

Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

Program Studi Teknik Informatika	SKPL – SITB	25/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada perangkat desktop yang menggunakan sistem operasi Windows XP minimal SP2, Windows Vista, Windows 7, atau Windows 8.

3 Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak SITB meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form-form.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SITB adalah:

Komponen	Minimal	Dianjurkan
Processor	400 MHz	1 GHz
RAM	96 MB	256 MB
Hard Disk	280 MB	1 GB
Printer	Printer hitam putih yang sesuai dengan OS	Printer berwarna yang sesuai dengan OS
Barcode Scanner	Barcode scanner yang dapat membaca barcode 128 dan sesuai dengan OS	Barcode scanner yang dapat membaca barcode 128 dan sesuai dengan OS

3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SITB adalah :

1. Perangkat Client

Perangkat lunak yang dibutuhkan pada sisi client adalah :

1.1. Nama : .NET Framework 3.5

Sumber : Microsoft.

Sebagai aplikasi framework untuk menjalankan aplikasi SITB dalam perangkat desktop client.

1.2. Nama : Windows XP/ Vista/ 7/ 8

Sumber : Microsoft

Sebagai sistem operasi untuk perangkat client.

1.3. Nama : Crystal Report

Sumber : -.

Sebagai aplikasi untuk mencetak report/resi.

2. Perangkat Server

Perangkat lunak yang dibutuhkan pada sisi server adalah :

2.1. Nama : Microsoft SQL Server 2005

Sumber : Microsoft

Sebagai database management system (DBMS) yang digunakan untuk penyimpanan data di sisi server.

2.2. Nama : Windows Server 2003

Sumber : Microsoft.

Sebagai sistem operasi untuk web server.

2.3. Nama : IIS

Sumber : Microsoft.

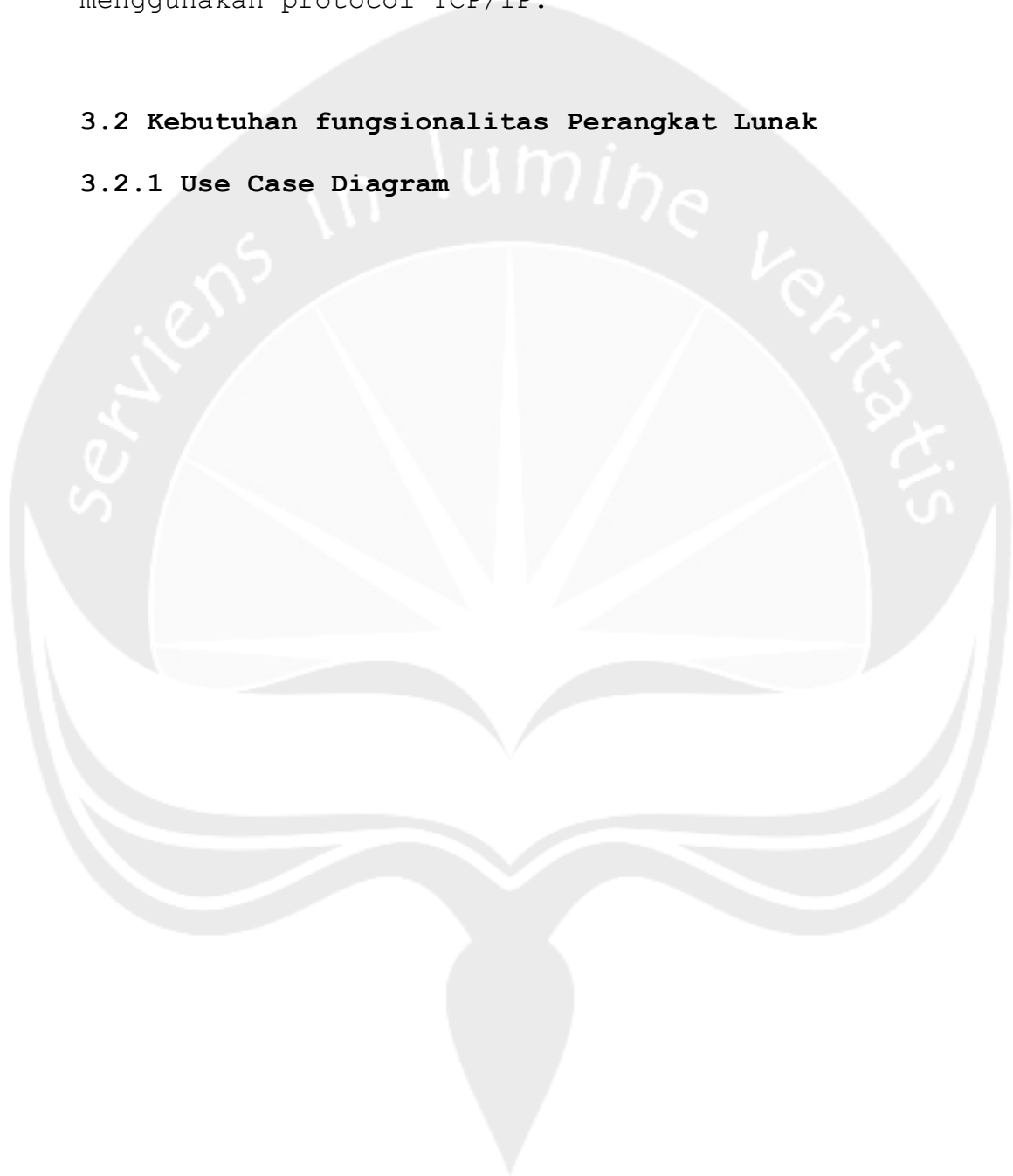
Sebagai web server.

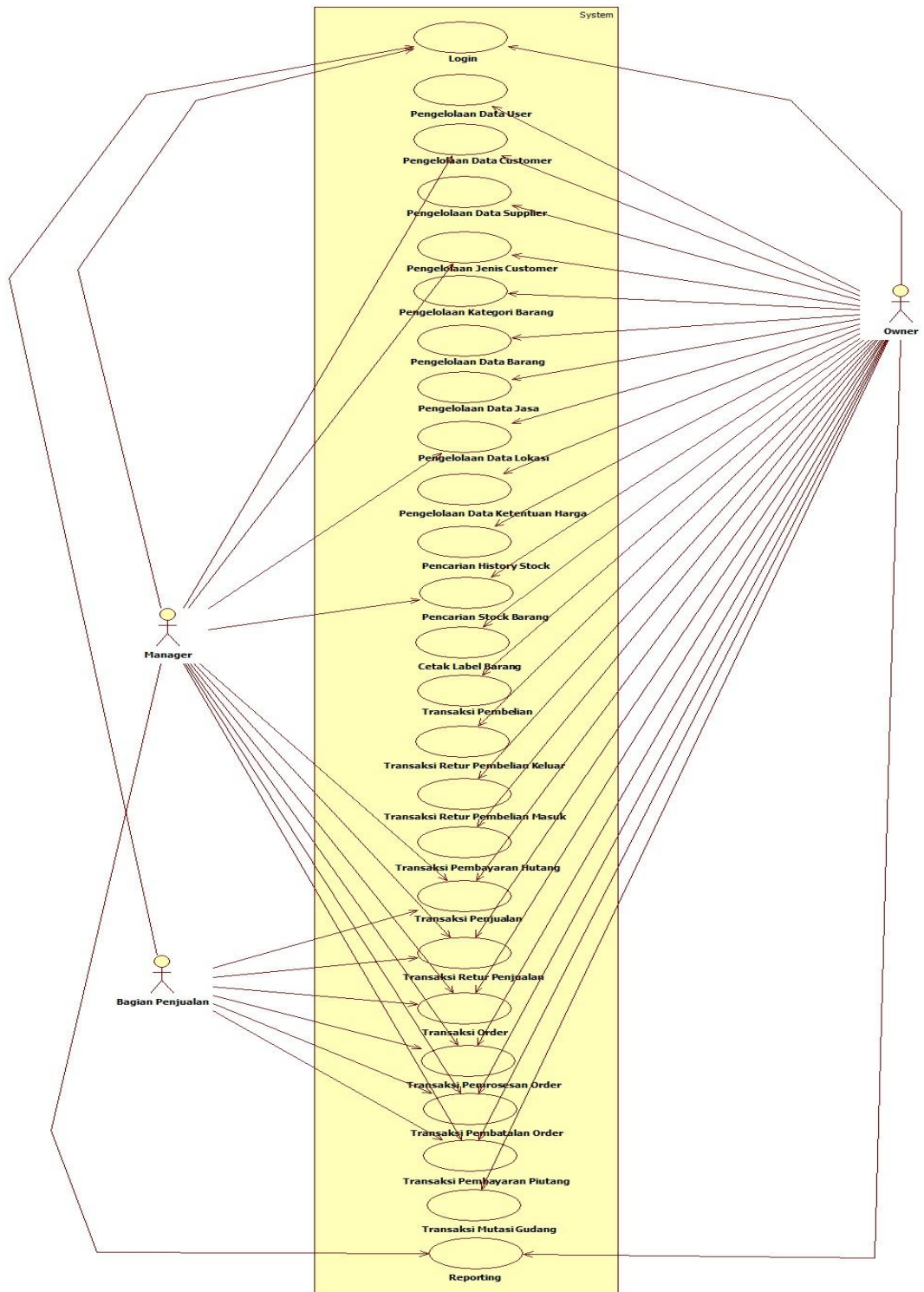
3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SITB menggunakan protocol TCP/IP.

3.2 Kebutuhan fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram





Gambar 2 Use Case Diagram SITB

4 Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

APLIKASI DEKSTOP

4.1.1 Use case Spesification : *Login*

1. Brief Description

Use Case ini digunakan oleh aktor untuk dapat masuk ke dalam sistem.

2. Primary Actor

1. Owner
2. Manager
3. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor hendak masuk ke sistem.
2. Sistem menampilkan form login.
3. Aktor menginputkan data nama pengguna dan kata sandi.
4. Sistem melakukan pemeriksaan data login yang diinputkan yaitu nama pengguna dan kata sandi dan mencocokkan dengan data nama pengguna dan kata sandi yang telah tersimpan di dalam database.
5. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Data nama pengguna dan kata sandi yang diinputkan tidak sesuai.

Program Studi Teknik Informatika	SKPL – SITB	30/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan tidak sesuai.

2. Kembali ke Basic Flow Langkah ke 3.

7. PreConditions

1. Aktor masih berada di luar sistem.

8. PostConditions

1. Aktor telah masuk ke dalam sistem dan dapat menggunakan fungsi yang ada pada sistem.

4.1.2 Use case Spesification : Pengelolaan Data User

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data user atau pengguna yang diijinkan untuk menggunakan sistem. Aktor dapat melakukan tambah data user, ubah data user, atau hapus data user.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data user.

2. Sistem akan menampilkan form pengelolaan data user.

3. Sistem memberikan pilihan untuk melakukan tambah data user, ubah data user, atau hapus data user.

4. Aktor klik tombol tambah untuk melakukan tambah data user.

A-1 Aktor memilih untuk mengubah data user

A-2 Aktor memilih untuk menghapus data user

Program Studi Teknik Informatika	SKPL – SITB	31/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

5. Sistem membangkitkan nomor id pengguna secara otomatis.
6. Aktor mengisi bidang isian.
7. Aktor klik tombol simpan untuk menyimpan data user yang telah diinputkan.
8. Sistem mengecek data user yang telah diinputkan.
E-1 Data user yang diinputkan aktor salah.
9. Sistem menyimpan data user ke database.
10. Sistem menampilkan pemberitahuan bahwa data user berhasil disimpan.
11. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk mengubah data user.
1. Sistem menampilkan data user pada tabel.
 2. Aktor memilih data user yang ingin diubah dengan klik pada salah satu baris data pada tabel.
 3. Data user yang dipilih aktor akan muncul pada bidang isian.
 4. Aktor klik tombol ubah.
 5. Aktor mengubah data user yang ditampilkan pada bidang isian.
 6. Aktor klik tombol simpan untuk menyimpan data user yang telah diubah.
 7. Sistem melakukan pengecekan terhadap data user yang telah diubah.
E-2 Data user yang diubah aktor salah.
 8. Berlanjut ke Basic Flow langkah ke 9.
- A-2 Aktor memilih untuk menghapus data user.
1. Sistem menampilkan data user pada tabel.

Program Studi Teknik Informatika	SKPL – SITB	32/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Aktor memilih data user yang ingin diubah dengan klik pada salah satu baris data pada tabel.
 3. Data user yang dipilih aktor akan muncul pada bidang isian.
 4. Aktor klik tombol hapus.
 5. Sistem menampilkan pesan pertanyaan data user yakin akan dihapus.
 6. Aktor klik tombol yes.
 7. Sistem menghapus data user yang dipilih aktor.
- E-3 Data user tidak bisa dihapus aktor.
8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

E-1 Data user yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke Basic Flow langkah ke 6.

E-2 Data user yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke Alternative Flow A-1 langkah 2.

E-3 Data user tidak bisa dihapus aktor.

1. Sistem memberikan pesan peringatan bahwa data tidak bisa dihapus.
2. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data user di database telah dilakukan penambahan.

Program Studi Teknik Informatika	SKPL – SITB	33/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4.1.3 Use case Spesification : Pengelolaan Data Customer

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data customer. Aktor dapat melakukan tambah data customer, ubah data customer, atau hapus data customer.

2. Primary Actor

1. Owner
2. Manager

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data customer.
2. Sistem akan menampilkan form pengelolaan data customer.
3. Sistem memberikan pilihan untuk melakukan tambah data customer, ubah data customer, atau hapus data customer.
4. Aktor klik tombol tambah untuk melakukan tambah data customer.
 - A-1 Aktor memilih untuk mengubah data customer
 - A-2 Aktor memilih untuk menghapus data customer
5. Aktor mengisi bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data customer yang telah diinputkan.
7. Sistem mengecek data customer yang telah diinputkan.

Program Studi Teknik Informatika	SKPL – SITB	34/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- E-1 Data customer yang diinputkan aktor salah.
8. Sistem menyimpan data customer ke database.
 9. Sistem menampilkan pemberitahuan bahwa data customer berhasil disimpan.
 10. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk mengubah data customer.

1. Sistem menampilkan data customer pada tabel.
2. Aktor memilih data customer yang ingin diubah dengan klik pada salah satu baris data pada tabel.
3. Data customer yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol ubah.
5. Aktor mengubah data customer yang ditampilkan pada bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data customer yang telah diubah.
7. Sistem melakukan pengecekan terhadap data customer yang telah diubah.

E-2 Data customer yang diubah aktor salah.

8. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk menghapus data customer.

1. Sistem menampilkan data customer pada tabel.
2. Aktor memilih data customer yang ingin diubah dengan klik pada salah satu baris data pada tabel.

3. Data customer yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol hapus.
5. Sistem menampilkan pesan pertanyaan data customer yakin akan dihapus.
6. Aktor klik tombol yes.
7. Sistem menghapus data customer yang dipilih aktor.

E-3 Data customer tidak bisa dihapus aktor.

8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

E-1 Data customer yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke Basic Flow Langkah ke 5.

E-2 Data customer yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke Alternative Flow A-1 langkah 2.

E-3 Data customer tidak bisa dihapus aktor.

1. Sistem memberikan pesan peringatan bahwa data tidak bisa dihapus.
2. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data customer di database telah dilakukan penambahan.

Program Studi Teknik Informatika	SKPL – SITB	36/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4.1.4 Use case Spesification : Pengelolaan Data Supplier

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data supplier. Aktor dapat melakukan tambah data supplier, ubah data supplier, atau hapus data supplier.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data supplier.
2. Sistem akan menampilkan form pengelolaan data supplier.
3. Sistem memberikan pilihan untuk melakukan tambah data supplier, ubah data supplier, atau hapus data supplier.
4. Aktor klik tombol tambah untuk melakukan tambah data supplier.
A-1 Aktor memilih untuk mengubah data supplier
A-2 Aktor memilih untuk menghapus data supplier
5. Aktor mengisi bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data supplier yang telah diinputkan.
7. Sistem mengecek data supplier yang telah diinputkan.

- E-1 Data supplier yang diinputkan aktor salah.
8. Sistem menyimpan data supplier ke database.
 9. Sistem menampilkan pemberitahuan bahwa data supplier berhasil disimpan.
 10. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk mengubah data supplier.

1. Sistem menampilkan data supplier pada tabel.
2. Aktor memilih data supplier yang ingin diubah dengan klik pada salah satu baris data pada tabel.
3. Data supplier yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol ubah.
5. Aktor mengubah data supplier yang ditampilkan pada bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data supplier yang telah diubah.
7. Sistem melakukan pengecekan terhadap data supplier yang telah diubah.

E-2 Data supplier yang diubah aktor salah.

8. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk menghapus data supplier.

1. Sistem menampilkan data supplier pada tabel.
2. Aktor memilih data supplier yang ingin diubah dengan klik pada salah satu baris data pada tabel.

3. Data supplier yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol hapus.
5. Sistem menampilkan pesan pertanyaan data supplier yakin akan dihapus.
6. Aktor klik tombol yes.
7. Sistem menghapus data supplier yang dipilih aktor.

E-3 Data supplier tidak bisa dihapus aktor.

8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

E-1 Data supplier yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data supplier yang diinputkan salah.
2. Kembali ke Basic Flow Langkah ke 5.

E-2 Data supplier yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data supplier yang diubah salah.
2. Kembali ke Alternative Flow A-1 langkah 2.

E-3 Data supplier tidak bisa dihapus aktor.

1. Sistem memberikan pesan peringatan bahwa data supplier tidak bisa dihapus.
2. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data supplier di database telah dilakukan penambahan.

Program Studi Teknik Informatika	SKPL – SITB	39/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4.1.5 Use case Spesification : Pengelolaan Jenis Customer

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data jenis customer. Aktor dapat melakukan tambah data jenis customer, ubah data jenis customer, atau hapus data jenis customer.

2. Primary Actor

1. Owner
2. Manager

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data jenis customer.
2. Sistem akan menampilkan form pengelolaan data jenis customer.
3. Sistem memberikan pilihan untuk melakukan tambah data jenis customer, ubah data jenis customer, atau hapus data jenis customer.
4. Aktor klik tombol tambah untuk melakukan tambah data jenis customer.
 - A-1 Aktor memilih untuk mengubah data jenis customer
 - A-2 Aktor memilih untuk menghapus data jenis customer
5. Sistem membangkitkan nomor id jenis customer.
6. Aktor mengisi bidang isian.
7. Aktor klik tombol simpan untuk menyimpan data jenis customer yang telah diinputkan.

8. Sistem mengecek data jenis customer yang telah diinputkan.

E-1 Data jenis customer yang diinputkan aktor salah.

9. Sistem menyimpan data jenis customer ke database.

10. Sistem menampilkan pemberitahuan bahwa data jenis customer berhasil disimpan.

11. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk mengubah data jenis customer.

1. Sistem menampilkan data jenis customer pada tabel.

2. Aktor memilih data jenis customer yang ingin diubah dengan klik pada salah satu baris data pada tabel.

3. Data jenis customer yang dipilih aktor akan muncul pada bidang isian.

4. Aktor klik tombol ubah.

5. Aktor mengubah data jenis customer yang ditampilkan pada bidang isian.

6. Aktor klik tombol simpan untuk menyimpan data jenis customer yang telah diubah.

7. Sistem melakukan pengecekan terhadap data jenis customer yang telah diubah.

E-2 Data jenis customer yang diubah aktor salah.

8. Berlanjut ke Basic Flow langkah ke 9.

A-2 Aktor memilih untuk menghapus data jenis customer.

1. Sistem menampilkan data jenis customer pada tabel.
 2. Aktor memilih data jenis customer yang ingin diubah dengan klik pada salah satu baris data pada tabel.
 3. Data jenis customer yang dipilih aktor akan muncul pada bidang isian.
 4. Aktor klik tombol hapus.
 5. Sistem menampilkan pesan pertanyaan data jenis customer yakin akan dihapus.
 6. Aktor klik tombol yes.
 7. Sistem menghapus data jenis customer yang dipilih aktor.
- E-3 Data jenis customer tidak bisa dihapus aktor.
8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

- E-1 Data jenis customer yang diinputkan aktor salah.
1. Sistem memberikan pesan peringatan bahwa data jenis customer yang diinputkan salah.
 2. Kembali ke Basic Flow langkah ke 6.
- E-2 Data jenis customer yang diubah aktor salah.
1. Sistem memberikan pesan peringatan bahwa data jenis customer yang diubah salah.
 2. Kembali ke Alternative Flow A-1 langkah 2.
- E-3 Data jenis customer tidak bisa dihapus aktor.
1. Sistem memberikan pesan peringatan bahwa data jenis customer tidak bisa dihapus.
 2. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data jenis customer di database telah dilakukan penambahan.

4.1.6 Use case Spesification : Pengelolaan Kategori Barang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data kategori barang. Aktor dapat melakukan tambah data jenis kategori barang, ubah data jenis kategori barang, atau hapus data jenis kategori barang.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data kategori barang.
2. Sistem akan menampilkan form pengelolaan data kategori barang.
3. Sistem memberikan pilihan untuk melakukan tambah data kategori barang, ubah data kategori barang, atau hapus data kategori barang.
4. Aktor klik tombol tambah untuk melakukan tambah data kategori barang.
A-1 Aktor memilih untuk mengubah data kategori barang

Program Studi Teknik Informatika	SKPL – SITB	43/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- A-2 Aktor memilih untuk menghapus data kategori barang
5. Sistem membangkitkan nomor id kategori barang secara otomatis.
 6. Aktor mengisi bidang isian.
 7. Aktor klik tombol simpan untuk menyimpan data kategori barang yang telah diinputkan.
 8. Sistem mengecek data kategori barang yang telah diinputkan.

- E-1 Data kategori barang yang diinputkan aktor salah.
9. Sistem menyimpan data kategori barang ke database.
 10. Sistem menampilkan pemberitahuan bahwa data kategori barang berhasil disimpan.
 11. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk mengubah data kategori barang.
1. Sistem menampilkan data kategori barang pada tabel.
 2. Aktor memilih data kategori barang yang ingin diubah dengan klik pada salah satu baris data pada tabel.
 3. Data kategori barang yang dipilih aktor akan muncul pada bidang isian.
 4. Aktor klik tombol ubah.
 5. Aktor mengubah data kategori barang yang ditampilkan pada bidang isian.
 6. Aktor klik tombol simpan untuk menyimpan data kategori barang yang telah diubah.

7. Sistem melakukan pengecekan terhadap data kategori barang yang telah diubah.

E-2 Data kategori barang yang diubah aktor salah.

8. Berlanjut ke Basic Flow langkah ke 9.

A-2 Aktor memilih untuk menghapus data kategori barang.

1. Sistem menampilkan data kategori barang pada tabel.

2. Aktor memilih data kategori barang yang ingin diubah dengan klik pada salah satu baris data pada tabel.

3. Data kategori barang yang dipilih aktor akan muncul pada bidang isian.

4. Aktor klik tombol hapus.

5. Sistem menampilkan pesan pertanyaan data kategori barang yakin akan dihapus.

6. Aktor klik tombol yes.

7. Sistem menghapus data kategori barang yang dipilih aktor.

E-3 Data kategori barang tidak bisa dihapus aktor.

8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

E-1 Data kategori barang yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data kategori barang yang diinputkan salah.

2. Kembali ke Basic Flow langkah ke 6.

E-2 Data kategori barang yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data kategori barang yang diubah salah.
 2. Kembali ke Alternative Flow A-1 langkah 2.
- E-3 Data kategori barang tidak bisa dihapus aktor.
1. Sistem memberikan pesan peringatan bahwa data kategori barang tidak bisa dihapus.
 2. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

2. Data kategori barang di database telah dilakukan penambahan.

4.1.7 Use case Spesification : Pengelolaan Data Barang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data barang. Aktor dapat melakukan tambah data barang, ubah data barang, atau hapus data barang.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data barang.
2. Sistem akan menampilkan form pengelolaan data barang.

Program Studi Teknik Informatika	SKPL – SITB	46/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Sistem memberikan pilihan untuk melakukan tambah data barang, ubah data barang, atau hapus data barang.
4. Aktor klik tombol tambah untuk melakukan tambah data barang.
A-1 Aktor memilih untuk mengubah data barang
A-2 Aktor memilih untuk menghapus data barang
5. Aktor mengisi bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data barang yang telah diinputkan.
7. Sistem mengecek data barang yang telah diinputkan.
E-1 Data barang yang diinputkan aktor salah.
8. Sistem menyimpan data barang ke database.
9. Sistem menampilkan pemberitahuan bahwa data barang berhasil disimpan.
10. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk mengubah data barang.
1. Sistem menampilkan data barang pada tabel.
 2. Aktor memilih data barang yang ingin diubah dengan klik pada salah satu baris data pada tabel.
 3. Data barang yang dipilih aktor akan muncul pada bidang isian.
 4. Aktor klik tombol ubah.
 5. Aktor mengubah data barang yang ditampilkan pada bidang isian.
 6. Aktor klik tombol simpan untuk menyimpan data barang yang telah diubah.

7. Sistem melakukan pengecekan terhadap data barang yang telah diubah.

E-2 Data barang yang diubah aktor salah.

8. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk menghapus data barang.

1. Sistem menampilkan data barang pada tabel.

2. Aktor memilih data barang yang ingin diubah dengan klik pada salah satu baris data pada tabel.

3. Data barang yang dipilih aktor akan muncul pada bidang isian.

4. Aktor klik tombol hapus.

5. Sistem menampilkan pesan pertanyaan data barang yakin akan dihapus.

6. Aktor klik tombol yes.

7. Sistem menghapus data barang yang dipilih aktor.

E-3 Data barang tidak bisa dihapus aktor.

8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

E-1 Data barang yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data barang yang diinputkan salah.

2. Kembali ke Basic Flow Langkah ke 5.

E-2 Data barang yang diubah aktor salah.

3. Sistem memberikan pesan peringatan bahwa data barang yang diubah salah.

4. Kembali ke Alternative Flow A-1 langkah 2.

E-3 Data barang tidak bisa dihapus aktor.

3. Sistem memberikan pesan peringatan bahwa data barang tidak bisa dihapus.

4. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data barang di database telah dilakukan penambahan.

4.1.8 Use case Spesification : Pengelolaan Data Jasa

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data jasa. Aktor dapat melakukan tambah data jasa, ubah data jasa, atau hapus data jasa.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data jasa.
2. Sistem akan menampilkan form pengelolaan data jasa.
3. Sistem memberikan pilihan untuk melakukan tambah data jasa, ubah data jasa, atau hapus data jasa.
4. Aktor klik tombol tambah untuk melakukan tambah data jasa.
A-1 Aktor memilih untuk mengubah data jasa
A-2 Aktor memilih untuk menghapus data jasa
5. Aktor mengisi bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data jasa yang telah diinputkan.

Program Studi Teknik Informatika	SKPL – SITB	49/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

7. Sistem mengecek data jasa yang telah diinputkan.

E-1 Data jasa yang diinputkan aktor salah.

8. Sistem menyimpan data jasa ke database.

9. Sistem menampilkan pemberitahuan bahwa data jasa berhasil disimpan.

10. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk mengubah data jasa.

1. Sistem menampilkan data jasa pada tabel.
2. Aktor memilih data jasa yang ingin diubah dengan klik pada salah satu baris data pada tabel.
3. Data jasa yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol ubah.
5. Aktor mengubah data jasa yang ditampilkan pada bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data jasa yang telah diubah.
7. Sistem melakukan pengecekan terhadap data jasa yang telah diubah.

E-2 Data jasa yang diubah aktor salah.

8. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk menghapus data jasa.

1. Sistem menampilkan data jasa pada tabel.
2. Aktor memilih data jasa yang ingin diubah dengan klik pada salah satu baris data pada tabel.
3. Data jasa yang dipilih aktor akan muncul pada bidang isian.

Program Studi Teknik Informatika	SKPL – SITB	50/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Aktor klik tombol hapus.
5. Sistem menampilkan pesan pertanyaan data jasa yakin akan dihapus.
6. Aktor klik tombol yes.
7. Sistem menghapus data jasa yang dipilih aktor.
E-3 Data jasa tidak bisa dihapus aktor.
8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

- E-1 Data jasa yang diinputkan aktor salah.
1. Sistem memberikan pesan peringatan bahwa data jasa yang diinputkan salah.
 2. Kembali ke Basic Flow Langkah ke 5.
- E-2 Data jasa yang diubah aktor salah.
1. Sistem memberikan pesan peringatan bahwa data jasa yang diubah salah.
 2. Kembali ke Alternative Flow A-1 langkah 2.
- E-3 Data jasa tidak bisa dihapus aktor.
1. Sistem memberikan pesan peringatan bahwa data jasa tidak bisa dihapus.
 2. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data jasa di database telah dilakukan penambahan.

4.1.9 Use case Spesification : Pengelolaan Data Lokasi

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data lokasi. Aktor dapat melakukan tambah data lokasi, ubah data lokasi, atau hapus data lokasi.

2. Primary Actor

1. Owner
2. Manager

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data lokasi.
2. Sistem akan menampilkan form pengelolaan data lokasi.
3. Sistem memberikan pilihan untuk melakukan tambah data lokasi, ubah data lokasi, atau hapus data lokasi.
4. Aktor klik tombol tambah untuk melakukan tambah data lokasi.
A-1 Aktor memilih untuk mengubah data lokasi
A-2 Aktor memilih untuk menghapus data lokasi
5. Aktor mengisi bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data lokasi yang telah diinputkan.
7. Sistem mengecek data lokasi yang telah diinputkan.
E-1 Data lokasi yang diinputkan aktor salah.
8. Sistem menyimpan data lokasi ke database.

9. Sistem menampilkan pemberitahuan bahwa data lokasi berhasil disimpan.

10. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk mengubah data lokasi.

1. Sistem menampilkan data lokasi pada tabel.
2. Aktor memilih data lokasi yang ingin diubah dengan klik pada salah satu baris data pada tabel.
3. Data lokasi yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol ubah.
5. Aktor mengubah data lokasi yang ditampilkan pada bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data lokasi yang telah diubah.
7. Sistem melakukan pengecekan terhadap data lokasi yang telah diubah.

E-2 Data lokasi yang diubah aktor salah.

8. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk menghapus data lokasi.

1. Sistem menampilkan data lokasi pada tabel.
2. Aktor memilih data lokasi yang ingin diubah dengan klik pada salah satu baris data pada tabel.
3. Data lokasi yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol hapus.
5. Sistem menampilkan pesan pertanyaan data lokasi yakin akan dihapus.
6. Aktor klik tombol yes.

7. Sistem menghapus data lokasi yang dipilih aktor.

E-3 Data lokasi tidak bisa dihapus aktor.

8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

E-1 Data lokasi yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data lokasi yang diinputkan salah.
2. Kembali ke Basic Flow Langkah ke 5.

E-2 Data lokasi yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data lokasi yang diubah salah.
2. Kembali ke Alternative Flow A-1 langkah 2.

E-3 Data lokasi tidak bisa dihapus aktor.

1. Sistem memberikan pesan peringatan bahwa data lokasi tidak bisa dihapus.
2. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data lokasi di database telah dilakukan penambahan.

4.1.10 Use case Spesification : Pengelolaan Ketentuan Harga

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data ketentuan harga. Aktor dapat melakukan tambah data ketentuan harga, ubah data ketentuan harga, atau hapus data ketentuan harga.

2. Primary Actor

Program Studi Teknik Informatika	SKPL – SITB	54/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pengelolaan data ketentuan harga.
2. Sistem akan menampilkan form pengelolaan data ketentuan harga.
3. Sistem memberikan pilihan untuk melakukan tambah data ketentuan harga, ubah data ketentuan harga, atau hapus data ketentuan harga.
4. Aktor klik tombol tambah untuk melakukan tambah data ketentuan harga.
 - A-1 Aktor memilih untuk mengubah data ketentuan harga
 - A-2 Aktor memilih untuk menghapus data ketentuan harga
5. Sistem membangkitkan nomor id ketentuan harga secara otomatis.
6. Aktor mengisi bidang isian.
7. Aktor klik tombol simpan untuk menyimpan data ketentuan harga yang telah diinputkan.
8. Sistem mengecek data ketentuan harga yang telah diinputkan.
 - E-1 Data ketentuan harga yang diinputkan aktor salah.
9. Sistem menyimpan data ketentuan harga ke database.
10. Sistem menampilkan pemberitahuan bahwa data ketentuan harga berhasil disimpan.

Program Studi Teknik Informatika	SKPL – SITB	55/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

11. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk mengubah data ketentuan harga.

1. Sistem menampilkan data ketentuan harga pada tabel.
2. Aktor memilih data ketentuan harga yang ingin diubah dengan klik pada salah satu baris data pada tabel.
3. Data ketentuan harga yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol ubah.
5. Aktor mengubah data ketentuan harga yang ditampilkan pada bidang isian.
6. Aktor klik tombol simpan untuk menyimpan data ketentuan harga yang telah diubah.
7. Sistem melakukan pengecekan terhadap data ketentuan harga yang telah diubah.

E-2 Data ketentuan harga yang diubah aktor salah.

8. Berlanjut ke Basic Flow langkah ke 9.

A-2 Aktor memilih untuk menghapus data ketentuan harga.

1. Sistem menampilkan data ketentuan harga pada tabel.
2. Aktor memilih data ketentuan harga yang ingin diubah dengan klik pada salah satu baris data pada tabel.
3. Data ketentuan harga yang dipilih aktor akan muncul pada bidang isian.
4. Aktor klik tombol hapus.

5. Sistem menampilkan pesan pertanyaan data ketentuan harga yakin akan dihapus.
6. Aktor klik tombol yes.
7. Sistem menghapus data ketentuan harga yang dipilih aktor.
E-3 Data ketentuan harga tidak bisa dihapus aktor.
8. Berlanjut ke Basic Flow langkah ke 9.

6. Error Flow

E-1 Data ketentuan harga yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data ketentuan harga yang diinputkan salah.
2. Kembali ke Basic Flow langkah ke 6.

E-2 Data ketentuan harga yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data ketentuan harga yang diubah salah.
2. Kembali ke Alternative Flow A-1 langkah 2.

E-3 Data ketentuan harga tidak bisa dihapus aktor.

1. Sistem memberikan pesan peringatan bahwa data ketentuan harga tidak bisa dihapus.
2. Kembali ke Alternative Flow A-2 langkah 2.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data jenis customer di database telah dilakukan penambahan.

4.1.11 Use case Spesification : Pencarian History Stock

1. Brief Description

Use Case ini digunakan oleh aktor untuk melihat data perubahan stock barang yang terjadi. Aktor dapat melakukan pilihan atas pencarian history stock barang berdasarkan kode barang dan periode waktu.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pencarian history stock.
2. Sistem akan menampilkan form pencarian history stock.
3. Aktor mengisi bidang isian.
4. Aktor klik tombol cari.
5. Sistem menampilkan hasil pencarian data history stock dari database yang ditampilkan pada tabel.
6. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data history stock ditampilkan.

4.1.12 Use case Spesification : Pencarian Stock Barang

1. Brief Description

Use Case ini digunakan oleh aktor untuk melihat data stock barang yang ada pada setiap lokasi yang ada.

2. Primary Actor

1. Owner
2. Manager
3. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pencarian stock barang.
2. Sistem akan menampilkan form pencarian stock barang.
3. Aktor mengisi bidang isian barang yang ingin dicari stocknya.
A-1 Aktor memilih untuk mencetak data stock.
4. Aktor klik tombol cari.
5. Sistem menampilkan hasil pencarian data stock dari database yang ditampilkan pada tabel.
6. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk mencetak data stock.
1. Aktor klik tombol cetak.
 2. Sistem akan menampilkan report data stock yang siap untuk dicetak.
 3. Kembali ke Basic Flow langkah ke 6.

6. Error Flow

none

Program Studi Teknik Informatika	SKPL – SITB	59/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data stock ditampilkan.

4.1.13 Use case Spesification : Cetak Label Barang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mencetak label barang.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu cetak label barang.
2. Sistem akan menampilkan form pengelolaan label.
3. Aktor memilih data barang yang ingin dicetak label dengan klik pada salah satu baris data pada tabel.
4. Aktor menginputkan jumlah label dan posisi mulai cetak label.
5. Aktor klik tombol cetak.
6. Sistem mengecek inputan aktor.
E-1 Inputan jumlah label dan posisi mulai cetak salah.
7. Sistem menampilkan report label barang yang siap untuk dicetak.
8. Use Case selesai.

Program Studi Teknik Informatika	SKPL – SITB	60/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

5. Alternative Flow

none

6. Error Flow

E-1 Inputan jumlah label dan posisi mulai cetak salah.

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 4.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Label stock siap dicetak

4.1.14 Use case Spesification : Transaksi Pembelian

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi pembelian barang. Aktor dapat membuat transaksi pembelian barang baru dari satu supplier dan aktor dapat menambahkan barang yang dibeli atau menghapus barang yang dibeli selama transaksi pembelian belum tersimpan.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi pembelian.
2. Sistem akan menampilkan form transaksi pembelian.
3. Aktor menginputkan data pembelian.

Program Studi Teknik Informatika	SKPL – SITB	61/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Aktor klik tombol transaksi baru untuk membuat satu transaksi pembelian baru.
5. Aktor menginputkan data detail pembelian dari barang yang dibeli pada bidang isian.
6. Aktor klik tombol tambah untuk menambahkan barang yang dibeli.
A-1 Aktor memilih untuk menghapus barang.
7. Sistem mengecek inputan data detail pembelian dari aktor.
E-1 Data detail pembelian barang yang diinput salah.
8. Sistem menambahkan data detail pembelian barang ke tabel.
9. Aktor klik tombol simpan transaksi untuk menyimpan data pembelian ke dalam database.
10. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk menghapus barang.
1. Aktor memilih data detail pembelian barang yang ingin dihapus dengan klik pada salah satu baris data pada tabel.
 2. Aktor klik tombol hapus.
 3. Kembali ke Basic Flow langkah ke 5.

6. Error Flow

- E-1 Data detail pembelian barang yang diinput salah.
1. Sistem menampilkan pesan kesalahan.
 2. Kembali ke Basic Flow langkah ke 5.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

Program Studi Teknik Informatika	SKPL – SITB	62/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Transaksi pembelian baru telah disimpan ke database

4.1.15 Use case Spesification : Transaksi Retur Pembelian Keluar

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi retur pembelian keluar.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi retur pembelian keluar.
2. Sistem akan menampilkan form transaksi retur pembelian keluar.
3. Aktor klik tombol retur baru untuk membuat satu transaksi retur pembelian keluar baru.
4. Aktor menginputkan data barang yang diretur dan jumlah.
5. Sistem mengecek data jumlah barang yang diinputkan dengan data jumlah barang pada database.
E-1 Data jumlah barang yang diinput salah.
6. Aktor klik tombol tambah untuk menambahkan barang yang diretur.
A-1 Aktor memilih untuk menghapus barang.
7. Sistem mengecek inputan data detil barang yang diretur.
E-2 Data detil barang yang diinput salah.

Program Studi Teknik Informatika	SKPL – SITB	63/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

8. Sistem menambahkan data detail barang yang diretur keluar ke tabel.
9. Aktor klik tombol simpan transaksi untuk menyimpan data retur pembelian keluar ke dalam database.
10. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk menghapus barang.

1. Aktor memilih data detail pembelian barang yang ingin dihapus dengan klik pada salah satu baris data pada tabel.
2. Aktor klik tombol hapus.
3. Kembali ke Basic Flow langkah ke 4.

6. Error Flow

E-1 Data jumlah barang yang diinput salah.

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 4.

E-2 Data detail barang yang diinput salah.

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 4.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Transaksi retur pembelian keluar baru telah disimpan ke database

4.1.16 Use case Spesification : Transaksi Retur Pembelian Masuk

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi retur pembelian masuk. Barang

Program Studi Teknik Informatika	SKPL – SITB	64/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

yang dikenakan transaksi retur pembelian masuk bisa berasal dari lebih dari satu supplier.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi retur pembelian masuk.
2. Sistem akan menampilkan form transaksi retur pembelian masuk.
3. Aktor klik tombol retur baru untuk membuat satu transaksi retur pembelian keluar baru.
4. Aktor menginputkan data supplier dan barang yang diretur dan jumlah.
5. Sistem mengecek data jumlah barang yang diinputkan dengan data jumlah barang yang pernah diretur keluar.
E-1 Data jumlah barang yang diinput salah.
6. Aktor klik tombol tambah untuk menambahkan barang yang diretur.
A-1 Aktor memilih untuk menghapus barang.
7. Sistem mengecek inputan data detail barang yang diretur.
E-2 Data detail barang yang diinput salah.
8. Sistem menambahkan data detail barang yang diretur masuk ke tabel.
9. Aktor klik tombol simpan transaksi untuk menyimpan data retur pembelian masuk ke dalam database.
10. Use Case selesai.

Program Studi Teknik Informatika	SKPL – SITB	65/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

5. Alternative Flow

A-1 Aktor memilih untuk menghapus barang.

1. Aktor memilih data detail pembelian barang yang ingin dihapus dengan klik pada salah satu baris data pada tabel.
2. Aktor klik tombol hapus.
3. Kembali ke Basic Flow langkah ke 4.

6. Error Flow

E-1 Data jumlah barang yang diinput salah.

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 4.

E-2 Data detail barang yang diinput salah.

3. Sistem menampilkan pesan kesalahan.
4. Kembali ke Basic Flow langkah ke 4.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Transaksi retur pembelian masuk baru telah disimpan ke database

4.1.17 Use case Spesification : Transaksi Pembayaran Hutang

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi pembayaran hutang.

2. Primary Actor

1. Owner

3. Supporting Actor

none

Program Studi Teknik Informatika	SKPL – SITB	66/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi pembayaran hutang.
2. Sistem akan menampilkan form pembayaran hutang.
3. Aktor menginputkan nomor faktur transaksi pembelian yang masih hutang.
4. Sistem mengecek inputan nomor faktur pembelian yang ada pada database.
E-1 Inputan nomor faktur salah.
5. Sistem menampilkan data hutang pada bidang isian.
6. Aktor menginputkan data jumlah pembayaran dan tanggal pembayaran.
E-2 Inputan jumlah pembayaran salah.
7. Sistem mengecek inputan aktor.
E-3 Inputan aktor salah.
8. Aktor klik tombol simpan untuk menyimpan data pembayaran hutang.
9. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

- E-1 Inputan nomor faktur salah.
1. Sistem menampilkan pesan kesalahan.
 2. Kembali ke Basic Flow langkah ke 4.
- E-2 Inputan jumlah pembayaran salah.
1. Sistem menampilkan pesan kesalahan.
 2. Kembali ke Basic Flow langkah ke 6.
- E-3 Inputan aktor salah.
1. Sistem menampilkan pesan kesalahan.

2. Kembali ke Basic Flow langkah ke 6.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Transaksi pembayaran hutang atas satu nomor faktur tersimpan ke dalam database

4.1.18 Use case Spesification : Transaksi Penjualan

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi penjualan barang.

2. Primary Actor

1. Owner
2. Manager
3. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi penjualan.
2. Sistem akan menampilkan form transaksi penjualan.
3. Aktor menginputkan kode customer.
4. Sistem mengecek inputan kode customer dari aktor.
E-1 Inputan kode customer salah.
5. Sistem membangkitkan nomor faktur penjualan secara otomatis.
6. Aktor menginputkan data detail penjualan barang.

Program Studi Teknik Informatika	SKPL – SITB	68/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

7. Aktor klik tombol tambah untuk menambahkan barang.

A-1 Aktor memilih untuk menghapus barang.

8. Sistem mengecek inputan data detail penjualan barang.

E-2 Data detail penjualan barang yang diinput salah.

9. Sistem menambahkan data detail penjualan barang ke tabel.

10. Aktor klik tombol bayar.

11. Sistem menampilkan form pembayaran.

12. Aktor menginputkan data pembayaran.

13. Aktor klik tombol selesai.

14. Sistem mengecek inputan data pembayaran.

E-3 Data pembayaran yang diinput salah.

15. Aktor klik tombol simpan transaksi untuk menyimpan data transaksi penjualan ke dalam database.

16. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk menghapus barang.

1. Aktor memilih data detail penjualan barang yang ingin dihapus dengan klik pada salah satu baris data pada tabel.

2. Aktor klik tombol hapus.

3. Kembali ke Basic Flow langkah ke 6.

6. Error Flow

E-1 Inputan kode customer salah.

1. Sistem menampilkan pesan kesalahan.

2. Kembali ke Basic Flow langkah ke 3.

E-2 Data detil penjualan barang yang diinput salah.

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 6.

E-3 Data pembayaran yang diinput salah.

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 12.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Transaksi penjualan baru telah disimpan ke database

4.1.19 Use case Spesification : Transaksi Retur Penjualan

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi retur penjualan masuk. Barang yang dikenakan transaksi retur penjualan bergantung pada nomor faktur transaksi penjualan.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi retur penjualan.
2. Sistem akan menampilkan form transaksi retur penjualan.
3. Aktor menginputkan nomor faktur penjualan.

Program Studi Teknik Informatika	SKPL – SITB	70/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Sistem mengecek nomor faktur penjualan yang ada pada database.

E-1 Nomor faktur penjualan yang diinput salah.

5. Aktor menginputkan data detil barang yang diretur.

6. Sistem mengecek inputan data detil barang yang diretur.

E-2 Data detil barang yang diinput salah.

7. Aktor klik tombol tambah untuk menambahkan barang yang diretur.

A-1 Aktor memilih untuk menghapus barang.

8. Sistem menambahkan data detil barang yang diretur penjualan ke tabel.

9. Aktor klik tombol simpan transaksi untuk menyimpan data retur penjualan ke dalam database.

10. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk menghapus barang.

1. Aktor memilih data detil pembelian barang yang ingin dihapus dengan klik pada salah satu baris data pada tabel.

2. Aktor klik tombol hapus.

3. Kembali ke Basic Flow langkah ke 5.

6. Error Flow

E-1 Nomor faktur penjualan yang diinput salah.

1. Sistem menampilkan pesan kesalahan.

2. Kembali ke Basic Flow langkah ke 3.

E-2 Data detil barang yang diinput salah.

1. Sistem menampilkan pesan kesalahan.

2. Kembali ke Basic Flow langkah ke 5.

Program Studi Teknik Informatika	SKPL – SITB	71/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Transaksi retur penjualan baru telah disimpan ke database

4.1.20 Use case Spesification : Transaksi Order

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi order. Ketika sebuah transaksi order dibuat, transaksi order akan mendapat status MULAI.

2. Primary Actor

1. Owner
2. Manager
3. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi order baru.
2. Sistem akan menampilkan form transaksi order.
3. Aktor menginputkan kode customer.
4. Sistem mengecek inputan kode customer dari aktor.
E-1 Inputan kode customer salah.
5. Sistem membangkitkan nomor faktur order secara otomatis.
6. Aktor menginputkan data detil order.
7. Aktor klik tombol tambah untuk menambahkan data detil order yang telah diinputkan.

Program Studi Teknik Informatika	SKPL – SITB	72/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- A-1 Aktor memilih untuk menghapus data detail order.
8. Sistem mengecek inputan data detail order.
- E-2 Data detail order yang diinput salah.
9. Sistem menambahkan data detail order ke tabel.
10. Aktor klik tombol bayar.
11. Sistem menampilkan form pembayaran.
12. Aktor menginputkan data pembayaran.
13. Aktor klik tombol selesai.
14. Sistem mengecek inputan data pembayaran.
- E-3 Data pembayaran yang diinput salah.
15. Aktor klik tombol simpan transaksi untuk menyimpan data transaksi order ke dalam database.
16. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk menghapus data detail order.
1. Aktor memilih data detail order yang ingin dihapus dengan klik pada salah satu baris data pada tabel.
 2. Aktor klik tombol hapus.
 3. Kembali ke Basic Flow langkah ke 6.

6. Error Flow

- E-1 Inputan kode customer salah.
1. Sistem menampilkan pesan kesalahan.
 2. Kembali ke Basic Flow langkah ke 3.
- E-2 Data detail order yang diinput salah.
1. Sistem menampilkan pesan kesalahan.
 2. Kembali ke Basic Flow langkah ke 6.
- E-3 Data pembayaran yang diinput salah.

Program Studi Teknik Informatika	SKPL – SITB	73/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 12.

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Transaksi order baru telah disimpan ke database

4.1.21 Use case Spesification : Transaksi Pemrosesan Order

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengupdate atau memperbarui status dari sebuah order. Status dari sebuah order melalui 3 tahap yaitu MULAI, PROSES, dan SELESAI.

2. Primary Actor

1. Owner
2. Manager
3. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pemrosesan order.
2. Sistem menampilkan form proses order.
3. Aktor memilih data transaksi order atau mencari data berdasarkan nomor faktur order.
4. Data transaksi order akan muncul pada bidang isian, dimana aktor tidak dapat mengubah isi data tersebut.

Program Studi Teknik Informatika	SKPL – SITB	74/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

5. Aktor klik tombol Simpan untuk menyimpan perubahan status order yang terjadi.
6. Sistem akan memberikan notifikasi bahwa data transaksi order telah dilakukan update status.
7. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data status transaksi order sudah diperbaharui

4.1.22 Use case Spesification : Transaksi Pembatalan Order

1. Brief Description

Use Case ini digunakan oleh aktor untuk membatalkan transaksi dari sebuah order. Transaksi order yang dapat dibatalkan hanya yang berstatus MULAI.

2. Primary Actor

1. Owner
2. Manager
3. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu pembatalan order.
2. Sistem menampilkan form pembatalan order.

Program Studi Teknik Informatika	SKPL – SITB	75/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Aktor memilih data transaksi order atau mencari data berdasarkan nomor faktur order.
4. Aktor klik tombol Batal Order untuk membatalkan sebuah transaksi order yang terjadi.
5. Sistem akan memberikan notifikasi bahwa data transaksi order telah dilakukan pembatalan.
6. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data status transaksi order sudah dibatalkan

4.1.23 Use case Spesification : Transaksi Pembayaran Piutang

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi pembayaran piutang.

2. Primary Actor

1. Owner
2. Manager
3. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi pembayaran piutang.

Program Studi Teknik Informatika	SKPL – SITB	76/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem akan menampilkan form pembayaran piutang.
3. Aktor menginputkan nomor faktur transaksi penjualan yang masih piutang.
4. Sistem mengecek inputan nomor faktur penjualan yang ada pada database.
E-1 Inputan nomor faktur salah.
5. Sistem menampilkan data piutang pada bidang isian.
6. Aktor menginputkan data jumlah pembayaran dan tanggal pembayaran.
E-2 Inputan jumlah pembayaran salah.
7. Sistem mengecek inputan aktor.
E-3 Inputan aktor salah.
8. Aktor klik tombol simpan untuk menyimpan data pembayaran piutang.
9. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

- E-1 Inputan nomor faktur salah.
1. Sistem menampilkan pesan kesalahan.
 2. Kembali ke Basic Flow langkah ke 4.
- E-2 Inputan jumlah pembayaran salah.
3. Sistem menampilkan pesan kesalahan.
 4. Kembali ke Basic Flow langkah ke 6.
- E-3 Inputan aktor salah.
3. Sistem menampilkan pesan kesalahan.
 4. Kembali ke Basic Flow langkah ke 6.

7. PreConditions

1. Aktor telah memasuki sistem

Program Studi Teknik Informatika	SKPL – SITB	77/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

8. PostConditions

1. Transaksi pembayaran piutang atas satu nomor faktur penjualan tersimpan ke dalam database

4.1.24 Use case Spesification : Transaksi Mutasi Gudang

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan transaksi mutasi gudang.

2. Primary Actor

1. Owner

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu transaksi mutasi gudang.
2. Sistem menampilkan form mutasi gudang.
3. Aktor klik tombol cari untuk memilih barang yang akan dimutasi.
4. Sistem menampilkan data barang pada bidang isian.
5. Aktor menginputkan jumlah barang dan data detail tujuan.
E-1 Jumlah barang yang diinputkan salah.
6. Aktor klik tombol simpan untuk menyimpan transaksi mutasi.
7. Sistem mengecek data yang diinputkan aktor.
E-2 Data yang diinputkan salah.
8. Sistem akan memberikan notifikasi bahwa data transaksi mutasi telah berhasil disimpan.
9. Use Case selesai.

5. Alternative Flow

Program Studi Teknik Informatika	SKPL – SITB	78/ 81
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

none

6. Error Flow

E-1 Jumlah barang yang diinputkan salah.

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 5.

E-2 Data yang diinputkan salah.

1. Sistem menampilkan pesan kesalahan.
2. Kembali ke Basic Flow langkah ke 5.

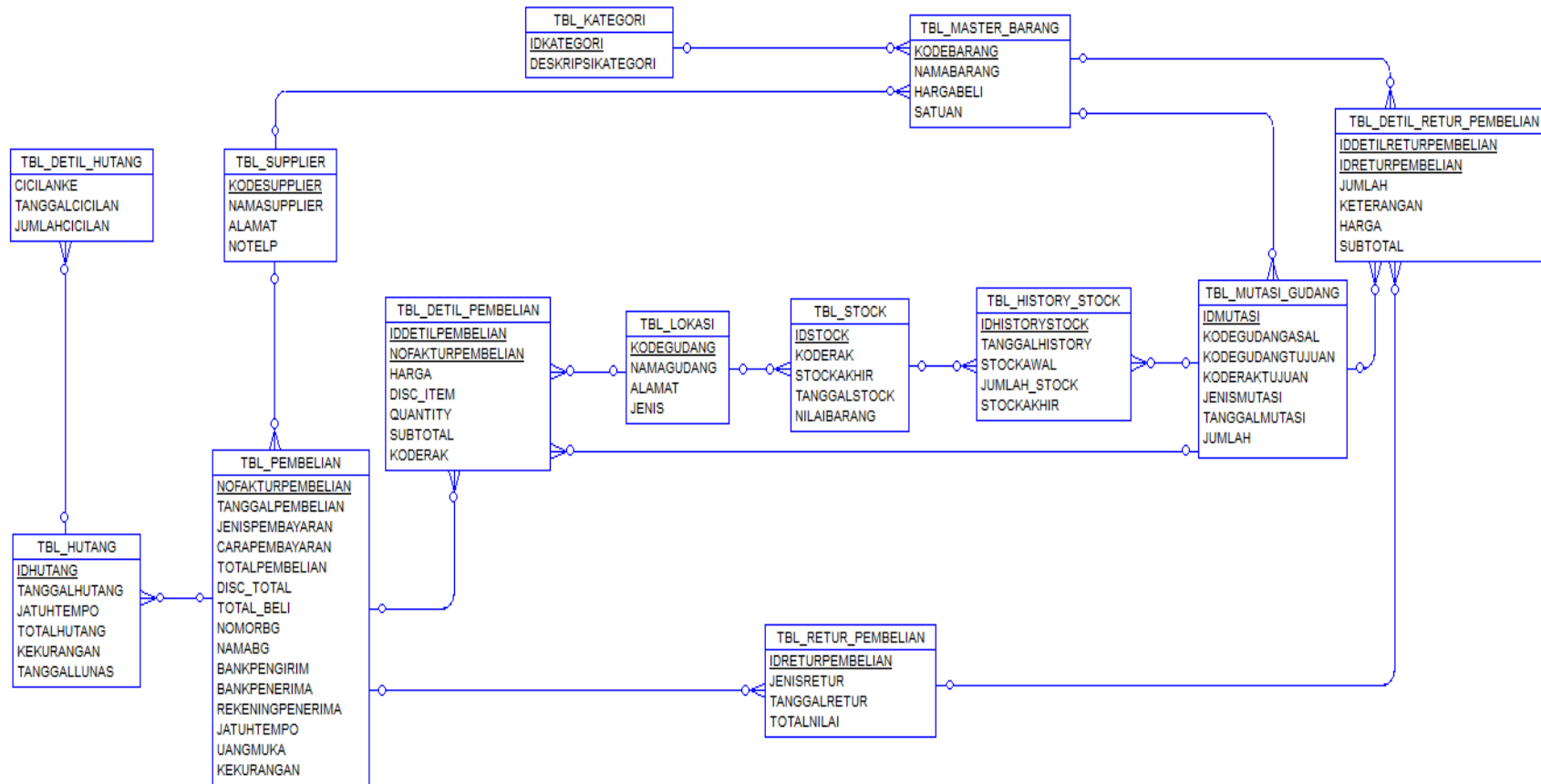
7. PreConditions

1. Aktor telah memasuki sistem

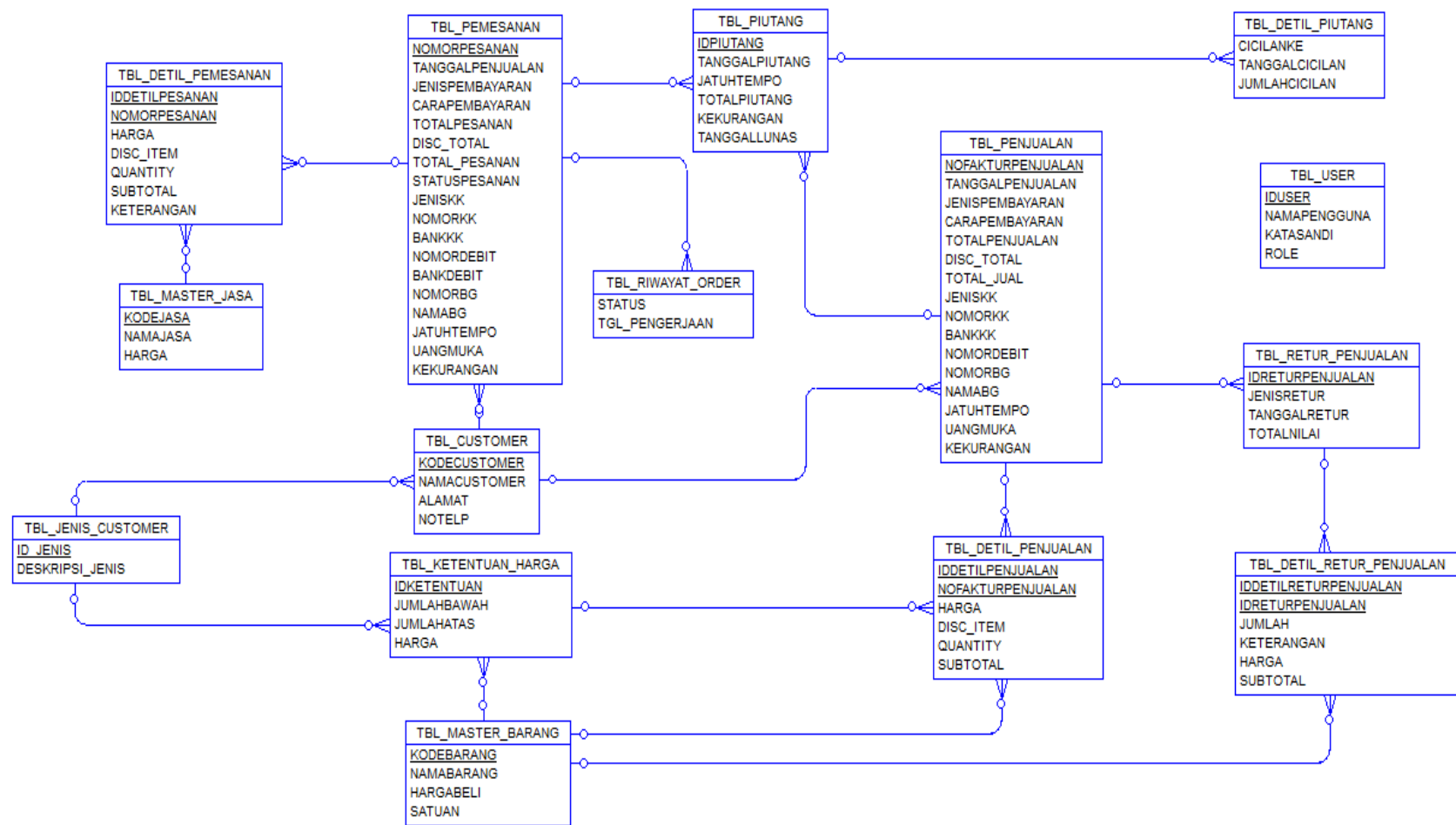
8. PostConditions

1. Data mutasi sudah tersimpan ke dalam database

5 ERD



Gambar 5.1 Entity Relationship Diagram



Gambar 5.2 Entity Relationship Diagram (lanjutan)

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SITB

(Sistem Informasi Toko Buku)

Untuk :


Toko Buku XYZ

Dipersiapkan oleh:

Yohanes Yanuar Adi N / 110706607

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
	Fakultas Teknologi Industri	DPPL SITB		1/122
		Revisi	-	

Program Studi Teknik Informatika	DPPL – SITB	1/122
----------------------------------	-------------	-------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	YYA N							
Diperik sa oleh								
Disetuj ui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	PENDAHULUAN	9
1.1	TUJUAN	9
1.2	LINGKUP MASALAH	9
1.3	DEFINISI, AKRONIM DAN SINGKATAN	10
1.4	REFERENSI	11
2	PERANCANGAN SISTEM	12
2.1	PERANCANGAN ARSITEKTUR	12
2.2	PERANCANGAN RINCI	14
2.2.1	<i>Sequence Diagram</i>	14
2.2.1.1	Login	14
2.2.1.2	Pengelolaan Data Barang	14
2.2.1.3	Pencarian History Stock	17
2.2.1.4	Pencarian Stock Barang	17
2.2.1.5	Cetak Label Barang	18
2.2.1.6	Transaksi Pembelian	19
2.2.1.7	Transaksi Retur Pembelian Keluar	20
2.2.1.8	Transaksi Retur Pembelian Masuk	21
2.2.1.9	Transaksi Pembayaran Hutang	22
2.2.1.10	Transaksi Penjualan	23
2.2.1.11	Transaksi Retur Penjualan	24
2.2.1.12	Transaksi Order	25
2.2.1.13	Transaksi Pemrosesan Order	26
2.2.1.14	Transaksi Pembatalan Order	26
2.2.1.15	Transaksi Pembayaran Piutang	27
2.2.1.16	Transaksi Mutasi Gudang	28
2.3	CLASS DIAGRAM	29
2.3.1	<i>Class Diagram Specific Description</i>	33
2.3.1.1	Specific Design Class LoginUI	33
2.3.1.2	Specific Design Class PengelolaanUserUI	33
2.3.1.3	Specific Design Class PengelolaanJenisCustomerUI	34
2.3.1.4	Specific Design Class PengelolaanCustomerUI	34
2.3.1.5	Specific Design Class PengelolaanKategoriBarangUI	35
2.3.1.6	Specific Design Class PengelolaanBarangUI	36
2.3.1.7	Specific Design Class PengelolaanSupplierUI	36
2.3.1.8	Specific Design Class PembelianUI	37
2.3.1.9	Specific Design Class ReturPembelianMasukUI	38
2.3.1.10	Specific Design Class ReturPembelianKeluarUI	38
2.3.1.11	Specific Design Class HutangUI	39
2.3.1.12	Specific Design Class CariStockUI	40
2.3.1.13	Specific Design Class DummyStockUI	40
2.3.1.14	Specific Design Class MutasiGudangUI	41
2.3.1.15	Specific Design Class PengelolaanLokasiUI	41
2.3.1.16	Specific Design Class HistoryStockUI	42
2.3.1.17	Specific Design Class PengelolaanKetentuanHargaUI	42
2.3.1.18	Specific Design Class PenjualanUI	43
2.3.1.19	Specific Design Class ReturPenjualanMasukUI	44
2.3.1.20	Specific Design Class PiutangUI	45
2.3.1.21	Specific Design Class Piutang_Control	46

2.3.1.22	Specific Design Class	ReturPenjualan_Control	47
2.3.1.23	Specific Design Class	Penjualan_Control	49
2.3.1.24	Specific Design Class	PengelolaanKetentuanHarga_Control	50
2.3.1.25	Specific Design Class	HistoryStock_Control	50
2.3.1.26	Specific Design Class	PengelolaanLokasi_Control	51
2.3.1.27	Specific Design Class	Mutasi_Control	52
2.3.1.28	Specific Design Class	DummyStock_Control	52
2.3.1.29	Specific Design Class	Stock_Control	53
2.3.1.30	Specific Design Class	PencarianStock_Control	53
2.3.1.31	Specific Design Class	Hutang_Control	54
2.3.1.32	Specific Design Class	ReturPembelian_Control	54
2.3.1.33	Specific Design Class	Pembelian_Control	56
2.3.1.34	Specific Design Class	PengelolaanSupplier_Control	57
2.3.1.35	Specific Design Class	PengelolaanBarang_Control	58
2.3.1.36	Specific Design Class	PengelolaanKategoriBarang_Control	58
2.3.1.37	Specific Design Class	PengelolaanCustomer_Control	59
2.3.1.38	Specific Design Class	PengelolaanJenisCustomer_Control	60
2.3.1.39	Specific Design Class	Login_Control	60
2.3.1.40	Specific Design Class	PengelolaanUser_Control	61
2.3.1.41	Specific Design Class	User	62
2.3.1.42	Specific Design Class	JenisCustomer	62
2.3.1.43	Specific Design Class	Customer	63
2.3.1.44	Specific Design Class	Barang	63
2.3.1.45	Specific Design Class	Barang	64
2.3.1.46	Specific Design Class	Supplier	64
2.3.1.47	Specific Design Class	Pembelian	65
2.3.1.48	Specific Design Class	DetilPembelian	67
2.3.1.49	Specific Design Class	ReturPembelian	68
2.3.1.50	Specific Design Class	DetilReturPembelian	69
2.3.1.51	Specific Design Class	Hutang	70
2.3.1.52	Specific Design Class	DetilHutang	71
2.3.1.53	Specific Design Class	Stock	71
2.3.1.54	Specific Design Class	Mutasi	72
2.3.1.55	Specific Design Class	Lokasi	73
2.3.1.56	Specific Design Class	HistoryStock	73
2.3.1.57	Specific Design Class	KetentuanHarga	74
2.3.1.58	Specific Design Class	Penjualan	75
2.3.1.59	Specific Design Class	DetilPenjualan	77
2.3.1.60	Specific Design Class	ReturPenjualan	78
2.3.1.61	Specific Design Class	DetilReturPenjualan	79
2.3.1.62	Specific Design Class	Piutang	80
2.3.1.63	Specific Design Class	DetilPiutang	81
2.3.1.64	Specific Design Class	Jasa	81
2.3.1.65	Specific Design Class	DetilOrder	82
2.3.1.66	Specific Design Class	Order	83
2.3.1.67	Specific Design Class	RiwayatOrder	85
3	PERANCANGAN DATA		86
3.1	DEKOMPOSISI DATA		86
3.1.1	Deskripsi Entitas TBL_USER		86
3.1.2	Deskripsi Entitas TBL_SUPPLIER		86
3.1.3	Deskripsi Entitas TBL_STOCK		86
3.1.4	Deskripsi Entitas TBL_RIWAYAT_ORDER		86
3.1.5	Deskripsi Entitas TBL_RETUR_PENJUALAN		87

3.1.6	Deskripsi Entitas	TBL_RETUR_PEMBELIAN.....	87
3.1.7	Deskripsi Entitas	TBL_PIUTANG.....	87
3.1.8	Deskripsi Entitas	TBL_PENJUALAN.....	87
3.1.9	Deskripsi Entitas	TBL_PEMESANAN.....	88
3.1.10	Deskripsi Entitas	TBL_PEMBELIAN.....	89
3.1.11	Deskripsi Entitas	TBL_MUTASI_GUDANG.....	89
3.1.12	Deskripsi Entitas	TBL_MASTER_JASA.....	90
3.1.13	Deskripsi Entitas	TBL_MASTER_BARANG.....	90
3.1.14	Deskripsi Entitas	TBL_LOKASI.....	90
3.1.15	Deskripsi Entitas	TBL_KETENTUAN_HARGA.....	90
3.1.16	Deskripsi Entitas	TBL_KATEGORI.....	90
3.1.17	Deskripsi Entitas	TBL_JENIS_CUSTOMER.....	91
3.1.18	Deskripsi Entitas	TBL_HUTANG.....	91
3.1.19	Deskripsi Entitas	TBL_HISTORY_STOCK.....	91
3.1.20	Deskripsi Entitas	TBL_DETIL_RETUR_PENJUALAN.....	91
3.1.21	Deskripsi Entitas	TBL_DETIL_RETUR_PEMBELIAN.....	92
3.1.22	Deskripsi Entitas	TBL_DETIL_PIUTANG.....	92
3.1.23	Deskripsi Entitas	TBL_DETIL_PENJUALAN.....	92
3.1.24	Deskripsi Entitas	TBL_DETIL_PEMESANAN.....	92
3.1.25	Deskripsi Entitas	TBL_DETIL_PEMBELIAN.....	93
3.1.26	Deskripsi Entitas	TBL_DETIL_HUTANG.....	93
3.1.27	Deskripsi Entitas	TBL_CUSTOMER.....	93
3.2	PHYSICAL DATA MODEL.....		94
4	DESKRIPSI PERANCANGAN ANTARMUKA.....		96
4.1	ANTARMUKA HALAMAN LOGIN.....		96
4.2	ANTARMUKA HALAMAN UTAMA.....		97
4.3	ANTARMUKA HALAMAN PENGELOLAAN USER.....		98
4.4	ANTARMUKA HALAMAN PENGELOLAAN CUSTOMER.....		99
4.5	ANTARMUKA HALAMAN PENGELOLAAN SUPPLIER.....		100
4.6	ANTARMUKA HALAMAN PENGELOLAAN JENIS CUSTOMER.....		101
4.7	ANTARMUKA HALAMAN PENGELOLAAN KATEGORI BARANG.....		102
4.8	ANTARMUKA HALAMAN PENGELOLAAN BARANG.....		103
4.9	ANTARMUKA HALAMAN PENGELOLAAN JASA.....		105
4.10	ANTARMUKA HALAMAN PENGELOLAAN LOKASI.....		106
4.11	ANTARMUKA HALAMAN PENGELOLAAN KETENTUAN HARGA.....		107
4.12	ANTARMUKA HALAMAN PENCARIAN HISTORY STOCK.....		108
4.13	ANTARMUKA HALAMAN PENCARIAN STOCK.....		109
4.14	ANTARMUKA HALAMAN CETAK LABEL BARANG.....		110
4.15	ANTARMUKA HALAMAN PEMBELIAN.....		111
4.16	ANTARMUKA HALAMAN RETUR PEMBELIAN KELUAR.....		112
4.17	ANTARMUKA HALAMAN RETUR PEMBELIAN MASUK.....		113
4.18	ANTARMUKA HALAMAN PEMBAYARAN HUTANG.....		114
4.19	ANTARMUKA HALAMAN PENJUALAN.....		115
4.20	ANTARMUKA HALAMAN RETUR PENJUALAN.....		116
4.21	ANTARMUKA HALAMAN ORDER BARU.....		117
4.22	ANTARMUKA HALAMAN PROSES ORDER.....		118
4.23	ANTARMUKA HALAMAN PEMBATALAN ORDER.....		119
4.24	ANTARMUKA HALAMAN PEMBAYARAN PIUTANG.....		120
4.25	ANTARMUKA HALAMAN MUTASI GUDANG.....		121
4.26	ANTARMUKA HALAMAN REPORTING.....		122

Daftar Gambar

Gambar 2.1 Perancangan Kelas.....	12
Gambar 2.2 Perancangan Kelas (lanjutan).....	13
Gambar 2.3 Sequence Diagram : Login.....	14
Gambar 2.4 Sequence Diagram : Entri Data Barang.....	14
Gambar 2.5 Sequence Diagram : Edit Data Barang.....	15
Gambar 2.6 Sequence Diagram : Delete Data Barang.....	15
Gambar 2.7 Sequence Diagram : Display Data Barang.....	16
Gambar 2.8 Sequence Diagram : Cari Data Barang.....	16
Gambar 2.9 Sequence Diagram : Pencarian History Stock.....	17
Gambar 2.10 Sequence Diagram : Pencarian Stock Barang.....	17
Gambar 2.11 Sequence Diagram : Cetak Label Barang.....	18
Gambar 2.12 Sequence Diagram : Transaksi Pembelian.....	19
Gambar 2.13 Sequence : Transaksi Retur Pembelian Keluar...20	
Gambar 2.14 Sequence : Transaksi Retur Pembelian Masuk...21	
Gambar 2.15 Sequence Diagram : Transaksi Pembayaran Hutang.22	
Gambar 2.16 Sequence Diagram : Transaksi Penjualan.....23	
Gambar 2.17 Sequence Diagram : Transaksi Retur Penjualan..24	
Gambar 2.18 Sequence Diagram : Transaksi Order.....25	
Gambar 2.19 Sequence Diagram : Transaksi Pembatalan Order.26	
Gambar 2.20 Sequence Diagram : Transaksi Pembatalan Order.26	
Gambar 2.21 Sequence : Transaksi Pembayaran Piutang.....27	
Gambar 2.22 Sequence Diagram : Transaksi Mutasi Gudang....28	
Gambar 2.23 Class Diagram.....29	
Gambar 2.24 Class Diagram (lanjutan 1).....30	
Gambar 2.25 Class Diagram (lanjutan 2).....31	
Gambar 2.26 Class Diagram (lanjutan 3).....32	
Gambar 3.1 Physical Data Model.....95	
Gambar 3.2 Physical Data Model (lanjutan).....96	
Gambar 4.1 Antarmuka Login.....97	
Gambar 4.2 Antarmuka Halaman Utama.....98	
Gambar 4.3 Antarmuka Pengelolaan User.....99	
Gambar 4.4 Antarmuka Pengelolaan Customer.....100	
Gambar 4.5 Antarmuka Pengelolaan Supplier.....101	
Gambar 4.6 Antarmuka Pengelolaan Jenis Customer.....102	
Gambar 4.7 Antarmuka Pengelolaan Kategori Barang.....103	
Gambar 4.8 Antarmuka Pengelolaan Barang.....104	
Gambar 4.9 Antarmuka Pengelolaan Jasa.....106	
Gambar 4.10 Antarmuka Pengelolaan Lokasi.....107	
Gambar 4.11 Antarmuka Pengelolaan Ketentuan Harga.....108	
Gambar 4.12 Antarmuka Pencarian History Stock.....109	
Gambar 4.13 Antarmuka Pencarian Stock.....110	
Gambar 4.14 Antarmuka Cetak Label Barang.....111	
Gambar 4.15 Antarmuka Pembelian.....112	
Gambar 4.16 Antarmuka Retur Pembelian Keluar.....113	
Gambar 4.17 Antarmuka Retur Pembelian Masuk.....114	
Gambar 4.18 Antarmuka Pembayaran Hutang.....115	

Program Studi Teknik Informatika	DPPL – SITB	7/ 122
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Gambar 4.19	Antarmuka Penjualan.....	116
Gambar 4.20	Antarmuka Retur Penjualan.....	118
Gambar 4.21	Antarmuka Order Baru.....	118
Gambar 4.22	Antarmuka Proses Order.....	119
Gambar 4.23	Antarmuka Pembatalan Order.....	120
Gambar 4.24	Antarmuka Pembayaran Piutang.....	121
Gambar 4.25	Antarmuka Mutasi Gudang.....	122
Gambar 4.26	Antarmuka Reporting.....	123



1 Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini bertujuan untuk mendefinisikan deskripsi perancangan perangkat lunak SITB (Sistem Informasi Toko Buku). DPPL-SITB ini digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap berikutnya.

1.2 Lingkup Masalah

Perangkat Lunak SITB dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan transaksi pembelian, retur pembelian, hutang, penjualan, retur penjualan, order, dan piutang yang terjadi pada Toko Buku XYZ.
2. Menangani pengelolaan inventory dari beberapa gudang.
3. Menangani pencatatan proses keluar dan masuk barang dari dan atau menuju gudang.
4. Menangani reporting.

Dan berjalan pada lingkungan dekstop Windows dan menggunakan Microsoft SQL Server 2005 sebagai DBMS.

Program Studi Teknik Informatika	DPPL – SITB	9/ 122
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Merupakan deskripsi dari perancangan perangkat lunak yang akan dikembangkan.
DPPL-SITB-XXX	Kode yang merepresentasikan kebutuhan pada SITB (Sistem Informasi Toko Buku) dimana XXX merupakan nomor fungsi produk.
SITB	Perangkat lunak yang dikembangkan untuk menangani pengelolaan proses transaksi Toko Buku XYZ.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Barcode	Kode yang berupa kombinasi garis hitam dan putih dengan berbagai kombinasi jarak untuk mengkodekan angka atau huruf.
Barcode scanner	Alat yang digunakan untuk membaca data yang ada didalam kode barcode tersebut dan mengirimkan data tersebut ke komputer.

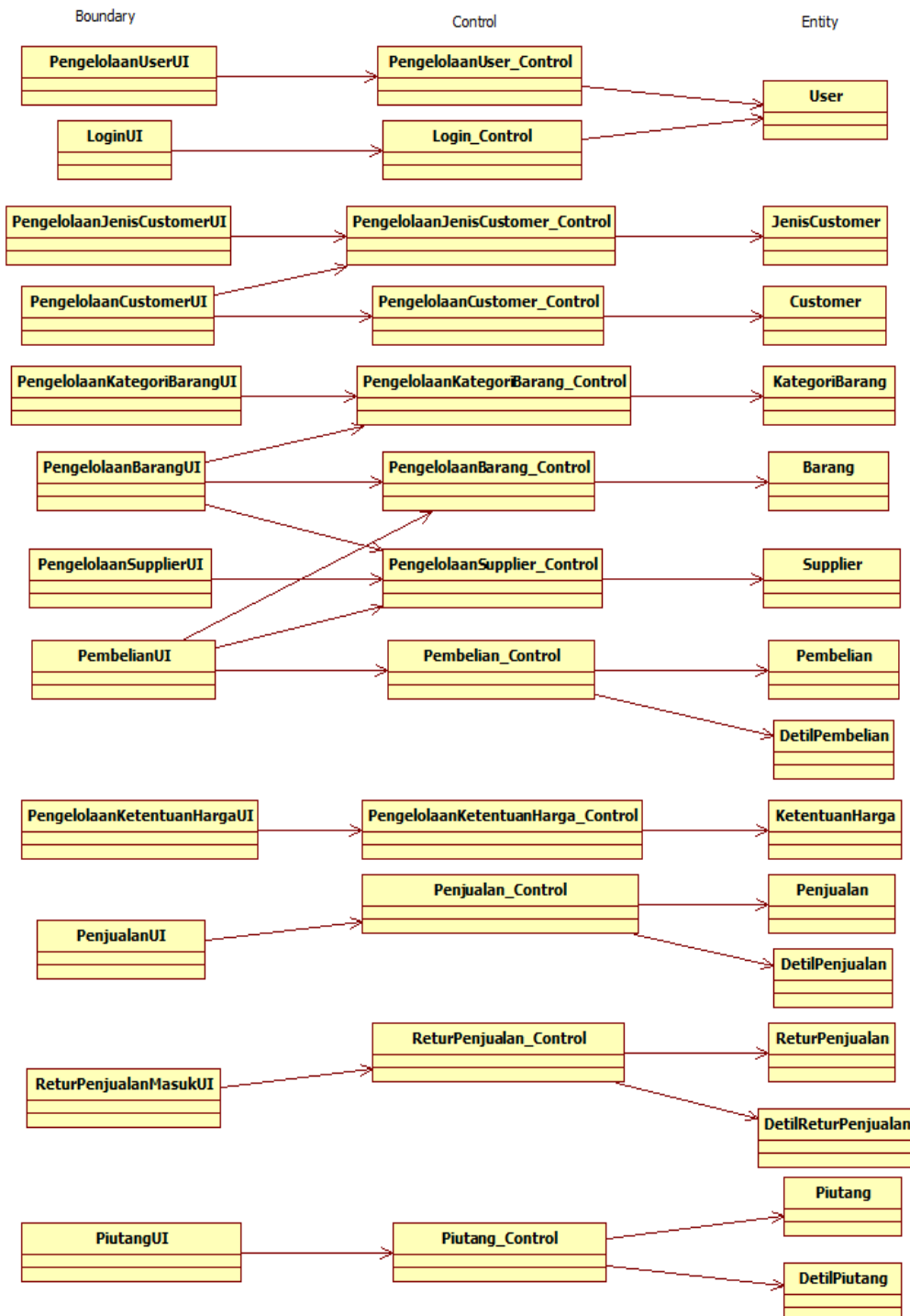
1.4 Referensi

Referensi yang digunakan pada SKPL ini adalah:

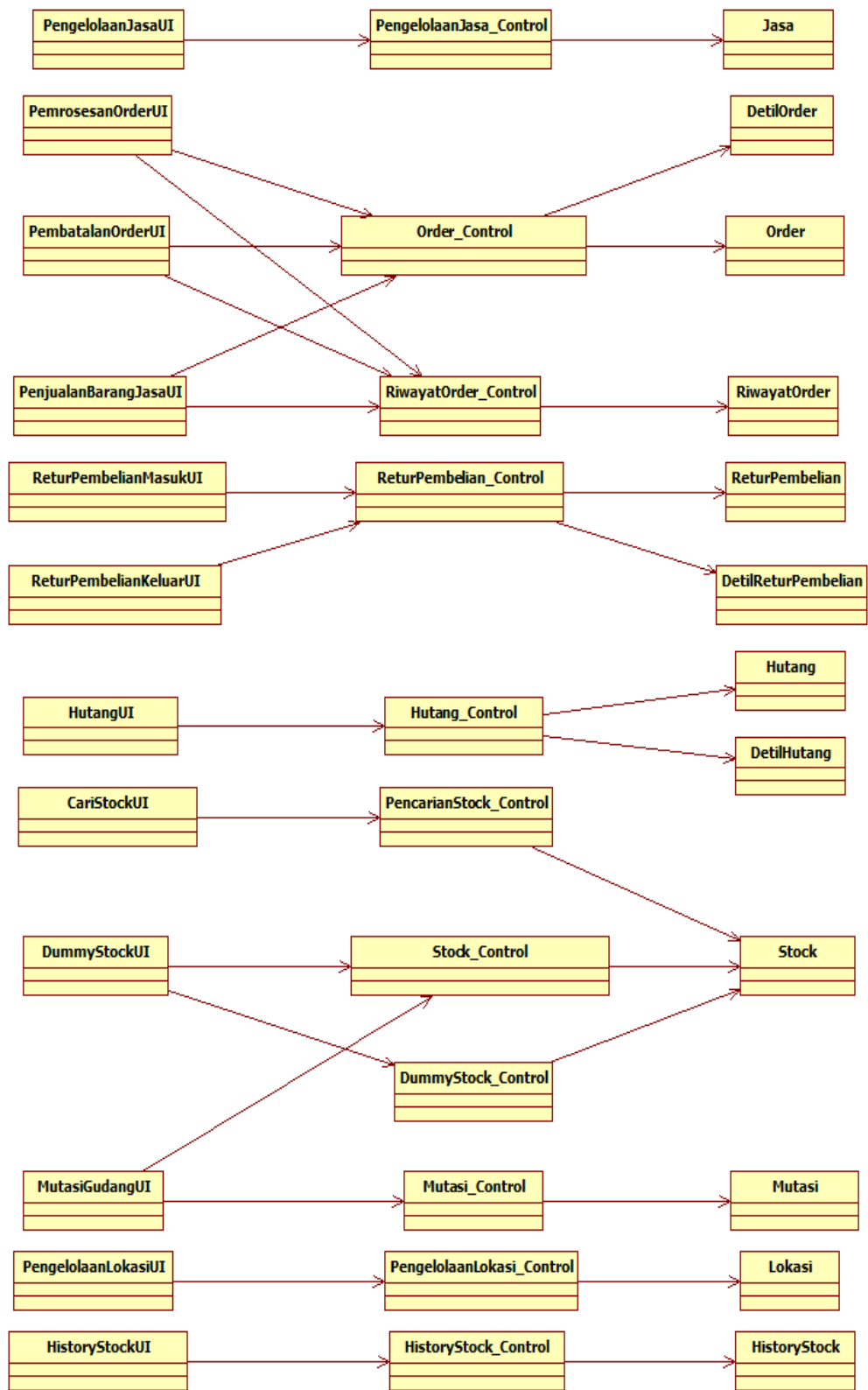
1. Damanik, Rudolfo Rizki. *Deskripsi Kebutuhan Perangkat Lunak (DPPL) Automatic Report System (ARS)*, Universitas Atma Jaya Yogyakarta, 2012.
2. Michael Aristian S, *Deskripsi Kebutuhan Perangkat Lunak (DPPL) LaSerS*, Universitas Atma Jaya Yogyakarta, 2012.
3. Nugroho, Yohanes Yanuar Adi. *Deskripsi Kebutuhan Perangkat Lunak (DPPL) Sistem Informasi Kana Jaya (SIKAJA)*, Universitas Atma Jaya Yogyakarta, 2013.

2 Perancangan Sistem

2.1 Perancangan Arsitektur



Gambar 2.1 Perancangan Kelas

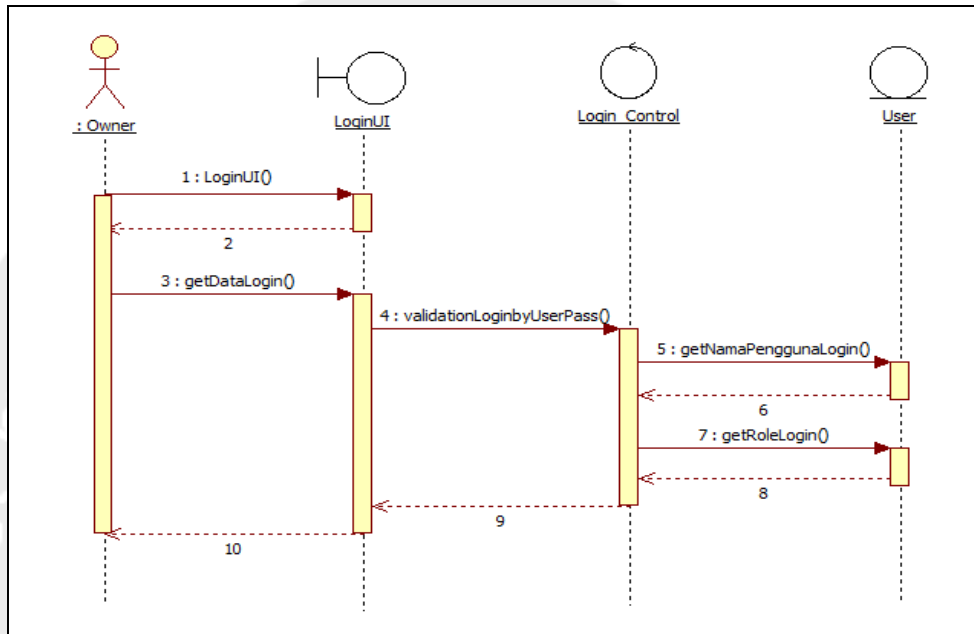


Gambar 2.2 Perancangan Kelas (lanjutan)

2.2 Perancangan Rinci

2.2.1 Sequence Diagram

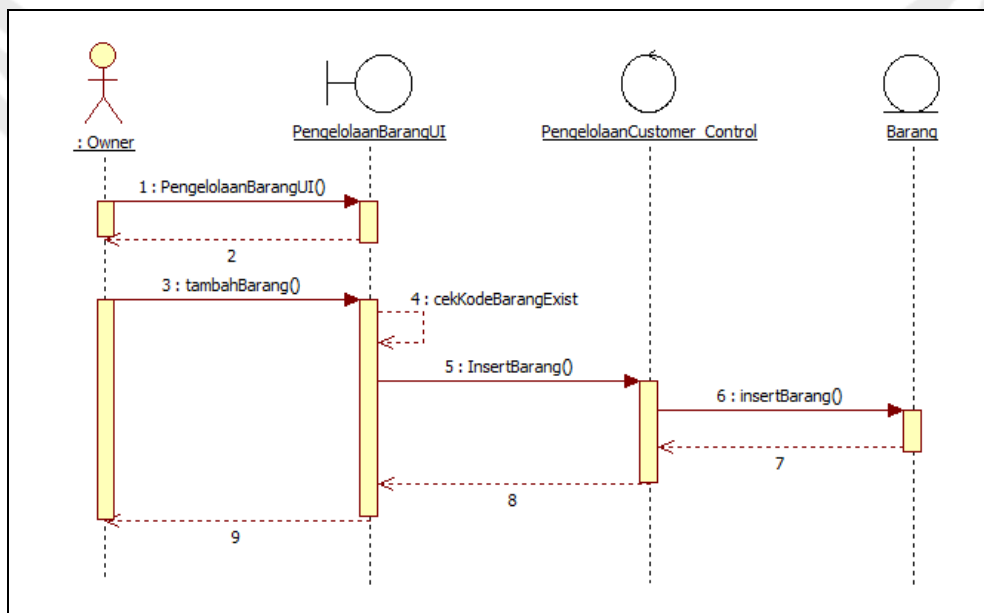
2.2.1.1 Login



Gambar 2.3 Sequence Diagram : Login

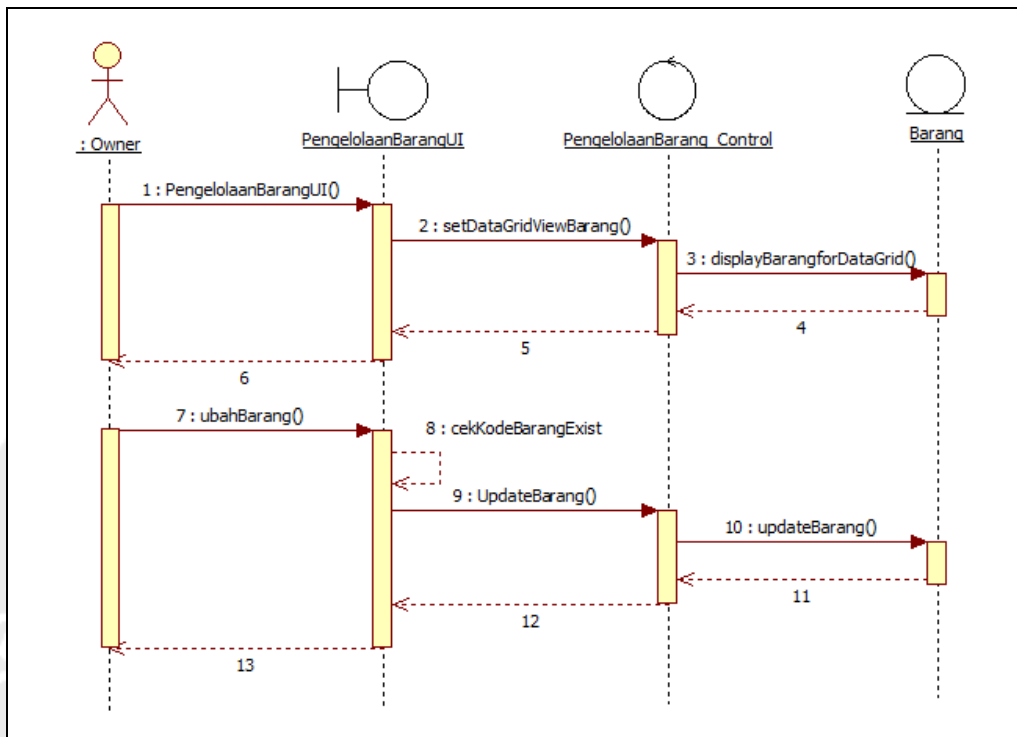
2.2.1.2 Pengelolaan Data Barang

2.2.1.2.1 Entri



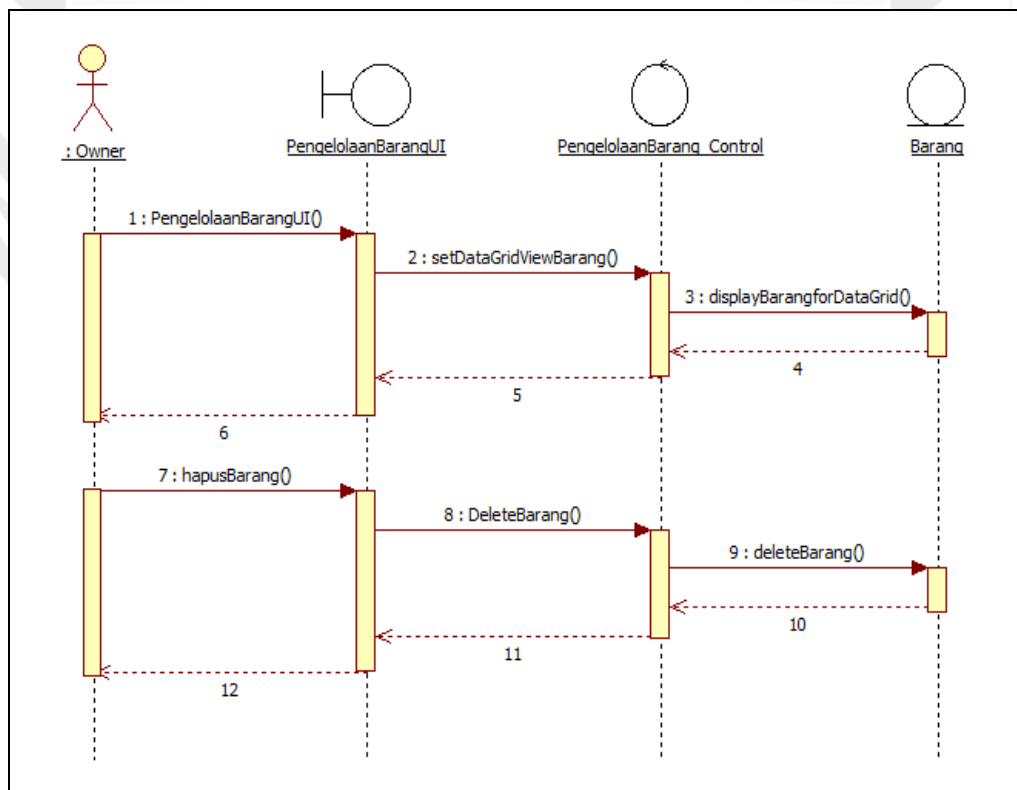
Gambar 2.4 Sequence Diagram : Entri Data Barang

2.2.1.2.2 Edit



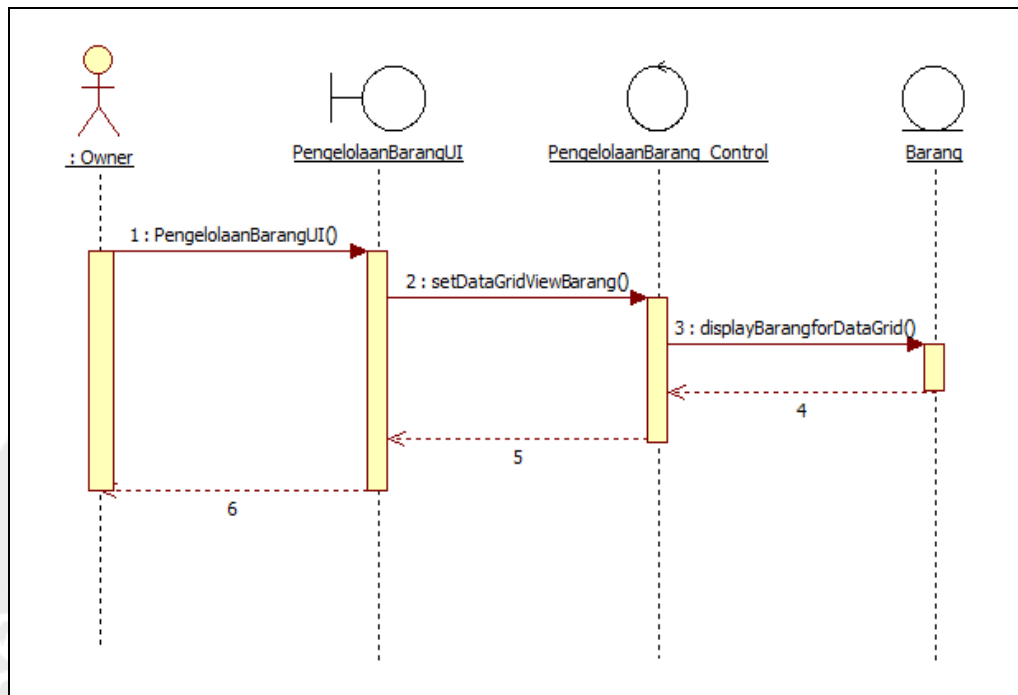
Gambar 2.5 Sequence Diagram : Edit Data Barang

2.2.1.2.3 Delete



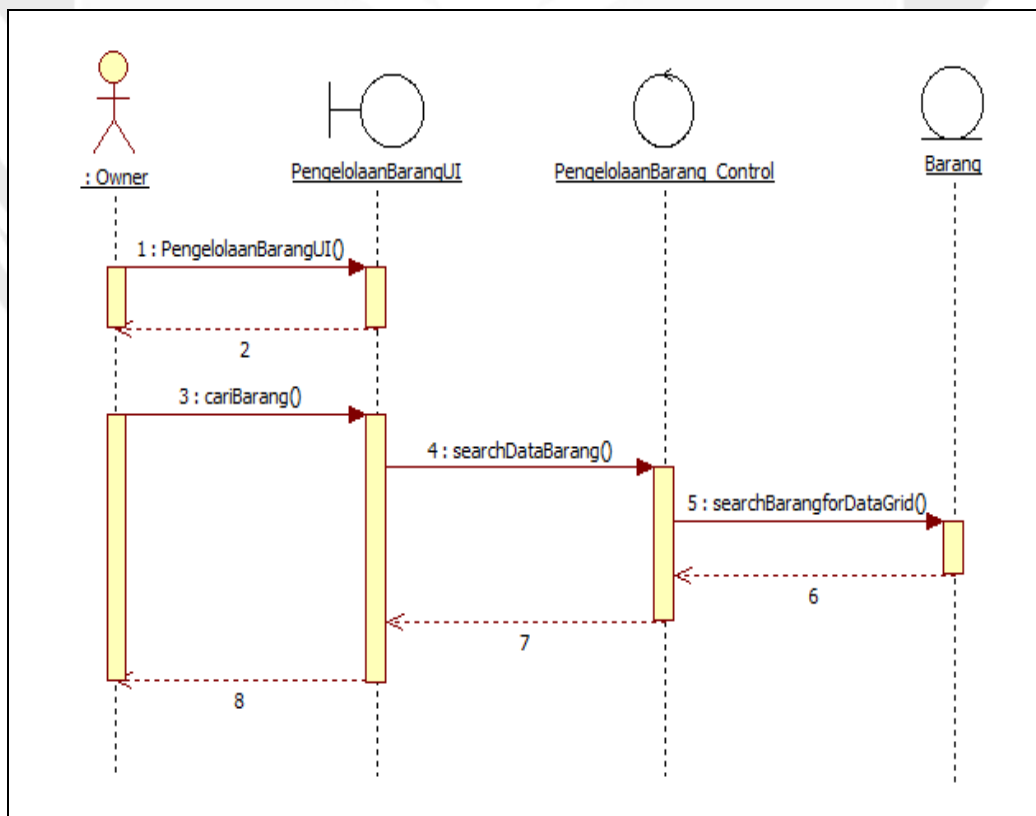
Gambar 2.6 Sequence Diagram : Delete Data Barang

2.2.1.2.4 Display



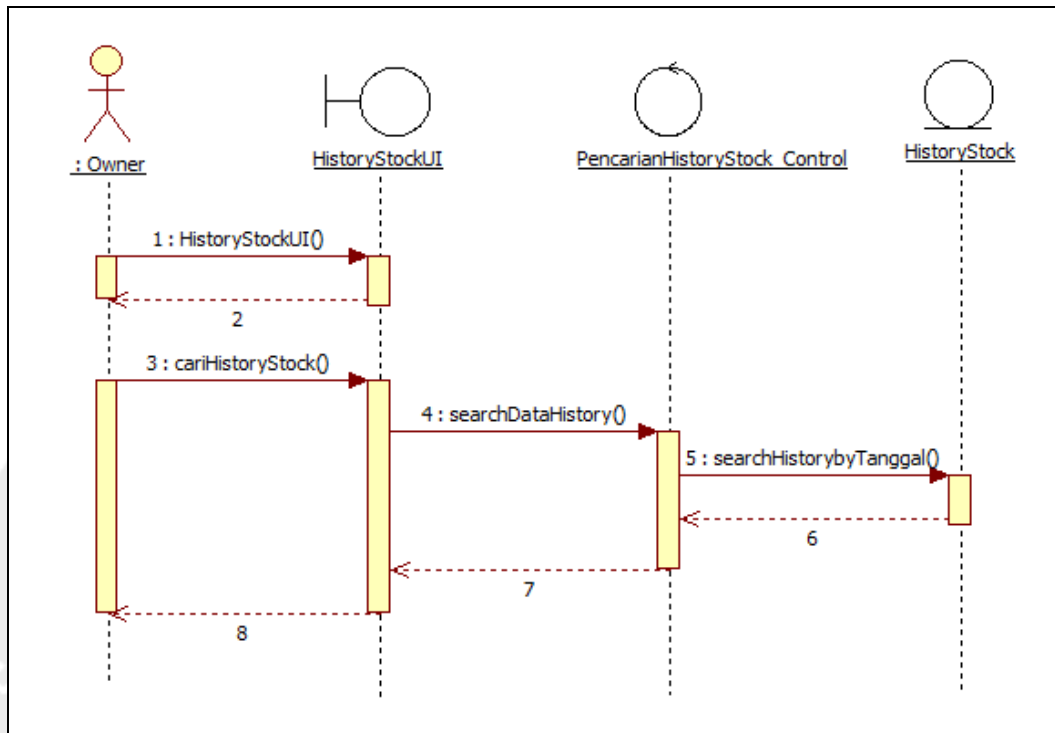
Gambar 2.7 Sequence Diagram : Display Data Barang

2.2.1.2.5 Cari



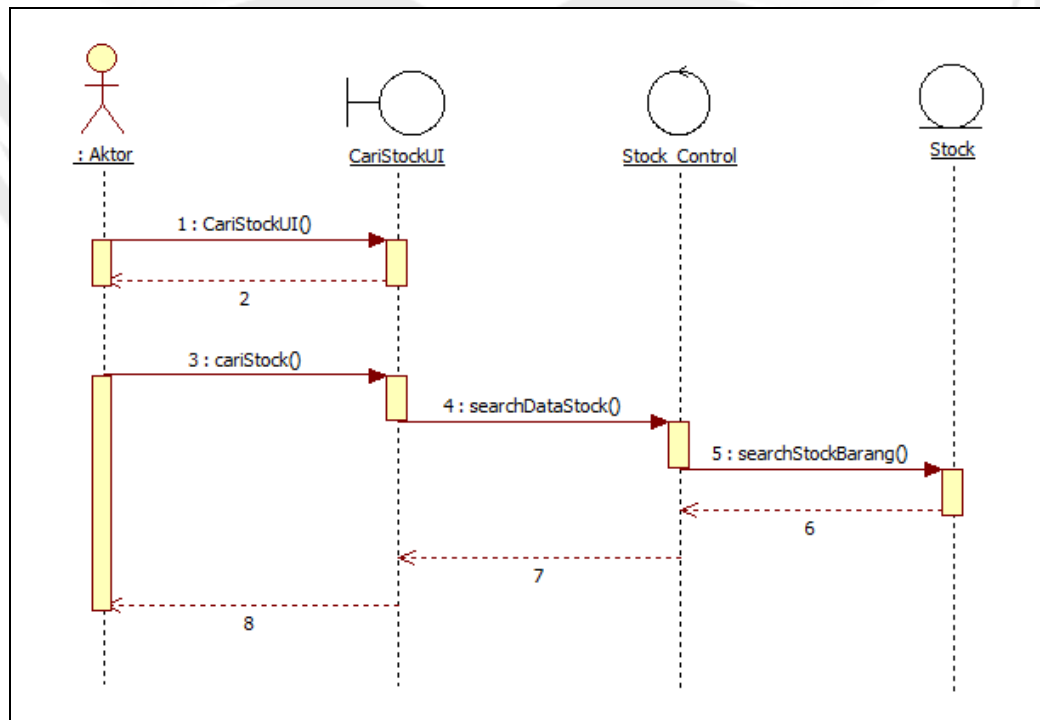
Gambar 2.8 Sequence Diagram : Cari Data Barang

2.2.1.3 Pencarian History Stock



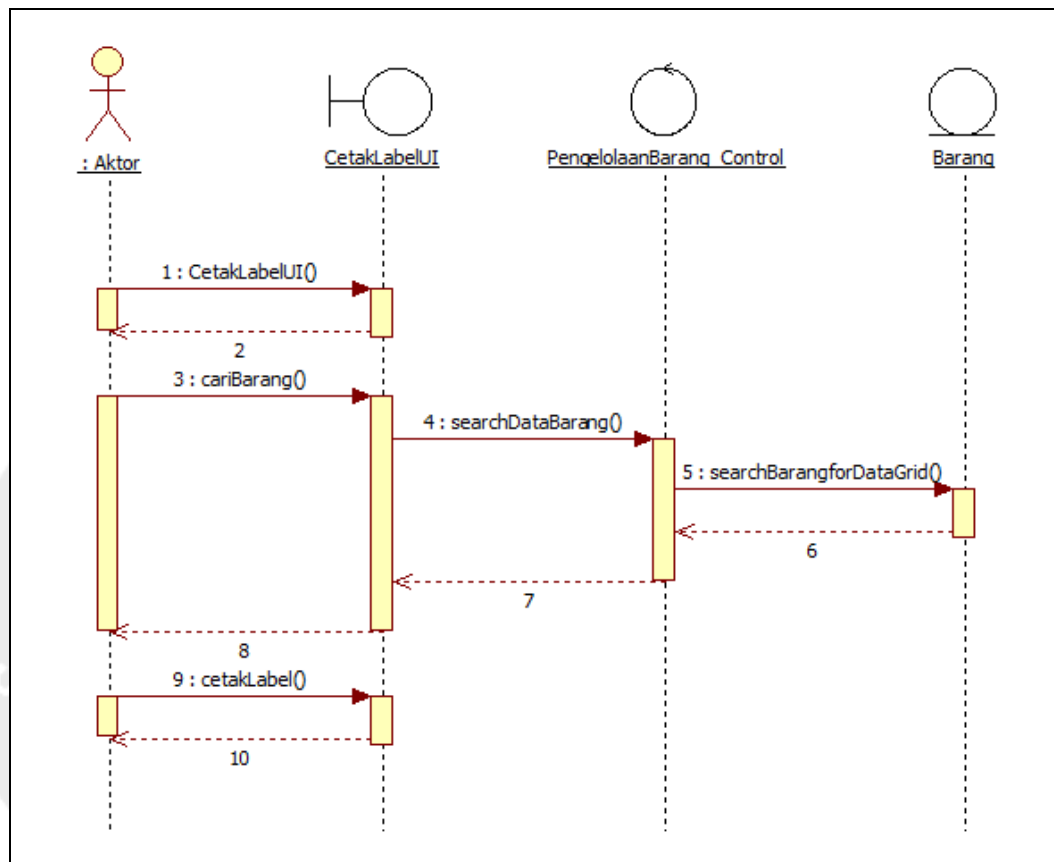
Gambar 2.9 Sequence Diagram : Pencarian History Stock

2.2.1.4 Pencarian Stock Barang



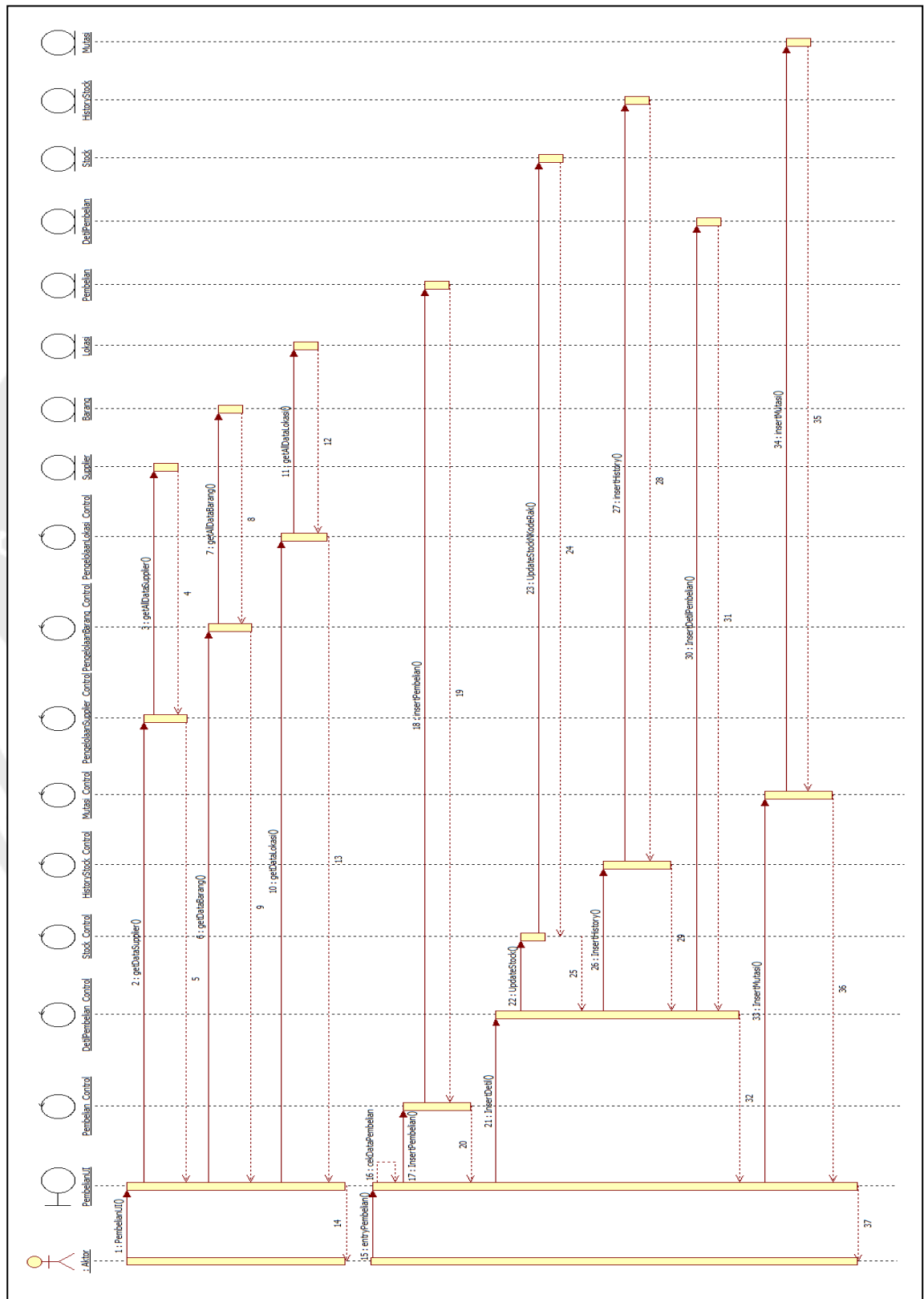
Gambar 2.10 Sequence Diagram : Pencarian Stock Barang

2.2.1.5 Cetak Label Barang



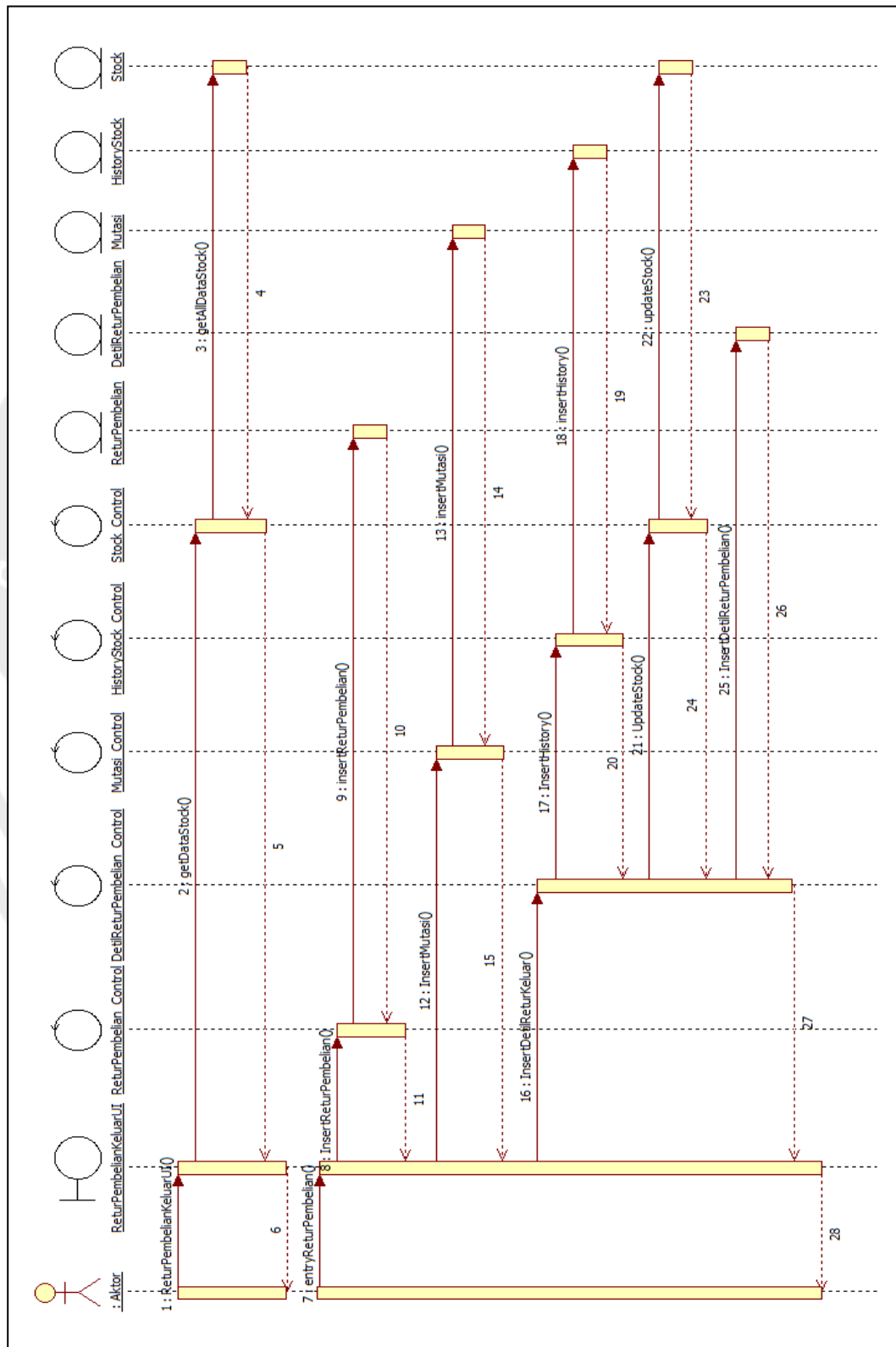
Gambar 2.11 Sequence Diagram : Cetak Label Barang

2.2.1.6 Transaksi Pembelian



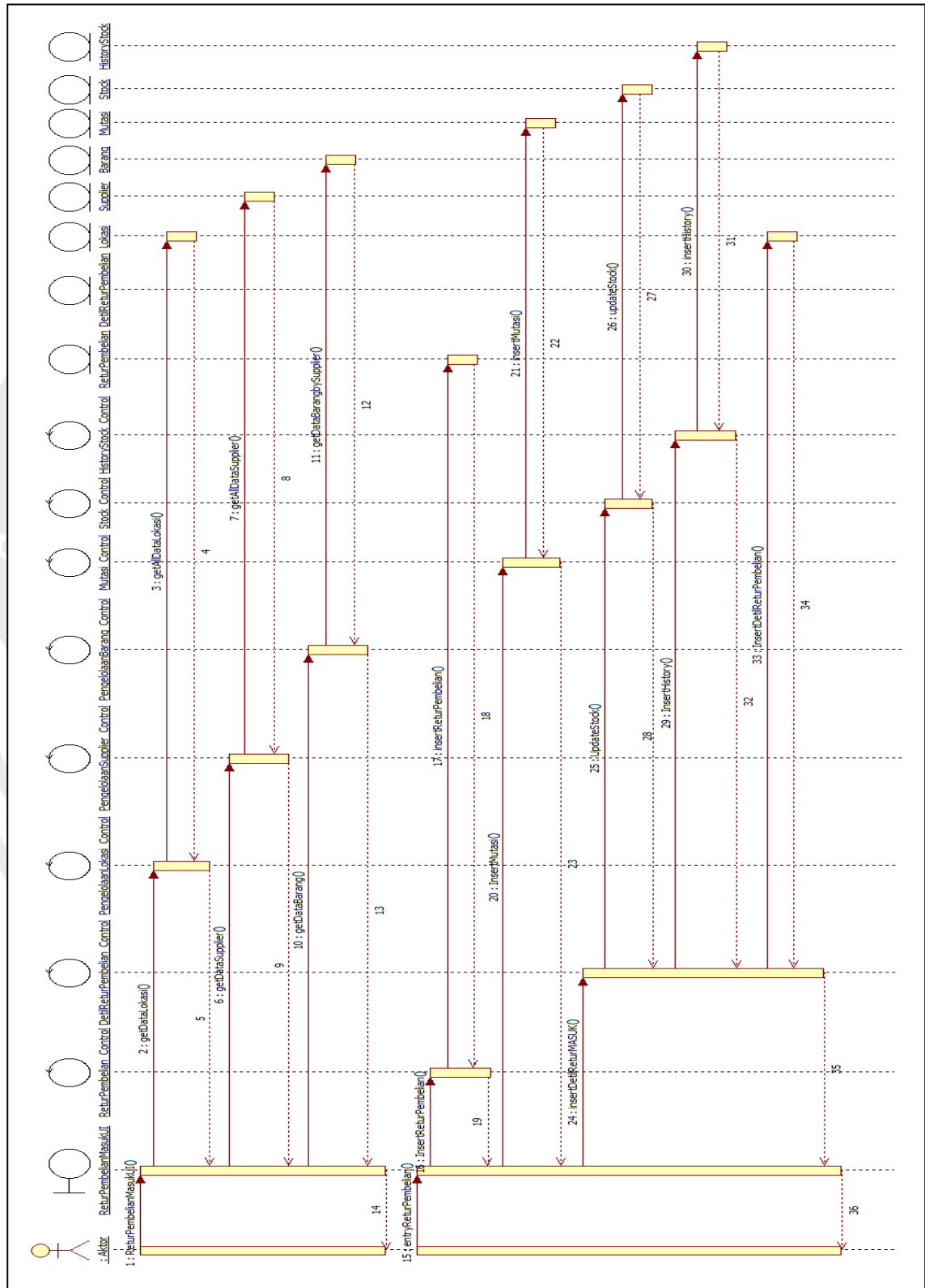
Gambar 2.12 Sequence Diagram : Transaksi Pembelian

2.2.1.7 Transaksi Retur Pembelian Keluar



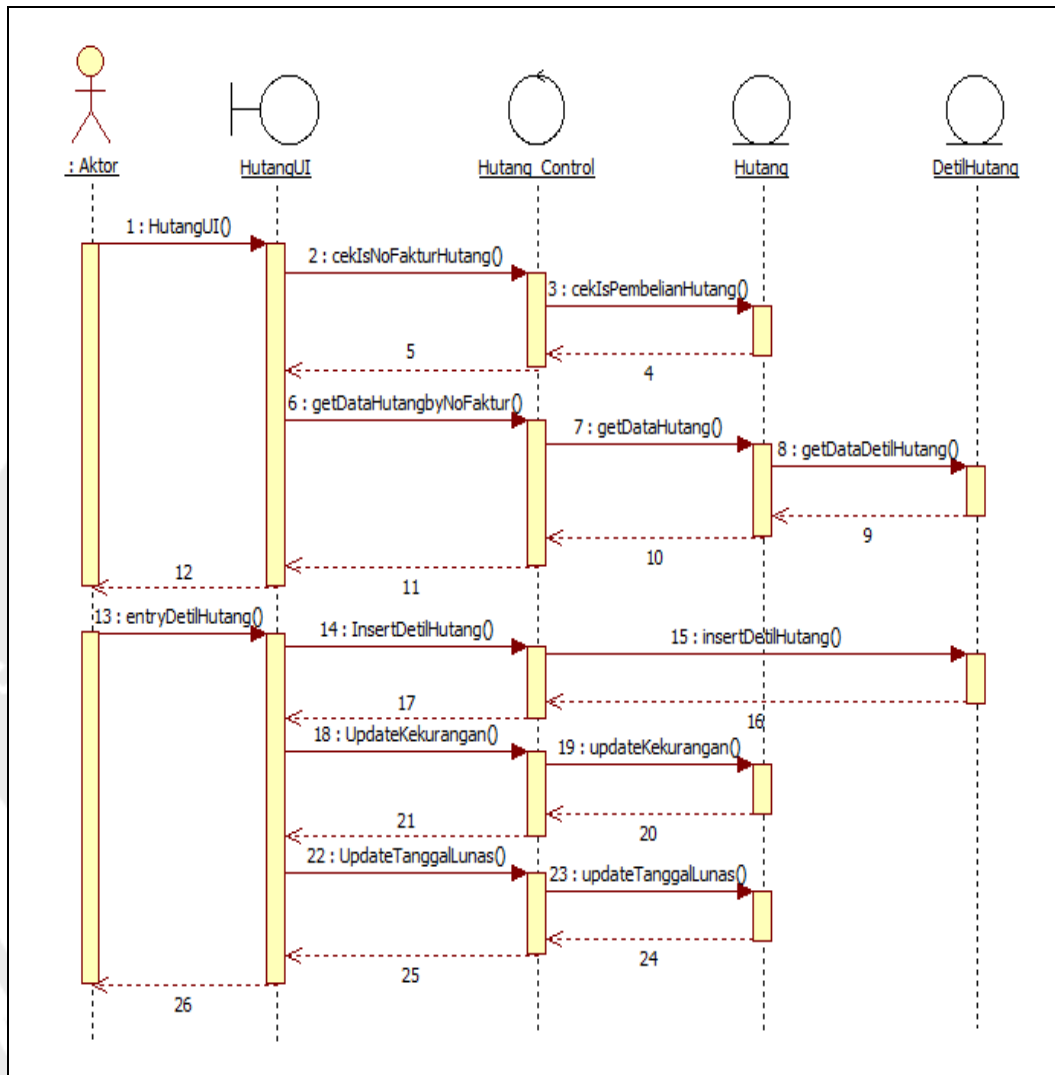
Gambar 2.13 Sequence Diagram : Transaksi Retur Pembelian Keluar

2.2.1.8 Transaksi Retur Pembelian Masuk



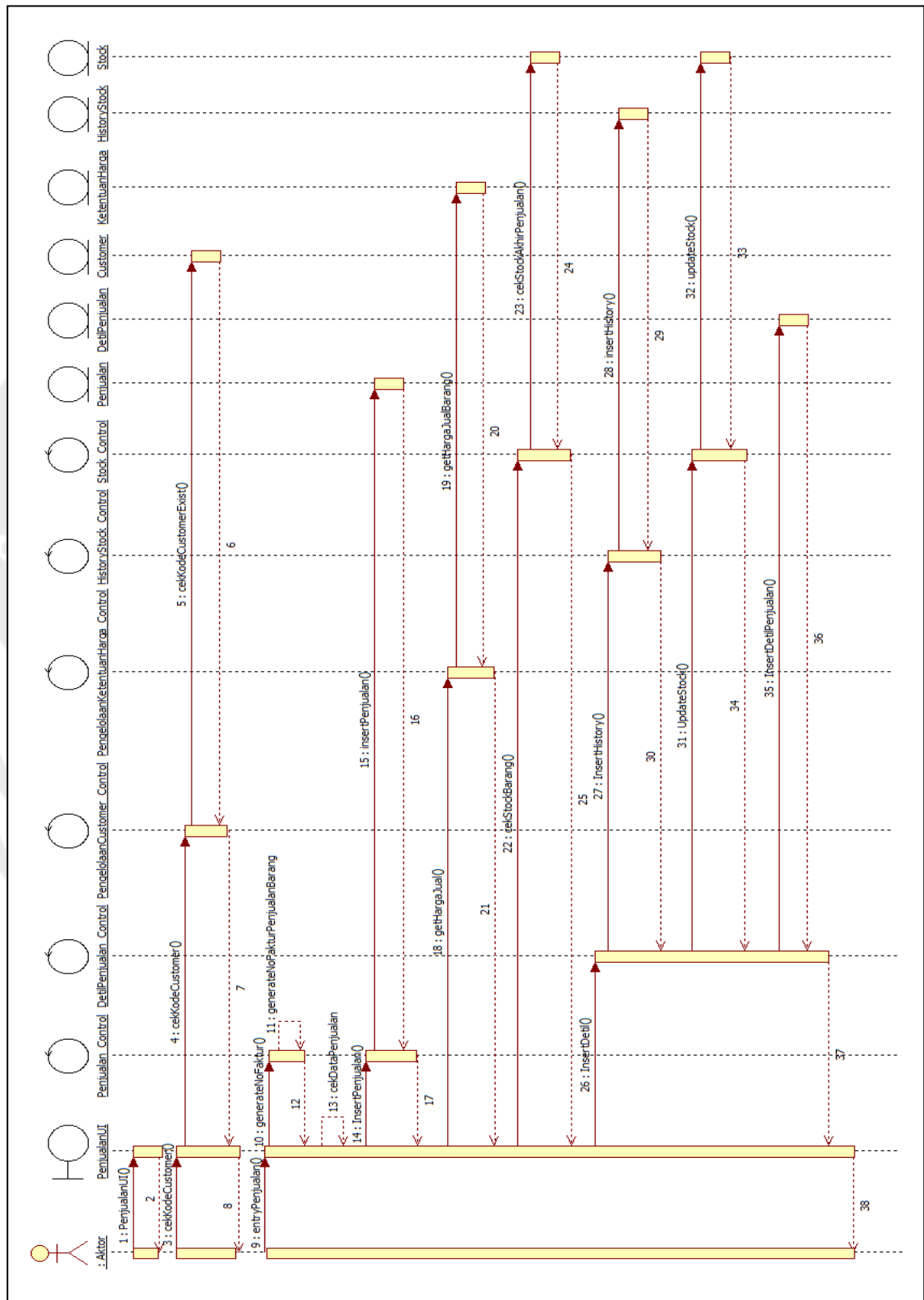
Gambar 2.14 Sequence Diagram : Transaksi Retur Pembelian Masuk

2.2.1.9 Transaksi Pembayaran Hutang



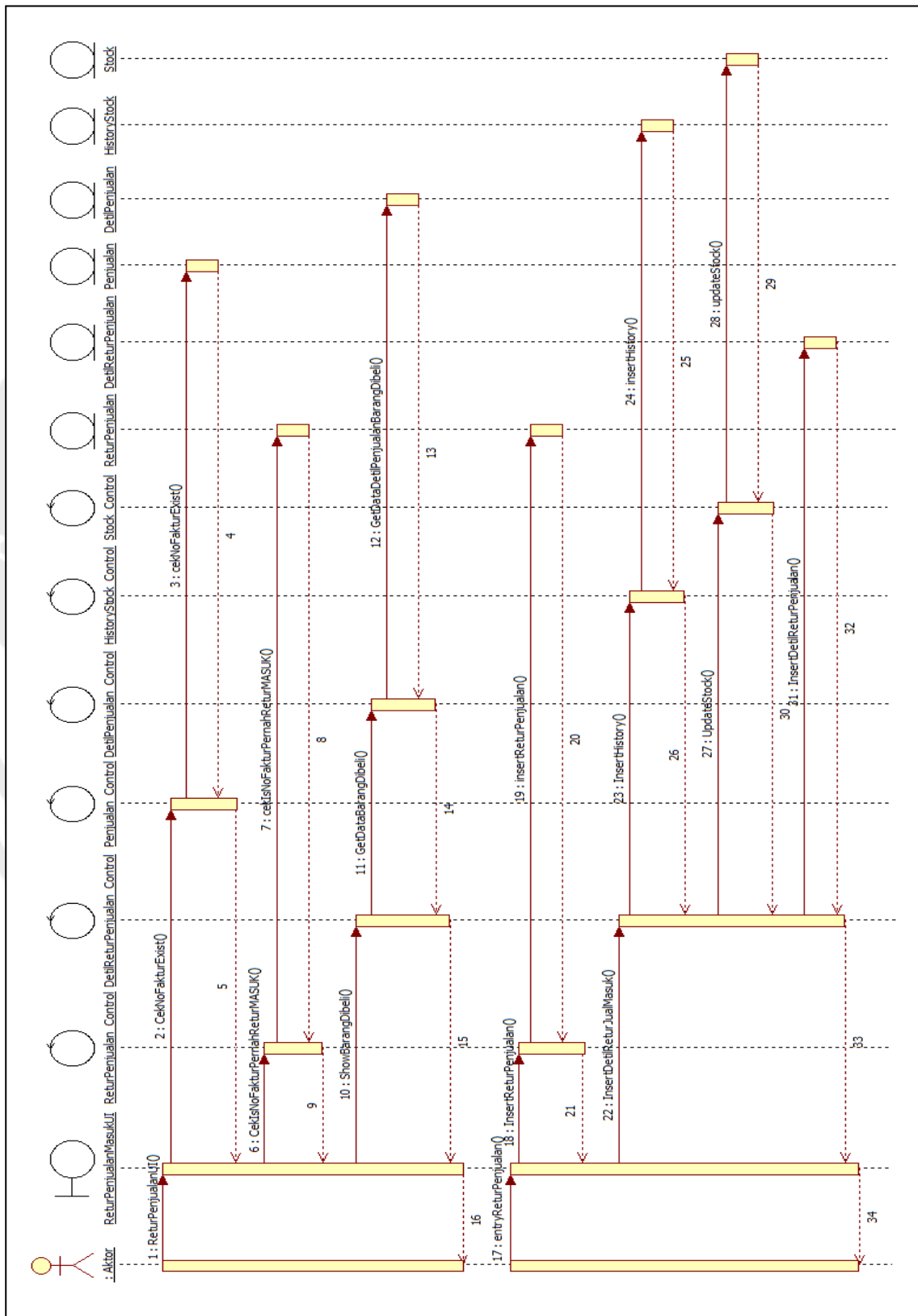
Gambar 2.15 Sequence Diagram : Transaksi Pembayaran Hutang

2.2.1.10 Transaksi Penjualan



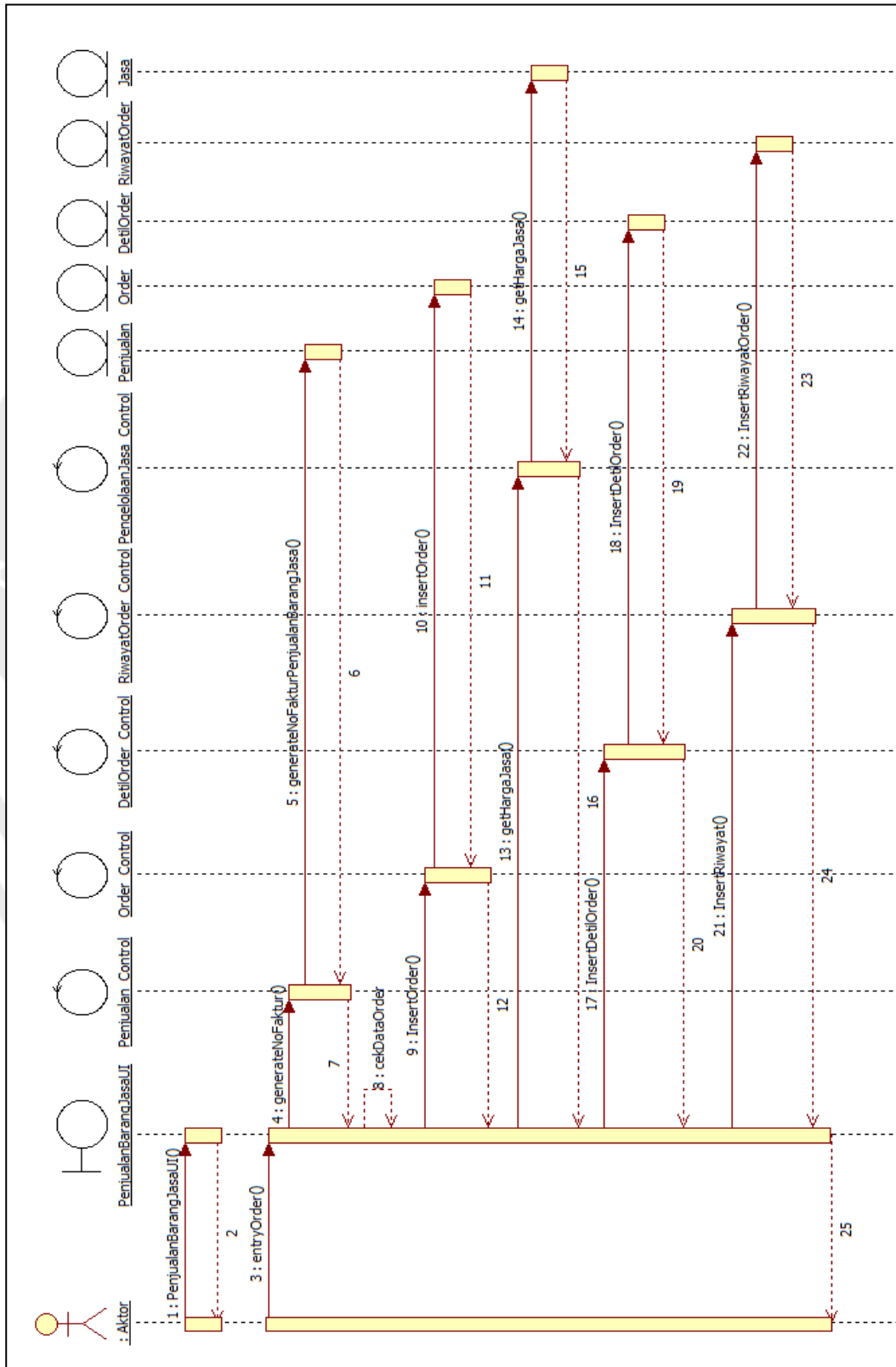
Gambar 2.16 Sequence Diagram : Transaksi Penjualan

2.2.1.11 Transaksi Retur Penjualan



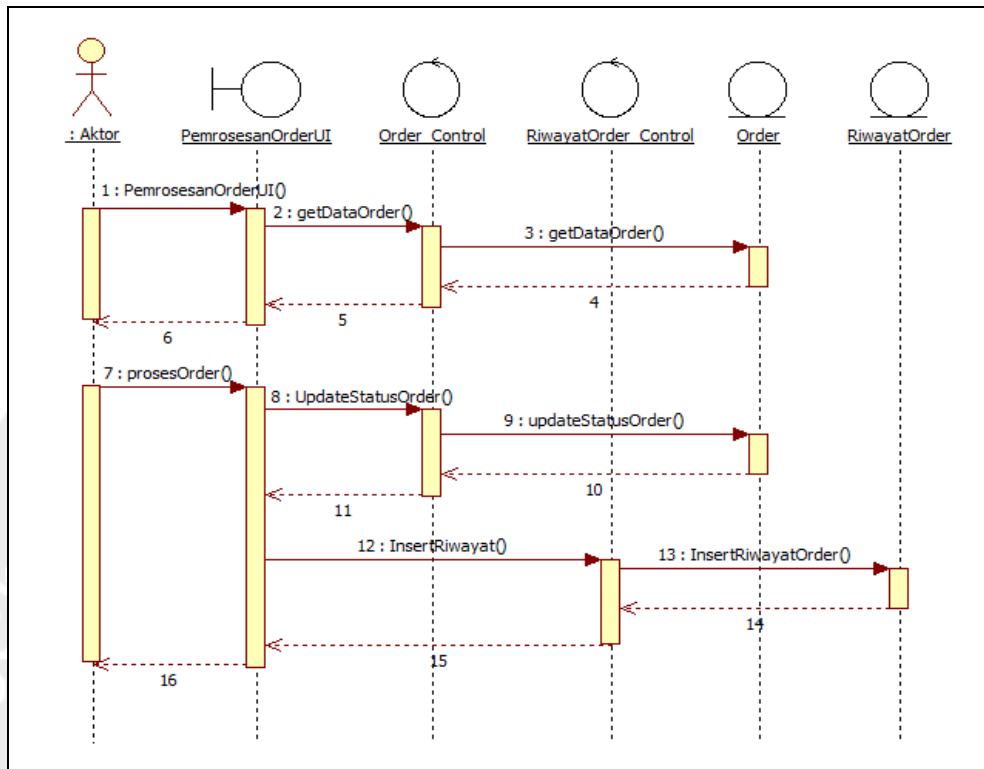
Gambar 2.17 Sequence Diagram : Transaksi Retur Penjualan

2.2.1.12 Transaksi Order



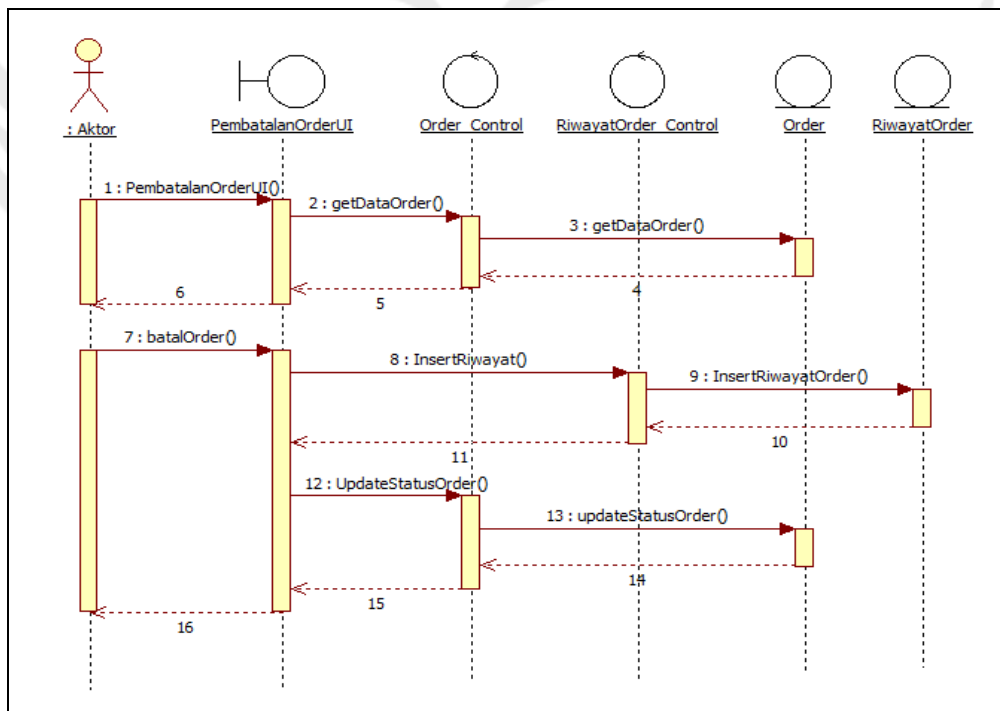
Gambar 2.18 Sequence Diagram : Transaksi Order

2.2.1.13 Transaksi Pemrosesan Order



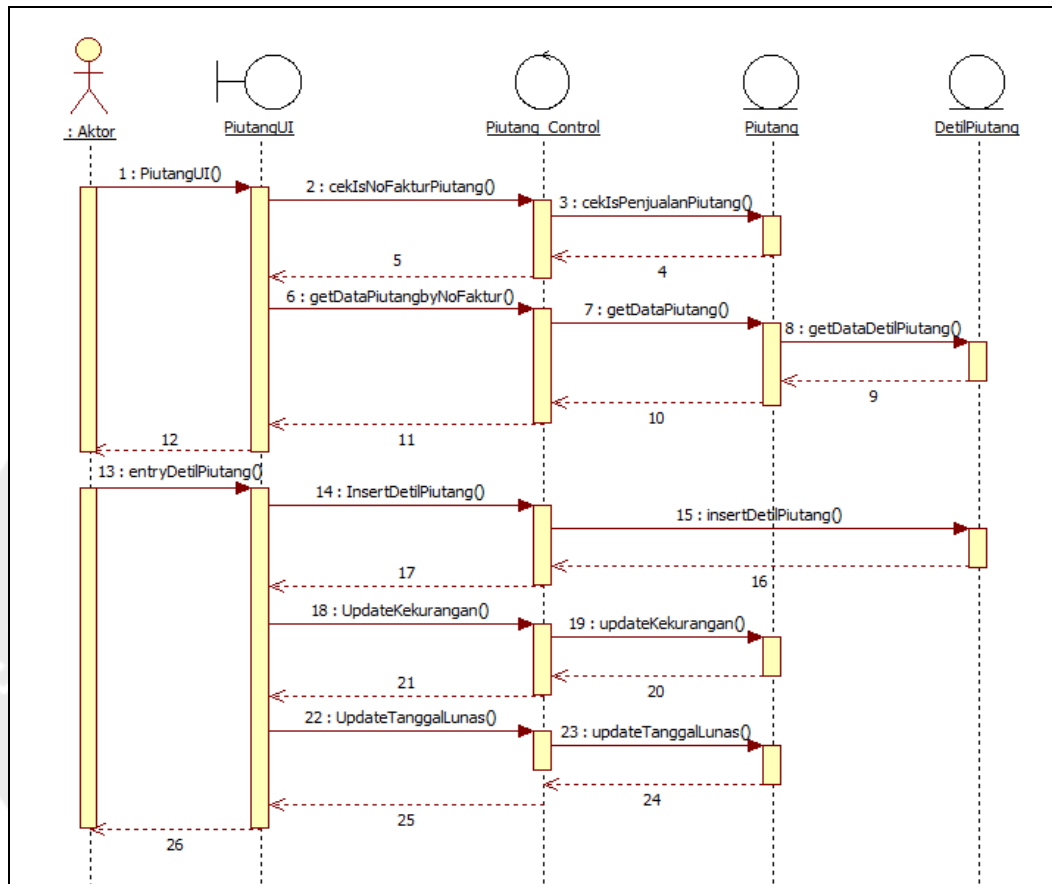
Gambar 2.19 Sequence Diagram : Transaksi Pemrosesan Order

2.2.1.14 Transaksi Pembatalan Order



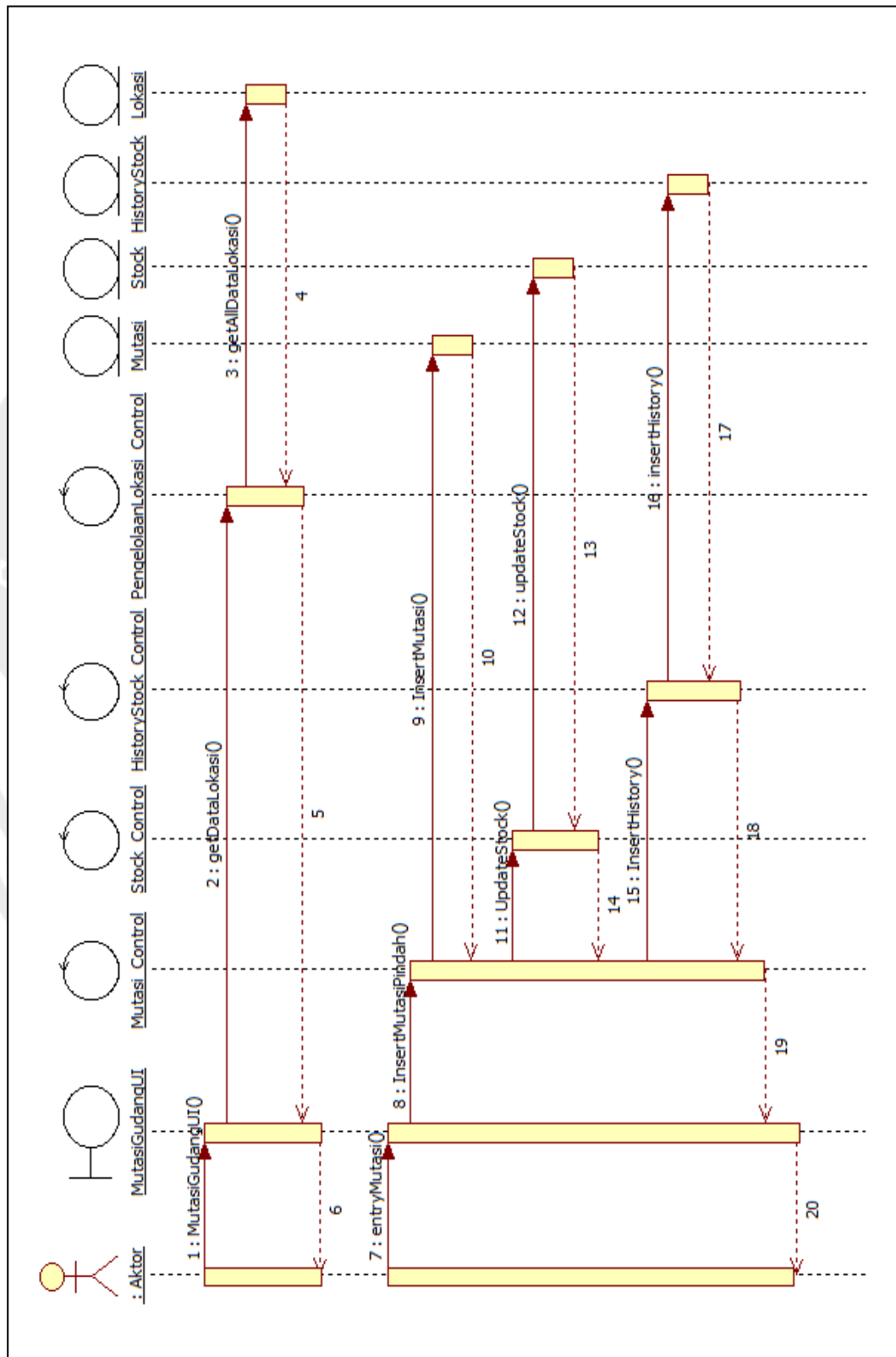
Gambar 2.20 Sequence Diagram : Transaksi Pembatalan Order

2.2.1.15 Transaksi Pembayaran Piutang



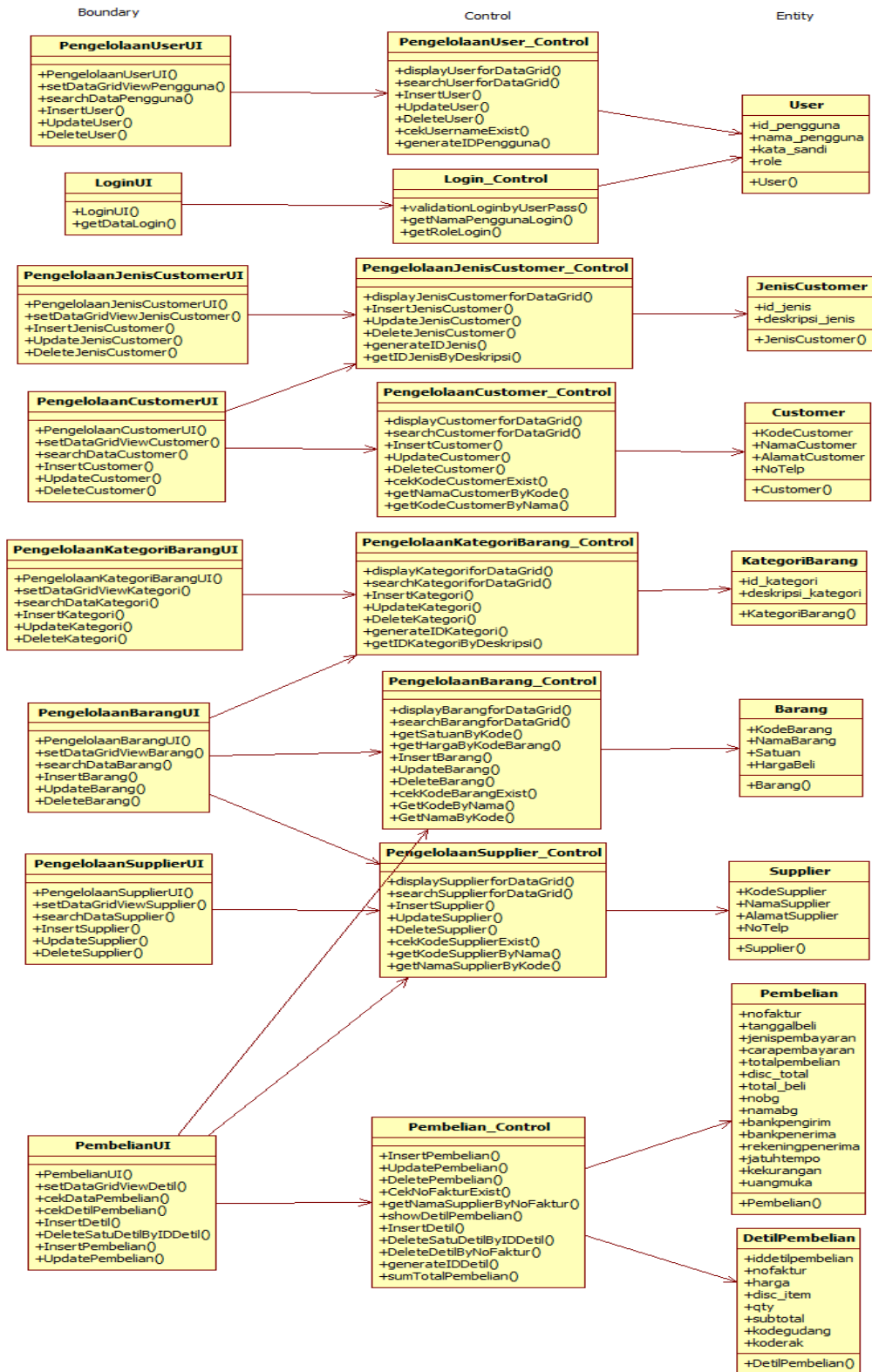
Gambar 2.21 Sequence Diagram : Transaksi Pembayaran Piutang

2.2.1.16 Transaksi Mutasi Gudang

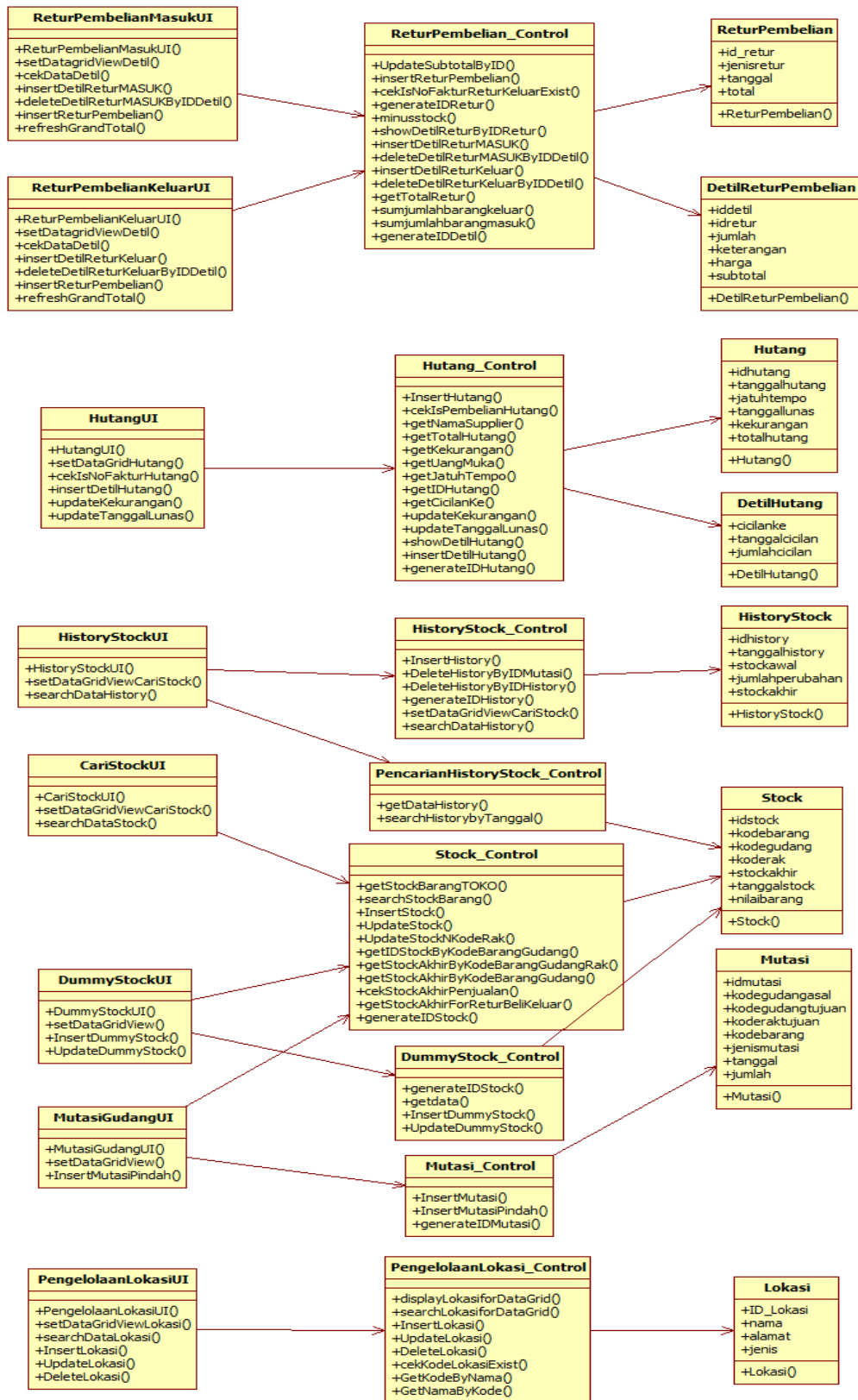


Gambar 2.22 Sequence Diagram : Transaksi Mutasi Gudang

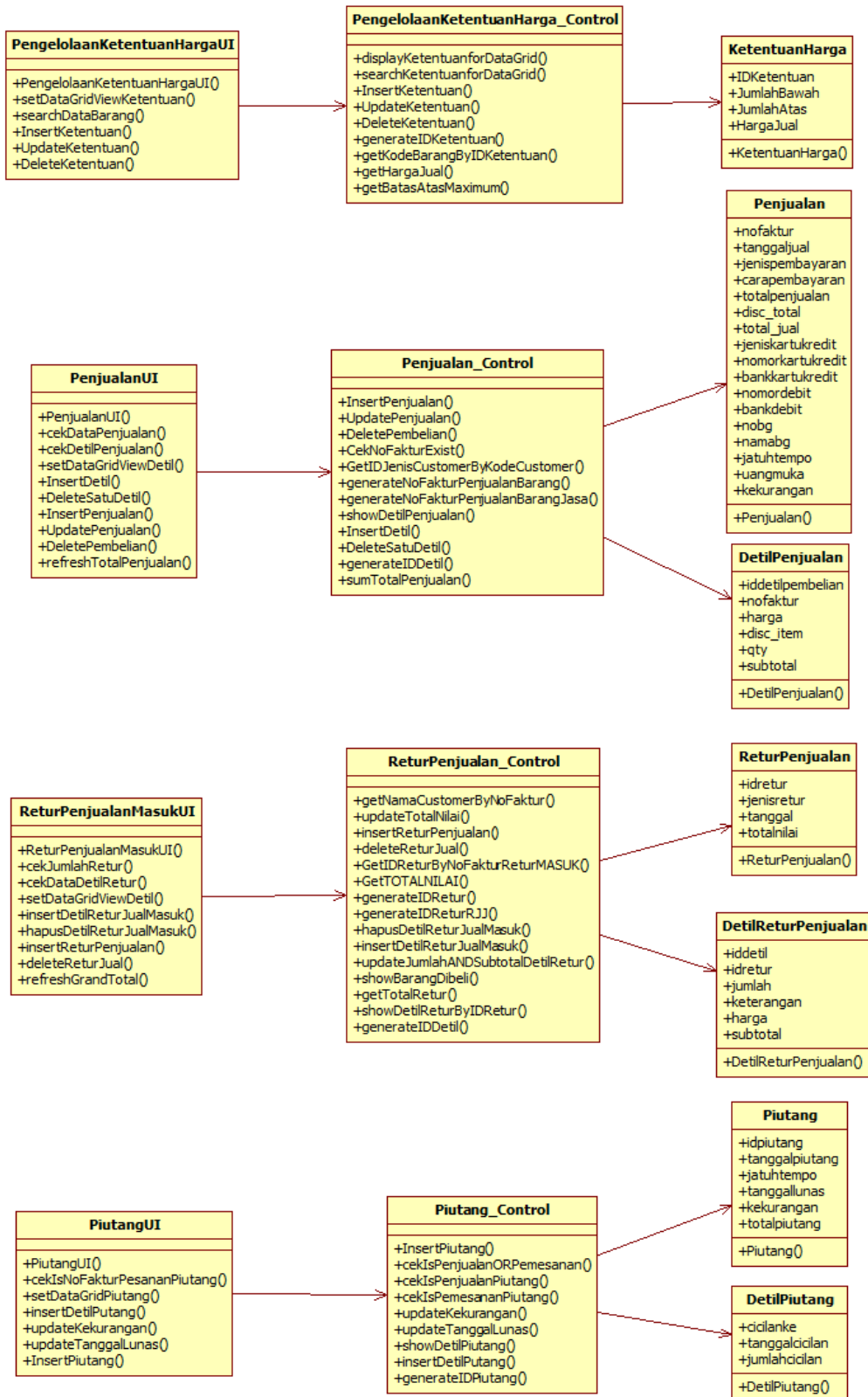
2.3 Class Diagram



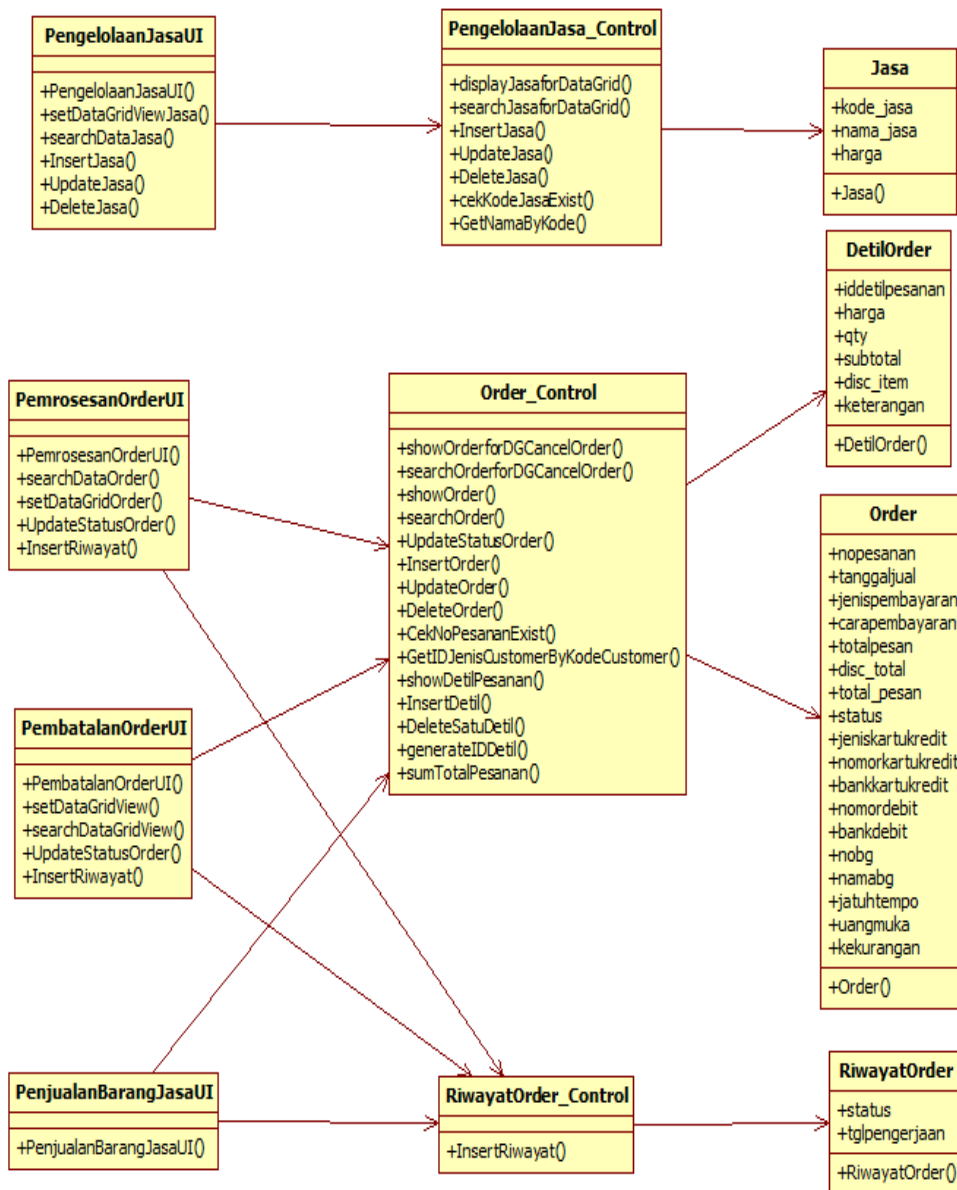
Gambar 2.23 Class Diagram



Gambar 2.24 Class Diagram (lanjutan 1)



Gambar 2.25 Class Diagram (lanjutan 2)



Gambar 2.26 Class Diagram (lanjutan 3)

2.3.1 Class Diagram Specific Description

2.3.1.1 Specific Design Class LoginUI

LoginUI	<<boundary>>
+LoginUI() Default konstruktor yang digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getDataLogin() Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu user name dan password.	

2.3.1.2 Specific Design Class PengelolaanUserUI

PengelolaanUserUI	<<boundary>>
+PengelolaanUserUI() Default konstruktor yang digunakan untuk inisialisasi semua attribute dari kelas ini.	
+setDataGridViewPengguna() Operasi ini digunakan untuk menampilkan data pengguna yang ada di database pada datagrid.	
+searchDataPengguna() Operasi ini digunakan untuk mencari data pengguna pada database berdasarkan kata kunci yang diinputkan.	
+InsertUser() Operasi ini digunakan untuk menambahkan data user baru ke dalam database.	
+UpdateUser() Operasi ini digunakan untuk mengubah data user yang sudah ada dan disimpan ke database.	
+DeleteUser() Operasi ini untuk menghapus data user pada database.	

2.3.1.3 Specific Design Class

PengelolaanJenisCustomerUI

PengelolaanJenisCustomerUI	<<boundary>>
<p>+PengelolaanJenisCustomerUI() Default konstruktor yang digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+setDataGridViewJenisCustomer() Operasi ini digunakan untuk menampilkan data jenis customer yang ada di database pada datagrid.</p> <p>+InsertJenisCustomer () Operasi untuk menambah data jenis customer.</p> <p>+UpdateJenisCustomer() Operasi digunakan untuk mengedit data jenis customer yang sudah tersimpan di database.</p> <p>+DeleteJenisCustomer() Operasi digunakan untuk menghapus data jenis customer dari database.</p>	

2.3.1.4 Specific Design Class PengelolaanCustomerUI

PengelolaanCustomerUI	<<boundary>>
<p>+PengelolaanCustomerUI() Default konstruktor yang digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+setDataGridViewCustomer() Operasi ini digunakan untuk menampilkan data customer yang ada di database pada datagrid.</p> <p>+searchDataCustomer() Operasi ini digunakan untuk mencari data customer pada database berdasarkan kata kunci yang diinputkan.</p>	

```
+InsertCustomer()
Operasi untuk menambah data customer.
+UpdateCustomer()
Operasi digunakan untuk mengedit data customer.
+DeleteCustomer()
Operasi ini digunakan untuk menghapus data customer
dari database.
```

2.3.1.5 Specific Design Class PengelolaanKategoriBarangUI

PengelolaanKategoriBarangUI	<<boundary>>
<pre>+PengelolaanKategoriBarangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +setDataGridViewKategori() Operasi ini digunakan untuk menampilkan data kategori barang yang ada di database pada datagrid. +searchDataKategori() Operasi ini digunakan untuk mencari data kategori barang pada database berdasarkan kata kunci yang diinputkan. +InsertKategori() Operasi ini digunakan untuk menambahkan data kategori. +DeleteKategori() Operasi ini digunakan untuk menghapus data kategori dari database. +UpdateKategori() Operasi digunakan untuk mengedit data kategori yang ada di database.</pre>	

2.3.1.6 Specific Design Class PengelolaanBarangUI

PengelolaanBarangUI	<<boundary>>
<p>+PengelolaanBarangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+InsertBarang() Operasi ini digunakan untuk menambahkan data barang.</p> <p>+DeleteBarang() Operasi ini digunakan untuk menghapus data barang dari database.</p> <p>+UpdateBarang() Operasi digunakan untuk mengedit data barang yang ada di database.</p>	

2.3.1.7 Specific Design Class PengelolaanSupplierUI

PengelolaanSupplierUI	<<boundary>>
<p>+PengelolaanSupplierUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+setDataGridViewSupplier() Operasi ini digunakan untuk menampilkan data supplier yang ada di database pada datagrid.</p> <p>+searchDataSupplier() Operasi ini digunakan untuk mencari data supplier pada Insert berdasarkan kata kunci yang diinputkan.</p> <p>+InsertSupplier() Operasi ini digunakan untuk menambahkan data supplier.</p> <p>+DeleteSupplier() Operasi ini digunakan untuk menghapus data supplier</p>	

dari database.

+UpdateSupplier()

Operasi digunakan untuk mengedit data supplier yang ada di database.

2.3.1.8 Specific Design Class PembelianUI

PembelianUI	<<boundary>>
<p>+PembelianUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+setDataGridViewDetil()</p> <p>Operasi untuk menampilkan data detil pembelian pada datagrid.</p> <p>+cekDataPembelian() : bool</p> <p>Operasi untuk mengecek input data pembelian. Jika data pembelian yang diinput tidak ada yang salah, maka akan mereturnkan nilai true.</p> <p>+cekDetilPembelian(): bool</p> <p>Operasi untuk mengecek input data detil pembelian. Jika data detil pembelian yang diinput tidak ada yang salah, maka akan mereturnkan nilai true.</p> <p>+InsertDetil()</p> <p>Operasi ini digunakan untuk menambahkan data detil pembelian.</p> <p>+DeleteSatuDetilByIDDetil()</p> <p>Operasi ini digunakan untuk menghapus satu detil pembelian berdasarkan ID detil.</p> <p>+InsertPembelian()</p> <p>Operasi ini digunakan untuk menambahkan data pembelian.</p> <p>+UpdatePembelian()</p> <p>Operasi ini digunakan untuk memperbarui data pembelian.</p>	

2.3.1.9 Specific Design Class ReturPembelianMasukUI

ReturPembelianMasukUI	<<boundary>>
<pre> +ReturPembelianMasukUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +setDataGridViewDetil() Operasi untuk menampilkan semua data detil retur pembelian ke datagrid. +cekDataDetil() Operasi untuk mengecek inputan data detil retur pembelian. +insertDetilReturMASUK() Operasi untuk memasukkan data detil retur pembelian masuk ke database. +deleteDetilReturMASUKByIDDettil() Operasi untuk menghapus data detil retur pembelian masuk dari database. +insertReturPembelian() Operasi untuk menambahkan data retur pembelian. +refreshGrandTotal() Operasi untuk menghitung ulang total nilai retur pembelian masuk yang terjadi. </pre>	

2.3.1.10 Specific Design Class ReturPembelianKeluarUI

ReturPembelianKeluarUI	<<boundary>>
<pre> +ReturPembelianKeluarUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +setDataGridViewDetil() </pre>	

Operasi ini digunakan untuk menampilkan data detail retur pembelian keluar.

+cekDataDetil()

Operasi ini digunakan untuk mengecek data detail retur pembelian keluar sudah valid.

+insertDetilReturKeluar()

Operasi ini digunakan untuk menambahkan data detail retur pembelian yang mempunyai jenis retur keluar dari database.

+deleteDetilReturKeluarByIDDetil()

Operasi ini digunakan untuk menghapus data detail retur pembelian yang mempunyai jenis retur Keluar yang ada pada database, berdasarkan id detail.

+insertReturPembelian()

Operasi ini digunakan untuk menambahkan data retur pembelian pada database.

+refreshGrandTotal()

Operasi untuk menghitung ulang total nilai retur pembelian keluar yang terjadi.

2.3.1.11 Specific Design Class HutangUI

HutangUI	<<boundary>>
+HutangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+setDataGridHutang() Operasi ini digunakan untuk menampilkan data hutang.	
+cekIsNoFakturHutang() Operasi ini digunakan untuk mengecek suatu nomor faktur pembelian merupakan hutang.	

+insertDetilHutang()

Operasi ini digunakan untuk menambahkan data detil hutang pada database.

+updateKekurangan()

Operasi ini digunakan untuk mengubah data kekurangan dari suatu data piutang.

+updateTanggalLunas()

Operasi ini digunakan untuk mengubah data tanggal lunas dari suatu data piutang.

2.3.1.12 Specific Design Class CariStockUI

CariStockUI	<<boundary>>
+CariStockUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+setDataGridViewCariStock() Operasi ini digunakan untuk menampilkan semua data history stock.	
+searchDataHistory() Operasi ini digunakan untuk mencari data history stock tertentu dari database.	

2.3.1.13 Specific Design Class DummyStockUI

DummyStockUI	<<boundary>>
+DummyStockUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+setDataGridView() Operasi ini digunakan untuk menampilkan semua data	

```

stock.
+InsertDummyStock()
Operasi ini digunakan untuk menambahkan data stock pada
database.
+UpdateDummyStock()
Operasi ini digunakan untuk mengubah data stock pada
database.

```

2.3.1.14 Specific Design Class MutasiGudangUI

MutasiGudangUI	<<boundary>>
<pre> +MutasiGudangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +setDataGridView() Operasi ini digunakan untuk menampilkan semua data mutasi gudang. +InsertMutasiPindah() Operasi ini digunakan untuk menambahkan data mutasi yang berjenis Pindah pada database. </pre>	

2.3.1.15 Specific Design Class PengelolaanLokasiUI

PengelolaanLokasiUI	<<boundary>>
<pre> +PengelolaanLokasiUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +setDataGridViewLokasi() Operasi ini digunakan untuk menampilkan semua data lokasi. +searchLokasi() </pre>	

Operasi ini digunakan untuk mencari data lokasi tertentu pada database.

+InsertLokasi()

Operasi ini digunakan untuk menambahkan data lokasi.

+UpdateLokasi()

Operasi ini digunakan untuk mengubah data lokasi.

+DeleteLokasi()

Operasi ini digunakan untuk menghapus data lokasi.

2.3.1.16 Specific Design Class HistoryStockUI

HistoryStockUI	<<boundary>>
<p>+HistoryStockUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+setDataGridViewCariStock()</p> <p>Operasi ini digunakan untuk menampilkan semua data history stock.</p> <p>+searchDataHistory()</p> <p>Operasi ini digunakan untuk mencari data history stock tertentu dari database.</p>	

2.3.1.17 Specific Design Class PengelolaanKetentuanHargaUI

PengelolaanKetentuanHargaUI	<<boundary>>
<p>+PengelolaanKetentuanHargaUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+setDataGridViewKetentuan()</p> <p>Operasi ini digunakan untuk menampilkan semua data ketentuan harga.</p>	

+searchDataBarang()

Operasi ini digunakan untuk mencari data barang.

+InsertKetentuan()

Operasi ini digunakan untuk menambahkan data ketentuan harga pada database.

+UpdateKetentuan()

Operasi ini digunakan untuk mengubah data ketentuan harga di database.

+DeleteKetentuan()

Operasi ini digunakan untuk menghapus data ketentuan harga dari database.

2.3.1.18 Specific Design Class PenjualanUI

PenjualanUI	<<boundary>>
<p>+PenjualanUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+cekDataPenjualan()</p> <p>Operasi ini digunakan untuk mengecek apakah data penjualan yang diinput sudah valid.</p> <p>+cekDetilPenjualan()</p> <p>Operasi ini digunakan untuk mengecek apakah data detil penjualan yang diinput sudah valid.</p> <p>+setDataGridViewDetil()</p> <p>Operasi ini digunakan untuk menampilkan semua data detil penjualan.</p> <p>+InsertDetil()</p> <p>Operasi ini digunakan untuk menambahkan data etil penjualan pada database.</p> <p>+DeleteSatuDetil()</p>	

Operasi ini digunakan untuk menghapus data detail penjualan pada database.

+InsertPenjualan()

Operasi ini digunakan untuk menambahkan data penjualan pada database.

+UpdatePenjualan()

Operasi ini digunakan untuk mengubah data penjualan di database.

+refreshTotalPenjualan()

Operasi untuk menghitung ulang total nilai penjualan yang terjadi.

2.3.1.19 Specific Design Class ReturPenjualanMasukUI

ReturPenjualanMasukUI	<<boundary>>
<p>+ReturPenjualanMasukUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+cekJumlahRetur()</p> <p>Operasi ini digunakan untuk mengecek jumlah retur mencukupi atau tidak.</p> <p>+cekDataDetilRetur()</p> <p>Operasi ini digunakan untuk mengecek data detail retur penjualan sudah valid.</p> <p>+setDataGridViewDetil()</p> <p>Operasi ini digunakan untuk menampilkan data detail retur penjualan.</p> <p>+insertDetilReturJualMasuk()</p> <p>Operasi ini digunakan untuk menambahkan data detail retur penjualan yang mempunyai jenis retur Masuk database.</p>	

```

+hapusDetilReturJualMasuk()
Operasi ini digunakan untuk menghapus data detil retur
penjualan yang mempunyai jenis retur Masuk database.
+insertReturPenjualan()
Operasi ini digunakan untuk menambahkan data retur
penjualan pada database.
+deleteReturJual()
Operasi ini digunakan untuk menghapus data retur
penjualan dari database.
+refreashGrandTotal()
Operasi untuk menghitung ulang total nilai retur
penjualan masuk yang terjadi.

```

2.3.1.20 Specific Design Class PiutangUI

PiutangUI	<<boundary>>
<pre> +PiutangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +cekIsNoFakturPesananPiutang() Operasi ini digunakan untuk mengecek apakah suatu nomor faktur pemesanan merupakan piutang. +setDataGridPiutang() Operasi ini digunakan untuk menampilkan semua data piutang. +insertDetilPiutang() Operasi ini digunakan untuk menambahkan data detil piutang pada database. +updateKekurangan() Operasi ini digunakan untuk mengubah data kekurangan dari suatu data piutang. </pre>	

+updateTanggalLunas()

Operasi ini digunakan untuk mengubah data tanggal lunas dari suatu data piutang.

+InsertPiutang()

Operasi ini digunakan untuk menambahkan data piutang pada database.

2.3.1.21 Specific Design Class Piutang_Control

Piutang_Control	<<control>>
<p>+InsertPiutang() Operasi ini digunakan untuk menambahkan data piutang pada database.</p> <p>+cekIsPenjualanORPemesanan() Operasi ini digunakan untuk mengecek suatu nomor faktur apakah merupakan penjualan atau pemesanan.</p> <p>+cekIsPenjualanPiutang() Operasi ini digunakan untuk mengecek apakah suatu nomor faktur penjualan merupakan piutang.</p> <p>+cekIsPemesananPiutang() Operasi ini digunakan untuk mengecek apakah suatu nomor pemesanan merupakan piutang.</p> <p>+updateKekurangan() Operasi ini digunakan untuk mengubah data kekurangan dari suatu data piutang.</p> <p>+updateTanggalLunas() Operasi ini digunakan untuk mengubah data tanggal lunas dari suatu data piutang.</p> <p>+showDetilPiutang() Operasi ini digunakan untuk menampilkan semua data detil piutang.</p>	

+insertDetilPiutang()

Operasi ini digunakan untuk menambahkan data detil piutang pada database.

+generateIDPiutang()

Operasi digunakan untuk generate id piutang.

2.3.1.22 Specific Design Class ReturPenjualan_Control

ReturPenjualan_Control	<<control>>
-------------------------------	--------------------------------

+getNamaCustomerByNoFaktur()

Operasi ini digunakan untuk mendapatkan data nama customer berdasarkan suatu nomor faktur penjualan.

+updateTotalNilai()

Operasi ini digunakan untuk mengubah total nilai suatu data retur penjualan pada database.

+insertReturPenjualan()

Operasi ini digunakan untuk menambahkan data retur penjualan pada database.

+deleteReturPenjualan()

Operasi ini digunakan untuk menghapus data retur penjualan dari database.

+GetIDReturByNoFakturReturMASUK()

Operasi ini digunakan untuk mendapatkan id retur berdasarkan nomor faktur retur penjualan masuk.

+GetTOTALNILAI()

Operasi ini digunakan untuk mendapatkan total nilai retur.

+generateIDRetur()

Operasi ini digunakan untuk generate id retur penjualan barang.

+generateIDReturRJJ()

Operasi ini digunakan untuk generate id retur penjualan barang jasa.

+hapusDetilReturJualMasuk()

Operasi ini digunakan untuk menghapus data detil retur penjualan yang mempunyai jenis retur Masuk database.

+insertDetilReturJualMasuk()

Operasi ini digunakan untuk menambahkan data detil retur penjualan yang mempunyai jenis retur Masuk database.

+updateJumlahANDSubtotalDetilRetur()

Operasi ini digunakan untuk mengubah data jumlah dan subtotal dari suatu data detil retur.

+showBarangDibeli()

Operasi ini digunakan untuk menampilkan semua barang yang dibeli berdasarkan suatu nomor faktur penjualan.

+getTotalRetur()

Operasi ini digunakan untuk mendapatkan nilai total retur.

+showDetilReturByIDRetur()

Operasi ini digunakan untuk menampilkan semua data detil retur penjualan berdasarkan id retur.

+generateIDDetil()

Operasi ini digunakan untuk generate id detil retur penjualan.

2.3.1.23 Specific Design Class Penjualan_Control

Penjualan_Control	<<control>>
<p>+InsertPenjualan() Operasi digunakan untuk menambahkan data penjualan pada database.</p> <p>+UpdatePenjualan() Operasi untuk mengubah data penjualan di database.</p> <p>+CekNoFakturExist() Operasi ini digunakan untuk mengecek apakah nomor faktur penjualan ada pada database.</p> <p>+generateNoFakturPenjualanBarang() Operasi ini digunakan untuk generate nomor faktur penjualan barang.</p> <p>+generateNoFakturPenjualanBarangJasa() Operasi ini digunakan untuk generate nomor faktur penjualan barang jasa.</p> <p>+showDetilPenjualan() Operasi ini digunakan untuk menampilkan semua data detil penjualan.</p> <p>+InsertDetil() Operasi ini digunakan untuk menambahkan data detil penjualan pada database.</p> <p>+DeleteSatuDetil() Operasi ini digunakan untuk menghapus satu data detil penjualan dari database.</p> <p>+generateIDDetil() Operasi digunakan untuk generate id detil penjualan.</p> <p>+sumTotalPenjualan() Operasi ini digunakan untuk mendapatkan jumlah total penjualan dari satu nomor faktur.</p>	

2.3.1.24 Specific Design Class

PengelolaanKetentuanHarga_Control

PengelolaanKetentuanHarga_Control	<<control>>
<pre>+displayKetentuanforDataGrid() Operasi ini digunakan untuk menampilkan semua data ketentuan harga. +searchKetentuan() Operasi ini digunakan untuk mencari data ketentuan harga tertentu pada database. +InsertKetentuan() Operasi ini digunakan untuk menambahkan data ketentuan harga pada database. +UpdateKetentuan() Operasi ini digunakan untuk mengubah data ketentuan harga di database. +DeleteKetentuan() Operasi ini digunakan untuk menghapus data ketentuan harga dari database. +generateIDKetentuan() Operasi ini digunakan untuk generate id ketentuan. +getHargaJual() Operasi ini digunakan untuk mendapatkan harga jual barang berdasarkan id ketentuan.</pre>	

2.3.1.25 Specific Design Class HistoryStock_Control

HistoryStock_Control	<<control>>
<pre>+InsertHistory() Operasi ini digunakan untuk menambahkan data history stock pada database.</pre>	

+DeleteHistoryByIDMutasi()

Operasi ini digunakan untuk menghapus data history stock dari database berdasarkan id mutasi.

+DeleteHistoryByIDHistory()

Operasi ini digunakan untuk menghapus data history stock dari database berdasarkan id history.

+generateIDHistory()

Operasi ini digunakan untuk generate id history.

2.3.1.26 Specific Design Class

PengelolaanLokasi_Control

PengelolaanLokasi_control	<<control>>
<p>+displayLokasiforDataGrid() Operasi ini digunakan untuk menampilkan semua data lokasi.</p> <p>+searchLokasi() Operasi ini digunakan untuk mencari data lokasi tertentu pada database.</p> <p>+InsertLokasi() Operasi ini digunakan untuk menambahkan data lokasi.</p> <p>+UpdateLokasi() Operasi ini digunakan untuk mengubah data lokasi.</p> <p>+DeleteLokasi() Operasi ini digunakan untuk menghapus data lokasi.</p> <p>+cekKodeLokasiExist() Operasi ini digunakan untuk mengecek apakah suatu koe lokasi ada pada database.</p> <p>+GetKodeByNama() Operasi ini digunakan untuk mendapatkan data kode lokasi berdasarkan nama lokasi.</p>	

+GetNamaByKode()

Operasi ini digunakan untuk mendapatkan data nama lokasi berdasarkan kode lokasi.

2.3.1.27 Specific Design Class Mutasi_Control

Mutasi_Control	<<control>>
+InsertMutasi() Operasi ini digunakan untuk menambahkan data mutasi pada database.	
+InsertMutasiPindah() Operasi ini digunakan untuk menambahkan data mutas yang berjenis Pindah pada database.	
+generateIDMutasi() Operasi ini digunakan untuk generate id mutasi.	

2.3.1.28 Specific Design Class DummyStock_Control

DummyStock_Control	<<control>>
+generateIDStock() Operasi ini digunakan untuk generate id stock.	
+getdata() Operasi ini digunakan untuk mendapatkan semua data stock dari database.	
+InsertDummyStock() Operasi ini digunakan untuk menambahkan data stock pada database.	
+UpdateDummyStock() Operasi ini digunakan untuk mengubah data stock pada database.	

2.3.1.29 Specific Design Class Stock_Control

Stock_Control	<<control>>
<pre>+getStockBarangTOKO() Operasi ini digunakan untuk mendapatkan semua data stock barang yang mempunyai lokasi barang di Toko. +searchStockBarang() Operasi ini digunakan untuk mencari data stock barang tertentu dari database. +InsertStock() Operasi ini digunakan untuk menambahkan data stock barang pada database. +UpdateStock() Operasi ini digunakan untuk mengubah data stock barang dari database. +UpdateStockNKodeRak() Operasi ini digunakan untuk mengubah data stock dan kode rak pada database. +generateIDStock() Operasi ini digunakan untuk generate id stock.</pre>	

2.3.1.30 Specific Design Class PencarianStock_Control

PencarianStock_Control	<<control>>
<pre>+getDataHistory() Operasi ini digunakan untuk mendapatkan semua data history stock dari database. +searchHistorybyTanggal() Operasi ini digunakan untuk mencari data history stock berdasarkan suatu tanggal.</pre>	

2.3.1.31 Specific Design Class Hutang_Control

Hutang_Control	<<control>>
<pre>+InsertHutang() Operasi ini digunakan untuk menambahkan data hutang pada database. +cekIsPembelianHutang() Operasi ini digunakan untuk mengecek apakah suatu data pembelian merupakan hutang. +updateKekurangan() Operasi ini digunakan untuk mengubah data kekurangan pada database. +updateTanggalLunas() Operasi ini digunakan untuk mengubah data tanggal lunas pada database. +showDetilHutang() Operasi ini digunakan untuk menampilkan semua data detil hutang dari database. +insertDetilHutang() Operasi ini digunakan untuk menambahkan data detil hutang pada database. +generateIDHutang() Operasi ini digunakan untuk generate id hutang.</pre>	

2.3.1.32 Specific Design Class ReturPembelian_Control

ReturPembelian_Control	<<control>>
<pre>+insertReturPembelian() Operasi ini digunakan untuk menambahkan data retur pembelian pada database. +cekIsNoFakturReturKeluarExist()</pre>	

Operasi ini digunakan untuk mengecek apakah suatu nomor faktur pembelian tertentu yang di retur keluar ada pada database.

+generateIDRetur()

Operasi ini digunakan untuk generate id retur pembelian.

+minusstock()

Operasi yang mengembalikan nilai minus dari suatu masukan angka.

+showDetilReturByIDRetur()

Operasi ini digunakan untuk menampilkan semua data detil retur pembelian berdasarkan id retur tertentu.

+insertDetilReturMASUK()

Operasi ini digunakan untuk menambahkan data detil retur pembelian yang mempunyai jenis retur Masuk dari database.

+deleteDetilReturMASUKByIDDetil()

Operasi ini digunakan untuk menghapus data detil retur pembelian yang mempunyai jenis retur Masuk yang ada pada database, berdasarkan id detil.

+insertDetilReturKeluar()

Operasi ini digunakan untuk menambahkan data detil retur pembelian yang mempunyai jenis retur keluar dari database.

+deleteDetilReturKeluarByIDDetil()

Operasi ini digunakan untuk menghapus data detil retur pembelian yang mempunyai jenis retur Keluar yang ada pada database, berdasarkan id detil.

+getTotalRetur()

Operasi ini digunakan untuk mendapatkan total nilai dari detil retur pembelian..

+generateIDDetil()

Operasi ini digunakan untuk generate id detil retur pembelian.

2.3.1.33 Specific Design Class Pembelian_Control

Pembelian_Control	<<control>>
<p>+InsertPembelian() Operasi ini digunakan untuk menambahkan data pembelian pada database.</p> <p>+UpdatePembelian() Operasi ini digunakan untuk mengubah data pembelian pada database.</p> <p>+DeletePembelian() Operasi ini digunakan untuk menghapus data pembelian dari database.</p> <p>+CekNoFakturExist() Operasi ini digunakan untuk mengecek apakah suatu data nomor faktur pembelian ada database.</p> <p>+showDetilPembelian() Operasi ini digunakan untuk menampilkan data detil pembelian.</p> <p>+InsertDetil() Operasi ini digunakan untuk menambahkan data detil pembelian pada database.</p> <p>+DeleteSatuDetilByIDDetil() Operasi ini digunakan untuk menghapus data detil pembelian pada database berdasarkan id detil pembelian tertentu.</p> <p>+DeleteDetilByNoFaktur() Operasi ini digunakan untuk menghapus data detil</p>	

pembelian pada database berdasarkan nomor faktur pembelian tertentu.

```
+generateIDDetil()
```

Operasi ini digunakan untuk generate id detil pembelian.

2.3.1.34 Specific Design Class PengelolaanSupplier_Control

PengelolaanSupplier_Control	<<control>>
<pre>+displaySupplierforDG()</pre> <p>Operasi ini digunakan untuk menampilkan semua data supplier.</p> <pre>+searchSupplier()</pre> <p>Operasi ini digunakan untuk mencari data supplier tertentu pada database.</p> <pre>+InsertSupplier()</pre> <p>Operasi ini digunakan untuk menambahkan data supplier pada database.</p> <pre>+UpdateSupplier()</pre> <p>Operasi ini digunakan untuk mengubah data supplier yang ada di database.</p> <pre>+DeleteSupplier()</pre> <p>Operasi ini digunakan untuk menghapus data supplier dari database.</p> <pre>+cekKodeSupplierExist()</pre> <p>Operasi ini digunakan untuk mengecek apakah suatu kode customer ada pada database.</p>	

2.3.1.35 Specific Design Class
PengelolaanBarang_Control

PengelolaanBarang_Control	<<control>>
<pre>+displayBarangforDG() Operasi ini digunakan untuk menampilkan semua data barang.</pre>	
<pre>+searchBarang() Operasi ini digunakan untuk mencari data barang tertentu pada database.</pre>	
<pre>+InsertBarang() Operasi ini digunakan untuk menambahkan data barang pada database.</pre>	
<pre>+UpdateBarang() Operasi ini digunakan untuk mengubah data barang di database.</pre>	
<pre>+DeleteBarang() Operasi ini digunakan untuk menghapus data barang dari database.</pre>	
<pre>+cekKodeBarangExist() Operasi ini digunakan untuk mengecek apakah kode barang ada pada database.</pre>	

2.3.1.36 Specific Design Class
PengelolaanKategoriBarang_Control

PengelolaanKategoriBarang_Control	<<control>>
<pre>+displayKategoriforDG() Operasi ini digunakan untuk menampilkan semua data kategori barang.</pre>	
<pre>+searchKategori()</pre>	

Program Studi Teknik Informatika	DPPL – SITB	58/ 122
----------------------------------	-------------	---------

Operasi ini digunakan untuk mencari data kategori tertentu pada database.

+InsertKategori()

Operasi ini digunakan untuk menambah data kategori pada database.

+UpdateKategori()

Operasi ini digunakan untuk mengubah data kategori di database.

+DeleteKategori()

Operasi ini digunakan untuk menghapus data kategori barang dari database.

+generateIDKategori()

Operasi ini digunakan untuk generate id kategori barang secara increment.

2.3.1.37 Specific Design Class PengelolaanCustomer_Control

PengelolaanCustomer_Control	<<control>>
<p>+displayCustomerforDG() Operasi ini digunakan untuk menampilkan semua data customer.</p> <p>+searchCustomer() Operasi ini digunakan untuk mencari data customer tertentu.</p> <p>+InsertCustomer() Operasi ini digunakan untuk menambahkan data customer pada database.</p> <p>+UpdateCustomer() Operasi ini digunakan untuk mengubah data customer di database.</p>	

+DeleteCustomer()

Operasi ini digunakan untuk menghapus data customer dari database.

+cekKodeCustomerExist()

Operasi ini digunakan untuk mengecek apakah suatu kode customer ada pada database.

2.3.1.38 Specific Design Class

PengelolaanJenisCustomer_Control

PengelolaanJenisCustomer_Control	<<control>>
+displayJenisCustomerforDataGrid() Operasi ini digunakan untuk menampilkan semua data jenis customer.	
+InsertJenisCustomer() Operasi ini digunakan untuk menambahkan data jenis customer pada database.	
+UpdateJenisCustomer() Operasi ini digunakan untuk mengubah data jenis customer.	
+DeleteJenisCustomer() Operasi ini digunakan untuk menghapus data jenis customer.	
+generateIDJenis() Operasi ini digunakan untuk generate ID jenis secara increment dari jumlah data.	

2.3.1.39 Specific Design Class Login_Control

Login_Control	<<control>>
+validationLoginbyUserPass() Operasi ini digunakan untuk memvalidasi username dan	

password apakah ada pada database.

+getNamaPeggunaLogin()
Operasi ini digunakan untuk mendapatkan nama pengguna yang login.

+getRoleLogin()
Operasi ini digunakan untuk mendapatkan data role dari user yang login.

2.3.1.40 Specific Design Class PengelolaanUser_Control

PengelolaanUser_Control	<<control>>
<p>+displayUserforDG() Operasi ini digunakan untuk menampilkan semua data user pada database ke datagrid.</p> <p>+searchUser() Operasi ini digunakan untuk mencari data user di database.</p> <p>+InsertUser() Operasi ini digunakan untuk menambahkan data user pada database.</p> <p>+UpdateUser() Operasi ini digunakan untuk mengubah data user pada database.</p> <p>+DeleteUser() Operasi untuk menghapus data user dari database.</p> <p>+cekUsernameExist() Operasi ini digunakan untuk melakukan pengecekan apakah username ada dalam database.</p> <p>+generateIDPegguna() Operasi ini digunakan untuk generate ID pengguna dengan cara increment dari jumlah data.</p>	

2.3.1.41 Specific Design Class User

User	<<entity>>
<p>-id_pengguna : integer Atribut ini digunakan untuk menyimpan data id dari pengguna.</p> <p>-nama_pengguna : string Atribut ini digunakan untuk menyimpan data nama pengguna.</p> <p>-kata_sandi : string Atribut ini digunakan untuk menyimpan data kata sandi untuk memasuki sistem.</p> <p>-role : string Atribut ini digunakan untuk menyimpan data role atau peran suatu user.</p>	
<p>+User() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.42 Specific Design Class JenisCustomer

JenisCustomer	<<entity>>
<p>-id_jenis : integer Atribut ini digunakan untuk menyimpan data id dari jenis customer.</p> <p>-deskripsi_jenis : string Atribut ini digunakan untuk menyimpan data deksripsi jenis customer.</p>	
<p>+JenisCustomer() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.43 Specific Design Class Customer

Customer	<<entity>>
<p>-KodeCustomer : string Atribut ini digunakan untuk menyimpan data kode customer.</p> <p>-NamaCustomer : string Atribut ini digunakan untuk menyimpan data nama customer.</p> <p>-AlamatCustomer : string Atribut ini digunakan untuk menyimpan data alamat customer.</p> <p>-NoTelp : string Atribut ini digunakan untuk menyimpan data nomor telpon customer.</p> <p>-IDJenis : integer Atribut ini digunakan untuk menyimpan data id dari jenis customer.</p>	
<p>+Customer() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.44 Specific Design Class Barang

KategoriBarang	<<entity>>
<p>-id_kategori : integer Atribut ini digunakan untuk menyimpan data id dari kategori barang.</p> <p>-deksripsi_kategori : string Atribut ini digunakan untuk menyimpan data deskripsi kategori barang.</p>	
<p>+KategoriBarang()</p>	

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.45 Specific Design Class Barang

Barang	<<entity>>
<p>-KodeBarang : string Atribut ini digunakan untuk menyimpan data kode barang.</p> <p>-NamaBarang : string Atribut ini digunakan untuk menyimpan data nama barang.</p> <p>-Satuan : string Atribut ini digunakan untuk menyimpan data satuan.</p> <p>-HargaBeli : decimal Atribut ini digunakan untuk menyimpan data harga beli suatu barang.</p> <p>-IDKategori : integer Atribut ini digunakan untuk menyimpan data id kategori barang.</p> <p>-kodesupplier : string Atribut ini digunakan untuk menyimpan data kode supplier.</p>	
<p>+Barang() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.46 Specific Design Class Supplier

Supplier	<<entity>>
<p>-KodeSupplier : string Atribut ini digunakan untuk menyimpan data kode supplier.</p> <p>-NamaSupplier : string Atribut ini digunakan untuk menyimpan data nama supplier.</p>	

-AlamatSupplier : string

Atribut ini digunakan untuk menyimpan data alamat supplier.

-NoTelp : string

Atribut ini digunakan untuk menyimpan data nomor telpon supplier.

+Supplier()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.47 Specific Design Class Pembelian

Pembelian	<<entity>>
<p>-nofaktur : string Atribut ini digunakan untuk menyimpan data nomor faktur pembelian.</p> <p>-tanggalbeli : date time Atribut ini digunakan untuk menyimpan data tanggal transaksi pembelian.</p> <p>-kodesupplier : string Atribut ini digunakan untuk menyimpan data kode supplier.</p> <p>-jenispembayaran : string Atribut ini digunakan untuk menyimpan data jenis pembayaran suatu pembelian. Data jenis pembayaran hanya ada Cash dan Hutang.</p> <p>-carapembayaran : string Atribut ini digunakan untuk menyimpan data cara pembayaran suatu pembelian. Data cara pembayaran pembelian hanya ada Tunai, Giro, Transfer dan Hutang.</p> <p>-totalpembelian : decimal Atribut ini digunakan untuk menyimpan data total nilai</p>	

pembelian.

-disc_total : decimal

Atribut ini digunakan untuk menyimpan data diskon total yang diberikan atas suatu pembelian.

-total_beli : decimal

Atribut ini digunakan untuk menyimpan data total nilai beli setelah dikurangi diskon total.

-nobg : string

Atribut ini digunakan untuk menyimpan data nomor giro, bila pembelian dibayar dengan cara giro.

-namabg : string

Atribut ini digunakan untuk menyimpan data nama bank giro, bila pembelian dibayar dengan cara giro.

-bankpengirim : string

Atribut ini digunakan untuk menyimpan data bank pengirim transfer, bila pembelian dibayar dengan cara transfer.

-bankpenerima : string

Atribut ini digunakan untuk menyimpan data bank penerima transfer, bila pembelian dibayar dengan cara transfer.

-rekeningpenerima : string

Atribut ini digunakan untuk menyimpan data nomor rekening penerima transfer, bila pembelian dibayar dengan cara transfer.

-jatuhtempo : date time

Atribut ini digunakan untuk menyimpan data tanggal jatuh tempo pembayaran pembelian, bila pembelian dibayar dengan cara hutang.

-uangmuka : decimal

Atribut ini digunakan untuk menyimpan data uang muka

yang dibayarkan customer atas suatu pembelian, bila pembelian dibayar dengan cara hutang.

-kekurangan : decimal

Atribut ini digunakan untuk menyimpan data kekurangan yang harus dibayarkan atas suatu pembelian, bila pembelian dibayar dengan cara hutang.

+Pembelian()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.48 Specific Design Class DetilPembelian

DetilPembelian	<<entity>>
<p>-iddetilpembelian : integer Atribut ini digunakan untuk menyimpan data id dari detil pembelian.</p> <p>-nofaktur : string Atribut ini digunakan untuk menyimpan data nomor faktur pembelian.</p> <p>-kodebarang : string Atribut ini digunakan untuk menyimpan data kode barang.</p> <p>-harga : decimal Atribut ini digunakan untuk menyimpan data harga beli satuan setiap barang.</p> <p>-disc_item : decimal Atribut ini digunakan untuk menyimpan data diskon yang diberikan untuk setiap item.</p> <p>-qty : integer Atribut ini digunakan untuk menyimpan data quantity atau jumlah.</p> <p>-subtotal : decimal Atribut ini digunakan untuk menyimpan data subtotal</p>	

<p>nilai detil pembelian.</p> <p>-kodegudang : string</p> <p>Atribut ini digunakan untuk menyimpan data kode gudang setiap barang detil pembelian disimpan.</p> <p>-koderak : string</p> <p>Atribut ini digunakan untuk menyimpan data kode rak setiap barang detil pembelian disimpan.</p> <p>-idmutasi : integer</p> <p>Atribut ini digunakan untuk menyimpan data id mutasi.</p>
<p>+DetilPembelian()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>

2.3.1.49 Specific Design Class ReturPembelian

ReturPembelian	<<entity>>
<p>-id_retur : integer</p> <p>Atribut ini digunakan untuk menyimpan data id dari suatu retur pembelian.</p> <p>-nofaktur : string</p> <p>Atribut ini digunakan untuk menyimpan data nomor faktur pembelian yang dikenakan retur pembelian.</p> <p>-jenisretur : string</p> <p>Atribut ini digunakan untuk menyimpan data jenis retur.</p> <p>-tanggal : date time</p> <p>Atribut ini digunakan untuk menyimpan data tanggal terjadi retur pembelian.</p> <p>-total : decimal</p> <p>Atribut ini digunakan untuk menyimpan data total nilai suatu retur pembelian.</p>	
<p>+ReturPembelian()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua</p>	

attribute dari kelas ini.

2.3.1.50 Specific Design Class DetilReturPembelian

DetilReturPembelian	<<entity>>
<p>-idddetil : integer Atribut ini digunakan untuk menyimpan data id dari detil retur pembelian.</p> <p>-idretur : integer Atribut ini digunakan untuk menyimpan data id dari retur pembelian.</p> <p>-kodebarang : string Atribut ini digunakan untuk menyimpan data kode barang yang diretur.</p> <p>-kodesupplier : string Atribut ini digunakan untuk menyimpan data kode supplier barang yang akan diretur.</p> <p>-kodegudang : string Atribut ini digunakan untuk menyimpan data kode gudang asal barang yang diretur.</p> <p>-koderak : string Atribut ini digunakan untuk menyimpan data kode rak asal barang yang diretur.</p> <p>-jumlah : integer Atribut ini digunakan untuk menyimpan data jumlah barang yang diretur pembelian.</p> <p>-keterangan : string Atribut ini digunakan untuk menyimpan data keterangan.</p> <p>-harga : decimal Atribut ini digunakan untuk menyimpan data harga barang setiap detil retur pembelian.</p> <p>-subtotal : decimal</p>	

Atribut ini digunakan untuk menyimpan data subtotal setiap detil retur pembelian.

-idmutasi : integer

Atribut ini digunakan untuk menyimpan data id mutasi.

+DetilReturPembelian()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.51 Specific Design Class Hutang

Hutang	<<entity>>
-idhutang : integer	
Atribut ini digunakan untuk menyimpan data id dari hutang.	
-nofaktur : string	
Atribut ini digunakan untuk menyimpan data nomor faktur pembelian yang hutang.	
-tanggalhutang : date time	
Atribut ini digunakan untuk menyimpan data tanggal terjadinya hutang.	
-jatuhtempo : date time	
Atribut ini digunakan untuk menyimpan data tanggal jatuh tempo suatu hutang.	
-tanggallunas : date time	
Atribut ini digunakan untuk menyimpan data tanggal lunas suatu hutang.	
-kekurangan : decimal	
Atribut ini digunakan untuk menyimpan data kekurangan yang harus dibayar atas hutang.	
-totalhutang : decimal	
Atribut ini digunakan untuk menyimpan data total hutang.	

+Hutang()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.52 Specific Design Class DetilHutang

DetilHutang	<<entity>>
<p>-idhutang : integer Atribut ini digunakan untuk menyimpan data id dari hutang.</p> <p>-cicilanke : integer Atribut ini digunakan untuk menyimpan data urutan pembayaran cicilan hutang.</p> <p>-tanggalcicilan : date time Atribut ini digunakan untuk menyimpan data tanggal pembayaran cicilan hutang.</p> <p>-jumlahcicilan : decimal Atribut ini digunakan untuk menyimpan data jumlah cicilan yang dibayar atas hutang.</p>	
<p>+DetilHutang() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.53 Specific Design Class Stock

Stock	<<entity>>
<p>-idstock : integer Atribut ini digunakan untuk menyimpan data id dari data stock.</p> <p>-kodebarang : string Atribut ini digunakan untuk menyimpan data kode barang.</p> <p>-kodegudang : string Atribut ini digunakan untuk menyimpan data kode gudang.</p> <p>-koderak : string</p>	

Atribut ini digunakan untuk menyimpan data kode rak tempat barang.

-stockakhir : integer

Atribut ini digunakan untuk menyimpan data jumlah stock terakhir.

-tanggalstock : date time

Atribut ini digunakan untuk menyimpan data tanggal terakhir perubahan data stock.

-nilaibarang : decimal

Atribut ini digunakan untuk menyimpan data nilai stock yang didapat dari stock akhir dikali harga beli barang.

+Stock()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.54 Specific Design Class Mutasi

Mutasi	<<entity>>
<p>-idmutasi : integer Atribut ini digunakan untuk menyimpan data id mutasi.</p> <p>-kodegudangasal : string Atribut ini digunakan untuk menyimpan data kode gudang asal mutasi.</p> <p>-kodegudangtujuan : string Atribut ini digunakan untuk menyimpan data kode gudang tujuan mutasi.</p> <p>-koderaktujuan : string Atribut ini digunakan untuk menyimpan data kode rak tujuan mutasi.</p> <p>-kodebarang : string Atribut ini digunakan untuk menyimpan data kode barang.</p> <p>-jenismutasi : string</p>	

<p>Atribut digunakan untuk menyimpan data jenis mutasi.</p> <p>-tanggal : date time</p> <p>Atribut ini digunakan untuk menyimpan data tanggal transaksi mutasi.</p> <p>-jumlah : integer</p> <p>Atribut ini digunakan untuk menyimpan data jumlah barang yang dimutasi.</p>
<p>+Mutasi()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>

2.3.1.55 Specific Design Class Lokasi

Lokasi	<<entity>>
<p>-ID_lokasi : string</p> <p>Atribut ini digunakan untuk menyimpan data id dari lokasi.</p> <p>-nama : string</p> <p>Atribut ini digunakan untuk menyimpan data nama lokasi.</p> <p>-alamat : string</p> <p>Atribut ini digunakan untuk menyimpan data alamat suatu lokasi.</p> <p>-jenis : string</p> <p>Atribut ini digunakan untuk menyimpan data jenis suatu lokasi, data hanya ada TOKO atau GUDANG.</p>	
<p>+Lokasi()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.56 Specific Design Class HistoryStock

HistoryStock	<<entity>>
<p>-idhistory : integer</p>	

<p>Atribut ini digunakan untuk menyimpan data id dari history stock.</p> <p>-idstock : integer</p> <p>Atribut ini digunakan untuk menyimpan data id stock.</p> <p>-tanggalhistory : date time</p> <p>Atribut ini digunakan untuk menyimpan data tanggal pencatatan history stock.</p> <p>-stockawal : integer</p> <p>Atribut ini digunakan untuk menyimpan data jumlah stock awal.</p> <p>-jumlahperubahan : integer</p> <p>Atribut ini digunakan untuk menyimpan data jumlah perubahan yang terjadi atas suatu stock.</p> <p>-stockakhir : integer</p> <p>Atribut ini digunakan untuk menyimpan data stock akhir.</p> <p>-idmutasi : integer</p> <p>Atribut ini digunakan untuk menyimpan data id mutasi.</p>
<p>+HistoryStock()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>

2.3.1.57 Specific Design Class KetentuanHarga

KetentuanHarga	<<entity>>
<p>-IDKetentuan : integer</p> <p>Atribut ini digunakan untuk menyimpan data id ketentuan harga.</p> <p>-kodeBarang : string</p> <p>Atribut ini digunakan untuk menyimpan data kode barang.</p> <p>-IDJenisCustomer : integer</p> <p>Atribut ini digunakan untuk menyimpan data id jenis customer.</p>	

-JumlahBawah : integer
 Atribut ini digunakan untuk menyimpan data jumlah minimum barang suatu ketentuan harga.

-JumlahAtas : integer
 Atribut ini digunakan untuk menyimpan data jumlah maksimum barang suatu ketentuan harga.

-HargaJual : decimal
 Atribut ini digunakan untuk menyimpan data harga jual.

+KetentuanHarga()
 Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.58 Specific Design Class Penjualan

Penjualan	<<entity>>
<p>-nofaktur : string Atribut ini digunakan untuk menyimpan data nomor faktur penjualan.</p> <p>-tanggaljual : date time Atribut ini digunakan untuk menyimpan data tanggal transaksi penjualan.</p> <p>-kodecustomer : string Atribut ini digunakan untuk menyimpan data kode customer yang melakukan transaksi penjualan.</p> <p>-jenispembayaran : string Atribut ini digunakan untuk menyimpan data jenis pembayaran suatu penjualan. Data jenis pembayaran hanya ada Cash dan Piutang.</p> <p>-carapembayaran : string Atribut ini digunakan untuk menyimpan data cara pembayaran suatu penjualan. Data cara pembayaran order hanya ada Tunai, Kartu Kredit, Kartu Debit, Giro dan</p>	

Piutang.

-totalpenjualan : decimal

Atribut ini digunakan untuk menyimpan data total nilai penjualan.

-disc_total : decimal

Atribut ini digunakan untuk menyimpan data diskon total yang diberikan atas suatu penjualan.

-total_jual : decimal

Atribut ini digunakan untuk menyimpan data total nilai penjualan setelah dikurangi diskon total.

-jeniskk : string

Atribut ini digunakan untuk menyimpan data jenis kartu kredit, bila penjualan dibayar dengan menggunakan kartu kredit.

-nomorkk : string

Atribut ini digunakan untuk menyimpan data nomor kartu kredit, bila penjualan dibayar dengan menggunakan kartu kredit.

-bankkk : string

Atribut ini digunakan untuk menyimpan data nama bank penerbit kartu kredit, bila penjualan dibayar dengan menggunakan kartu kredit.

-nomordebit : string

Atribut ini digunakan untuk menyimpan data nomor kartu debit, bila penjualan dibayar dengan cara menggunakan kartu debit.

-bankdebit : string

Atribut ini digunakan untuk menyimpan data nama bank penerbit kartu debit, bila penjualan dibayar dengan cara menggunakan kartu debit.

-nomorbg : string

Atribut ini digunakan untuk menyimpan data nomor giro, bila penjualan dibayar dengan cara giro.

-namabg : string

Atribut ini digunakan untuk menyimpan data nama bank giro, bila penjualan dibayar dengan cara giro.

-jatuhtempo : date time

Atribut ini digunakan untuk menyimpan data tanggal jatuh tempo pembayaran penjualan, bila penjualan dibayar dengan cara piutang.

-uangmuka : decimal

Atribut ini digunakan untuk menyimpan data uang muka yang dibayarkan customer atas suatu penjualan, bila penjualan dibayar dengan cara piutang.

-kekurangan : decimal

Atribut ini digunakan untuk menyimpan data kekurangan yang harus dibayarkan atas suatu penjualan, bila penjualan dibayar dengan cara piutang.

+Penjualan()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.59 Specific Design Class DetilPenjualan

DetilPenjualan	<<entity>>
<p>-idddetilpenjualan : integer</p> <p>Atribut ini digunakan untuk menyimpan data id dari setiap item detil penjualan.</p> <p>-nofaktur : string</p> <p>Atribut ini digunakan untuk menyimpan data nomor faktur penjualan.</p> <p>-kodebarang : string</p> <p>Atribut ini digunakan untuk menyimpan data kode barang.</p>	

-harga : decimal

Atribut ini digunakan untuk menyimpan data harga jual satuan setiap barang.

-disc_item : decimal

Atribut ini digunakan untuk menyimpan data diskon yang diberikan untuk setiap item barang detail penjualan.

-qty : integer

Atribut ini digunakan untuk menyimpan data jumlah/ quantity.

-subtotal : decimal

Atribut ini digunakan untuk menyimpan data subtotal nilai setiap detail penjualan.

+DetailPenjualan()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.60 Specific Design Class ReturPenjualan

ReturPenjualan	<<entity>>
<p>-idretur : string Atribut ini digunakan untuk menyimpan data id dari retur penjualan.</p> <p>-nofaktur : string Atribut ini digunakan untuk menyimpan data nomor faktur penjualan yang terjadi retur penjualan.</p> <p>-jenisretur : string Atribut ini digunakan untuk menyimpan data jenis retur penjualan.</p> <p>-tanggal : date time Atribut ini digunakan untuk menyimpan data tanggal transaksi retur penjualan.</p> <p>-totalnilai : decimal</p>	

Atribut ini digunakan untuk menyimpan data total nilai dari suatu transaksi retur penjualan.

+ReturPenjualan()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.61 Specific Design Class DetilReturPenjualan

DetilReturPenjualan	<<entity>>
----------------------------	-------------------------------

-idddetil : integer

Atribut ini digunakan untuk menyimpan data id dari setiap detil retur penjualan.

-idretur : string

Atribut ini digunakan untuk menyimpan data id dari retur penjualan.

-kodebarang : string

Atribut ini digunakan untuk menyimpan data kode barang yang diretur.

-jumlah : integer

Atribut ini digunakan untuk menyimpan data jumlah atau quantity item barang yang diretur.

-keterangan : string

Atribut ini digunakan untuk menyimpan data keterangan tambahan.

-harga : decimal

Atribut ini digunakan untuk menyimpan data harga setiap item barang detil retur penjualan.

-subtotal : decimal

Atribut ini digunakan untuk menyimpan data subtotal nilai dari setiap item detil retur penjualan.

+DetilReturPenjualan()

Default konstruktor, digunakan untuk inisialisasi semua

attribute dari kelas ini.

2.3.1.62 Specific Design Class Piutang

Piutang	<<entity>>
<p>-idpiutang : integer Atribut ini digunakan untuk menyimpan data id dari piutang.</p> <p>-nofaktur : string Atribut ini digunakan untuk menyimpan data nomor faktur yang ditransaksikan dengan cara piutang.</p> <p>-nopesanan : string Atribut ini digunakan untuk menyimpan data nomor pesanan yang ditransaksikan dengan cara piutang.</p> <p>-tanggalpiutang : date time Atribut ini digunakan untuk menyimpan data tanggal terjadinya piutang.</p> <p>-jatuhtempo : date time Atribut ini digunakan untuk menyimpan data tanggal jatuh tempo.</p> <p>-totalpiutang : decimal Atribut ini digunakan untuk menyimpan data total piutang.</p> <p>-kekurangan : decimal Atribut ini digunakan untuk menyimpan data kekurangan yang harus dibayarkan atas suatu piutang.</p> <p>-tanggallunas : date time Atribut ini digunakan untuk menyimpan data tanggal pelunasan piutang.</p>	
<p>+Piutang() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.63 Specific Design Class DetilPiutang

DetilPiutang	<<entity>>
<p>-idpiutang : integer Atribut ini digunakan untuk menyimpan data id dari piutang.</p> <p>-cicilanke : integer Atribut ini digunakan untuk menyimpan data urutan cicilan atas suatu piutang.</p> <p>-tanggalcicilan : date time Atribut ini digunakan untuk menyimpan data tanggal pembayaran cicilan.</p> <p>-jumlahcicilan : decimal Atribut ini digunakan untuk menyimpan data jumlah cicilan yang dibayarkan.</p>	
<p>+DetilPiutang() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.64 Specific Design Class Jasa

Jasa	<<entity>>
<p>-kode_jasa : string Atribut ini digunakan untuk menyimpan data kode unik yang diberikan untuk setiap jasa.</p> <p>-nama_jasa : string Atribut ini digunakan untuk menyimpan data nama jasa.</p> <p>-harga : decimal Atribut ini digunakan untuk menyimpan data harga jual atas suatu jasa.</p>	
<p>+Jasa() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.65 Specific Design Class DetilOrder

DetilOrder	<<entity>>
<p>-idddetilpesanan : integer Atribut ini digunakan untuk menyimpan data nomor id dari detil pesanan.</p> <p>-nopesanan : string Atribut ini digunakan untuk menyimpan data nomor pesanan.</p> <p>-kodejasa : string Atribut ini digunakan untuk menyimpan data kode jasa.</p> <p>-harga : decimal Atribut ini digunakan untuk menyimpan data harga untuk setiap satuan item order.</p> <p>-qty : integer Atribut ini digunakan untuk menyimpan data quantity atau jumlah untuk setiap item order.</p> <p>-subtotal : decimal Atribut ini digunakan untuk menyimpan data subtotal nilai dari setiap item order.</p> <p>-disc_item : decimal. Atribut ini digunakan untuk menyimpan data diskon yang diberikan untuk setiap item order.</p> <p>-keterangan : string Atribut ini digunakan untuk menyimpan data tambahan yang berupa keterangan. Data keterangan ini bersifat opsional.</p>	
<p>+DetilOrder() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

2.3.1.66 Specific Design Class Order

Order	<<entity>>
<p>-nopesanan : string Atribut ini digunakan untuk menyimpan data nomor pesanan yang unik untuk setiap order.</p> <p>-tanggaljual : date time Atribut ini digunakan untuk menyimpan data tanggal terjadinya transaksi order.</p> <p>-kodecustomer : string Atribut ini digunakan untuk menyimpan data kode customer yang melakukan transaksi order.</p> <p>-jenispembayaran : string Atribut ini digunakan untuk menyimpan data jenis pembayaran suatu order. Data jenis pembayaran hanya ada Cash dan Piutang.</p> <p>-carapembayaran : string Atribut ini digunakan untuk menyimpan data cara pembayaran suatu order. Data cara pembayaran order hanya ada Tunai, Kartu Kredit, Kartu Debit, Giro dan Piutang.</p> <p>-totalpesanan : decimal Atribut ini digunakan untuk menyimpan data total nilai pesanan.</p> <p>-disc_total : decimal Atribut ini digunakan untuk menyimpan data diskon total yang diberikan atas suatu order.</p> <p>-total_pesan : decimal Atribut ini digunakan untuk menyimpan data total nilai pesanan setelah dikurangi diskon total.</p> <p>-status : string Atribut ini digunakan untuk menyimpan data status</p>	

pengerjaan terakhir dari suatu order.

-jeniskartukredit : string

Atribut ini digunakan untuk menyimpan data jenis kredit, bila order dibayar dengan menggunakan kartu kredit.

-nomorkartukredit : string

Atribut ini digunakan untuk menyimpan data nomor kartu kredit, bila order dibayar dengan menggunakan kartu kredit.

-bankkartukredit : string

Atribut ini digunakan untuk menyimpan data nama bank penerbit kartu kredit, bila order dibayar dengan menggunakan kartu kredit.

-nomordebit : string

Atribut ini digunakan untuk menyimpan data nomor kartu debit, bila order dibayar dengan menggunakan kartu debit.

-bankdebit : string

Atribut ini digunakan untuk menyimpan data nama bank penerbit kartu debit, bila order dibayar dengan menggunakan kartu debit.

-nobg : string

Atribut ini digunakan untuk menyimpan data nomor giro, bila order dibayar dengan cara giro.

-namabg : string

Atribut ini digunakan untuk menyimpan data nama bank giro, bila order dibayar dengan cara giro.

-jatuh tempo : date time

Atribut ini digunakan untuk menyimpan data tanggal jatuh tempo pembayaran order, bila order dibayar dengan cara piutang.

-uangmuka : decimal

Atribut ini digunakan untuk menyimpan data uang muka yang dibayarkan customer atas suatu order, bila order dibayar dengan cara piutang.

-kekurangan : decimal

Atribut ini digunakan untuk menyimpan data kekurangan dari suatu order, bila order dibayar dengan cara piutang.

+Order()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

2.3.1.67 Specific Design Class RiwayatOrder

RiwayatOrder	<<entity>>
<p>-nomorpesanan : string</p> <p>Atribut ini digunakan untuk menyimpan data nomor order atau nomor pesanan.</p> <p>-status : string</p> <p>Atribut ini digunakan untuk menyimpan data status pengerjaan dari suatu order. Status hanya ada 3 tahap pengerjaan yaitu MULAI, PROSES, dan SELESAI.</p> <p>-tglpengerjaan : date time</p> <p>Atribut ini digunakan untuk menyimpan data tanggal pengerjaan status terakhir dari order.</p>	
<p>+RiwayatOrder()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

3 Perancangan Data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas TBL_USER

Nama	Tipe	Panjang	Keterangan
IDUSER	Integer	-	Nomor identitas unik dari setiap user, Primary Key
NAMAPENGGUNA	Variable character	20	Nama pengguna
KATASANDI	Variable character	20	Kata sandi pengguna
ROLE	Variable character	20	Role dari user

3.1.2 Deskripsi Entitas TBL_SUPPLIER

Nama	Tipe	Panjang	Keterangan
KODESUPPLIER	Variable character	5	Nomor identitas unik dari setiap supplier, Primary Key
NAMASUPPLIER	Variable character	50	Nama supplier
ALAMAT	Variable character	100	Alamat supplier
NOTELP	Variable character	100	Nomor telepon supplier

3.1.3 Deskripsi Entitas TBL_STOCK

Nama	Tipe	Panjang	Keterangan
IDSTOCK	Integer	-	Nomor identitas unik dari setiap data stock, Primary Key
KODEBARANG	Variable character	20	Kode barang, Foreign Key
KODEGUDANG	Variable character	6	Kode gudang, Foreign Key
KODERAK	Variable character	6	Kode rak
STOCKAKHIR	Integer	-	Jumlah stock barang terakhir
TANGGALSTOCK	DateTime	-	Tanggal stock barang
NILAIBARANG	Decimal	12	Nilai stock barang yang didapat dari perkalian STOCKAKHIR dan HargaBeliBarang

3.1.4 Deskripsi Entitas TBL_RIWAYAT_ORDER

Nama	Tipe	Panjang	Keterangan
NOMORPELANAN	Variable character	15	Nomor identitas unik dari setiap pesanan, Foreign Key
STATUS	Variable character	15	Status order
TGL_PENGERJAAN	DateTime	-	Tanggal pengerjaan tahapan order

3.1.5 Deskripsi Entitas TBL_RETUR_PENJUALAN

Nama	Tipe	Panjang	Keterangan
IDRETURPENJUALAN	Variable character	15	Nomor identitas unik dari setiap transaksi retur penjualan, Primary Key
NOFAKTURPENJUALAN	Variable character	15	No faktur penjualan dari suatu transaksi retur penjualan, Foreign Key
JENISRETUR	Variable character	20	Jenis retur penjualan
TANGGALRETUR	DateTime	-	Tanggal terjadi transaksi retur penjualan
TOTALNILAI	Decimal	12	Total nilai barang yang diretur

3.1.6 Deskripsi Entitas TBL_RETUR_PEMBELIAN

Nama	Tipe	Panjang	Keterangan
IDRETURPEMBELIAN	Integer	-	Nomor identitas unik dari setiap transaksi retur pembelian, Primary Key
NOFAKTURPEMBELIAN	Variable character	10	Nomor faktur pembelian atas transaksi retur pembelian
JENISRETUR	Variable character	20	Jenis retur pembelian
TANGGALRETUR	DateTime	-	Tanggal transaksi retur pembelian
TOTALNILAI	Decimal	12	Total nilai barang yang diretur

3.1.7 Deskripsi Entitas TBL_PIUTANG

Nama	Tipe	Panjang	Keterangan
IDPIUTANG	Integer	-	Nomor identitas unik dari setiap transaksi piutang, Primary Key
NOFAKTURPENJUALAN	Variable character	15	Nomor faktur penjualan atas piutang, Foreign Key
NOMORPESANAN	Variable character	15	Nomor pesanan atas piutang, Foreign Key
TANGGALPIUTANG	DateTime	-	Tanggal terjadinya piutang
JATUHTEMPO	DateTime	-	Tanggal jatuh tempo piutang
TOTALPIUTANG	Decimal	12	Total piutang
KEKURANGAN	Decimal	12	Jumlah kekurangan piutang
TANGGALLUNAS	DateTime	-	Tanggal pelunasan piutang

3.1.8 Deskripsi Entitas TBL_PENJUALAN

Nama	Tipe	Panjang	Keterangan
NOFAKTURPENJUALAN	Variable character	15	Nomor identitas unik dari setiap transaksi penjualan, Primary Key
TANGGALPENJUALAN	DateTime	-	Tanggal transaksi penjualan
KODECUSTOMER	Variable character	5	Kode customer, Foreign Key
JENISPEMBAYARAN	Variable character	10	Jenis pembayaran
CARAPEMBAYARAN	Variable character	10	Cara pembayaran
TOTALPENJUALAN	Decimal	12	Total bruto harga barang penjualan
DISC_TOTAL	Decimal	12	Diskon total atas suatu penjualan

TOTAL_JUAL	Decimal	12	Total netto harga penjualan setelah dikurangi DISC_TOTAL
JENISKK	Variable character	10	Jenis kartu kredit
NOMORKK	Variable character	20	Nomor kartu kredit
BANKKK	Variable character	50	Bank penerbit kartu kredit
NOMORDEBIT	Variable character	50	Nomor kartu debit
BANKDEBIT	Variable character	50	Bank penerbit kartu debit
NOMORBG	Variable character	50	Nomor bank giro
NAMABG	Variable character	50	Nama pemegang bank giro
JATUHTEMPO	DateTime	-	Tanggal jatuh tempo piutang
UANGMUKA	Decimal	12	Uang muka atas suatu piutang
KEKURANGAN	Decimal	20	Jumlah kekurangan atas piutang

3.1.9 Deskripsi Entitas TBL_PEMESANAN

Nama	Tipe	Panjang	Keterangan
NOMORPEMANAN	Variable character	15	Nomor identitas unik dari setiap transaksi pesanan, Primary Key
TANGGALPENJUALAN	DateTime	-	Tanggal transaksi
KODECUSTOMER	Variable character	5	Kode customer, Foreign Key
JENISPEMBAYARAN	Variable character	10	Jenis pembayaran
CARAPEMBAYARAN	Variable character	50	Cara pembayaran
TOTALPEMANAN	Decimal	12	Total bruto harga pesanan
DISC_TOTAL	Decimal	12	Diskon total atas suatu pesanan
TOTAL_PESANAN	Decimal	12	Total netto harga pesanan setelah dikurangi DISC_TOTAL
STATUSPEMANAN	Variable character	10	Status terakhir pengerjaan pesanan
JENISKK	Variable character	10	Jenis kartu kredit
NOMORKK	Variable character	20	Nomor kartu kredit
BANKKK	Variable character	50	Bank penerbit kartu kredit
NOMORDEBIT	Variable character	50	Nomor kartu debit
BANKDEBIT	Variable character	50	Bank penerbit kartu debit
NOMORBG	Variable character	50	Nomor bank giro
NAMABG	Variable character	50	Nama bank giro
JATUHTEMPO	DateTime	-	Tanggal jatuh tempo atas suatu piutang

UANGMUKA	Decimal	12	Uang muka atas piutang
KEKURANGAN	Decimal	12	Kekurangan atas piutang

3.1.10 Deskripsi Entitas TBL_PEMBELIAN

Nama	Tipe	Panjang	Keterangan
NOFAKTURPEMBELIAN	Variable character	10	Nomor identitas unik dari setiap transaksi pembelian, Primary Key
TANGGALPEMBELIAN	DateTime	-	Tanggal transaksi pembelian
KODESUPPLIER	Variable character	5	Kode supplier, Foreign Key
JENISPEMBAYARAN	Variable character	10	Jenis pembayaran
CARAPEMBAYARAN	Variable character	10	Cara pembayaran
TOTALPEMBELIAN	Decimal	12	Total bruto harga pembelian
DISC_TOTAL	Decimal	12	Diskon total atas pembelian
TOTAL_BELI	Decimal	12	Total netto harga pembelian setelah dikurangi DISC_TOTAL
NOMORBG	Variable character	50	Nomor bank giro
NAMABG	Variable character	50	Nama bank giro
BANKPENGIRIM	Variable character	50	Bank pengirim transfer
BANKPENERIMA	Variable character	50	Bank penerima transfer
REKENINGPENERIMA	Variable character	50	Nomor Rekening penerima transfer
JATUHTEMPO	DateTime	-	Tanggal jatuh tempo hutang
UANGMUKA	Decimal	12	Uangmuka hutang
KEKURANGAN	Decimal	12	Kekurangan atas hutang

3.1.11 Deskripsi Entitas TBL_MUTASI_GUDANG

Nama	Tipe	Panjang	Keterangan
IDMUTASI	Integer	-	Nomor identitas unik dari setiap transaksi mutasi, Primary Key
KODEGUDANGASAL	Variable character	6	Kode gudang asal mutasi
KODEGUDANGTUJUAN	Variable character	6	Kode gudang tujuan mutasi
KODERAKTUJUAN	Variable character	6	Kode rak tujuan mutasi
KODEBARANG	Variable character	20	Kode barang yang dimutasi, Foreign Key
JENISMUTASI	Variable character	25	Jenis mutasi
TANGGALMUTASI	DateTime	-	Tanggal mutasi
JUMLAH	Integer	-	Jumlah barang yang dimutasi

3.1.12 Deskripsi Entitas TBL_MASTER_JASA

Nama	Tipe	Panjang	Keterangan
KODEJASA	Variable character	10	Nomor identitas unik dari setiap jasa, Primary Key
NAMAJASA	Variable character	50	Nama jasa
HARGA	Decimal	12	Harga jual jasa

3.1.13 Deskripsi Entitas TBL_MASTER_BARANG

Nama	Tipe	Panjang	Keterangan
KODEBARANG	Variable character	20	Nomor identitas unik dari setiap barang, Primary Key
NAMABARANG	Variable character	50	Nama barang
HARGABELI	Decimal	12	Harga beli barang
IDKATEGORI	Integer	-	ID Kategori Barang, Foreign Key
KODESUPPLIER	Variable character	5	Kode Supplier, Foreign Key
SATUAN	Variable character	50	Satuan barang

3.1.14 Deskripsi Entitas TBL_LOKASI

Nama	Tipe	Panjang	Keterangan
KODEGUDANG	Variable character	6	Nomor identitas unik dari setiap lokasi, Primary Key
NAMAGUDANG	Variable character	30	Nama lokasi
ALAMAT	Variable character	100	Alamat lokasi
JENIS	Variable character	30	Jenis lokasi

3.1.15 Deskripsi Entitas TBL_KETENTUAN_HARGA

Nama	Tipe	Panjang	Keterangan
IDKETENTUAN	Integer	-	Nomor identitas unik dari setiap ketentuan harga, Primary Key
KODEBARANG	Variable character	20	Kode barang, Foreign Key
ID_JENIS	Integer	-	Id jenis customer, Foreign Key
JUMLAHBAWAH	Integer	-	Jumlah minimum
JUMLAHATAS	Integer	-	Jumlah maksimum
HARGA	Decimal	12	Harga jual barang

3.1.16 Deskripsi Entitas TBL_KATEGORI

Nama	Tipe	Panjang	Keterangan
IDKATEGORI	Integer	-	Nomor identitas unik dari setiap kategori barang, Primary Key
DESKRIPSIKATEGORI	Variable character	100	Deskripsi kategori barang

3.1.17 Deskripsi Entitas TBL_JENIS_CUSTOMER

Nama	Tipe	Panjang	Keterangan
ID_JENIS	Integer	-	Nomor identitas unik dari setiap jenis customer, Primary Key
DESKRIPSI_JENIS	Variable character	30	Deskripsi jenis customer

3.1.18 Deskripsi Entitas TBL_HUTANG

Nama	Tipe	Panjang	Keterangan
IDHUTANG	Integer	-	Nomor identitas unik dari setiap transaksi hutang, Primary Key
NOFAKTURPEMBELIAN	Variable character	10	Nomor faktur pembelian atas suatu transaksi hutang, Foreign Key
TANGGALHUTANG	DateTime	-	Tanggal transaksi
JATUHTEMPO	DateTime	-	Tanggal jatuh tempo hutang
TOTALHUTANG	Decimal	12	Total hutang
KEKURANGAN	Decimal	12	Kekurangan hutang
TANGGALLUNAS	DateTime	-	Tanggal pelunasan hutang

3.1.19 Deskripsi Entitas TBL_HISTORY_STOCK

Nama	Tipe	Panjang	Keterangan
IDHISTORYSTOCK	Integer	-	Nomor identitas unik dari setiap history stock, Primary Key
IDSTOCK	Integer	-	Id stock, Foreign Key
TANGGALHISTORY	DateTime	-	Tanggal pencatatan history stock
STOCKAWAL	Integer	-	Jumlah stock awal
JUMLAH_STOCK	Integer	-	Jumlah perubahan stock
STOCKAKHIR	Integer	-	Jumlah stock akhir
IDMUTASI	Integer	-	Id mutasi, Foreign Key

3.1.20 Deskripsi Entitas TBL_DETIL_RETUR_PENJUALAN

Nama	Tipe	Panjang	Keterangan
IDDETILRETURPENJUALAN	Integer	-	Nomor identitas unik dari setiap transaksi detil retur penjualan, Primary Key
IDRETURPENJUALAN	Variable character	15	Nomor identitas id retur penjualan, Foreign Key
KODEBARANG	Variable character	20	Kode barang, Foreign Key
JUMLAH	Integer	-	Jumlah barang
KETERANGAN	Variable character	100	Keterangan transaksi
HARGA	Decimal	12	Harga satuan barang yang diretur
SUBTOTAL	Decimal	12	Harga total barang yang diretur dari hasil perkalian JUMLAH dan HARGA

3.1.21 Deskripsi Entitas TBL_DETIL_RETUR_PEMBELIAN

Nama	Tipe	Panjang	Keterangan
IDDETILRETURPEMBELIAN	Integer	-	Nomor identitas unik dari setiap transaksi detail retur pembelian, Primary Key
IDRETURPEMBELIAN	Integer	-	Nomor identitas id retur pembelian, Foreign Key
KODEBARANG	Variable character	20	Kode barang, Foreign Key
KODESUPPLIER	Variable character	5	Kode supplier
KODEGUDANG	Variable character	6	Kode gudang
KODERAK	Variable character	6	Kode rak
JUMLAH	Integer	-	Jumlah barang
KETERANGAN	Variable character	100	Keterangan transaksi
IDMUTASI	Integer	-	Id mutasi, Foreign Key
HARGA	Decimal	12	Harga satuan barang
SUBTOTAL	Decimal	12	Harga total barang yang diretur dari hasil perkalian JUMLAH dan HARGA

3.1.22 Deskripsi Entitas TBL_DETIL_PIUTANG

Nama	Tipe	Panjang	Keterangan
IDPIUTANG	Integer	-	Nomor identitas unik dari setiap transaksi piutang, Foreign Key
CICILANKE	Integer	-	Urutan cicilan dari satu transaksi piutang
TANGGALCICILAN	DateTime	-	Tanggal dilakukan pembayaran cicilan piutang
JUMLAHCICILAN	Decimal	12	Jumlah cicilan

3.1.23 Deskripsi Entitas TBL_DETIL_PENJUALAN

Nama	Tipe	Panjang	Keterangan
IDDETILPENJUALAN	Integer	-	Nomor identitas unik dari setiap transaksi detail penjualan, Primary Key
NOFAKTURPENJUALAN	Variable character	15	Nomor faktur penjualan, Foreign Key
KODEBARANG	Variable character	20	Kode barang, Foreign Key
HARGA	Decimal	12	Harga jual per satuan barang
DISC_ITEM	Decimal	12	Diskon per satuan barang
QUANTITY	Integer	-	Jumlah barang
SUBTOTAL	Decimal	12	Subtotal dari (QUANTITY*HARGA) – (QUANTITY*DISC_ITEM)

3.1.24 Deskripsi Entitas TBL_DETIL_PEMESANAN

Nama	Tipe	Panjang	Keterangan
IDDETILPESANAN	Integer	-	Nomor identitas unik dari setiap transaksi detail pemesanan, Primary Key
NOMORPESANAN	Variable character	15	Nomor pesanan, Foreign Key

KODEJASA	Variable character	10	Kode jasa, Foreign Key
HARGA	Decimal	12	Harga jual per satuan jasa
DISC_ITEM	Decimal	12	Diskon per satuan
QUANTITY	Integer	-	Jumlah jual jasa
SUBTOTAL	Decimal	12	Subtotal dari (QUANTITY*HARGA) – (QUANTITY*DISC_ITEM)
KETERANGAN	Variable character	500	Keterangan transaksi

3.1.25 Deskripsi Entitas TBL_DETIL_PEMBELIAN

Nama	Tipe	Panjang	Keterangan
IDDETILPEMBELIAN	Integer	-	Nomor identitas unik dari setiap transaksi detil pembelian, Primary Key
NOFAKTURPEMBELIAN	Variable character	10	Nomor faktur pembelian, Foreign Key
KODEBARANG	Variable character	20	Kode barang, Foreign Key
HARGA	Decimal	12	Harga beli satuan barang
DISC_ITEM	Decimal	12	Diskon per satuan barang
QUANTITY	Integer	-	Jumlah beli
SUBTOTAL	Decimal	12	Subtotal harga barang setelah dikurangi diskon
KODEGUDANG	Variable character	6	Kode lokasi, Foreign Key
KODERAK	Variable character	6	Kode rak
IDMUTASI	Integer	-	Id mutasi, Foreign Key

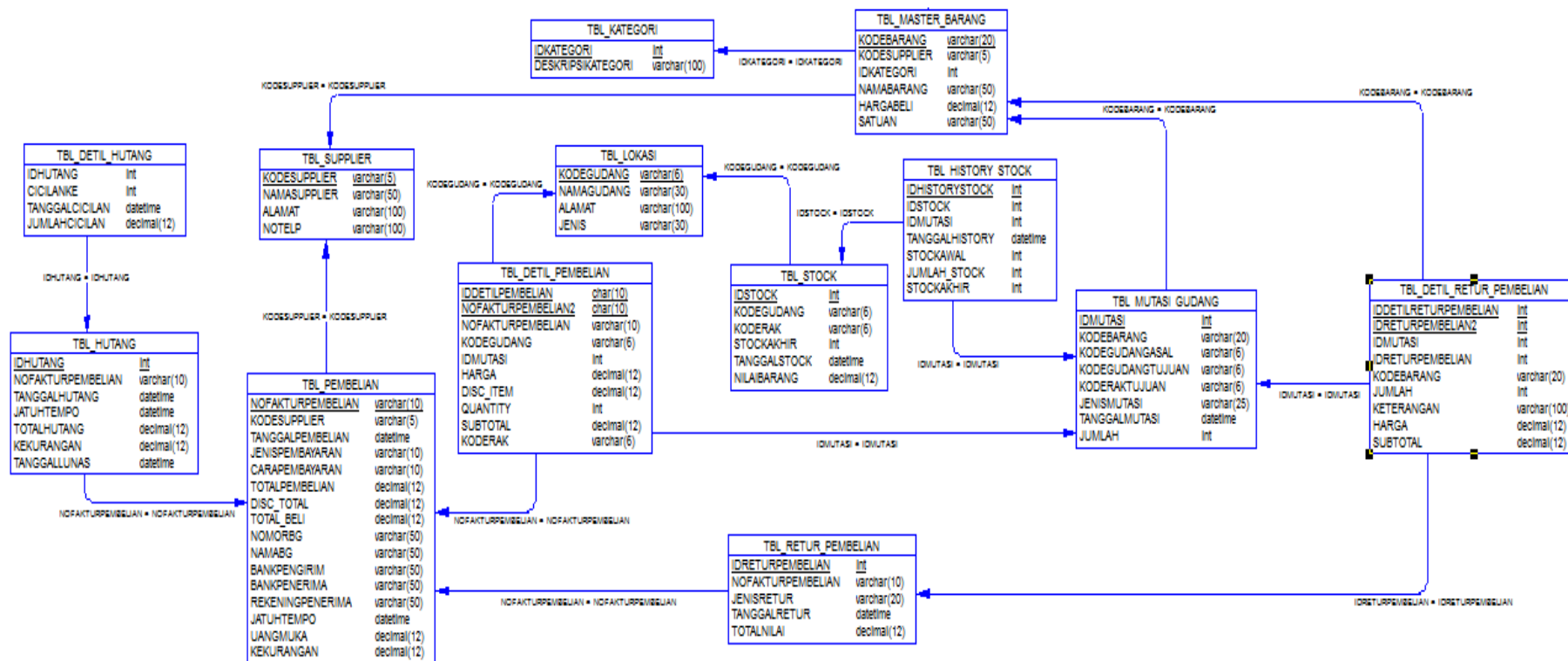
3.1.26 Deskripsi Entitas TBL_DETIL_HUTANG

Nama	Tipe	Panjang	Keterangan
IDHUTANG	Integer	-	Nomor identitas unik dari setiap transaksi hutang, Foreign Key
CICILANKE	Integer	-	Urutan cicilan dari suatu cicilan hutang
TANGGALCICILAN	DateTime	-	Tanggal pembayaran cicilan hutang
JUMLAHCICILAN	Decimal	12	Jumlah pembayaran

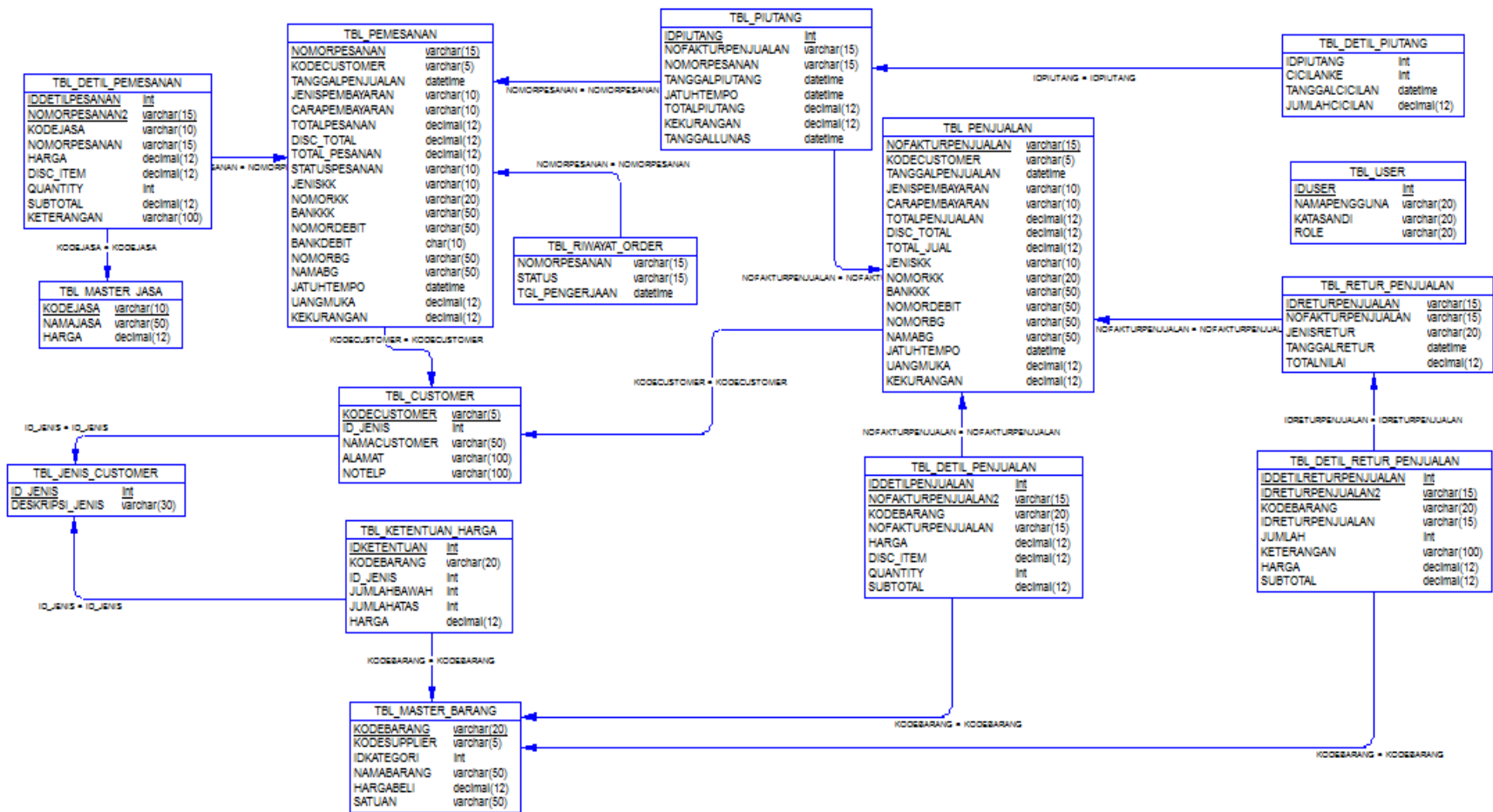
3.1.27 Deskripsi Entitas TBL_CUSTOMER

Nama	Tipe	Panjang	Keterangan
KODECUSTOMER	Variable character	5	Nomor identitas unik dari setiap customer, Primary Key
NAMACUSTOMER	Variable character	50	Nama customer
ALAMAT	Variable character	100	Alamat customer
NOTELP	Variable character	100	Nomor telpon customer
ID_JENIS	Integer	-	Id jenis customer, Foreign Key

3.2 Physical Data Model



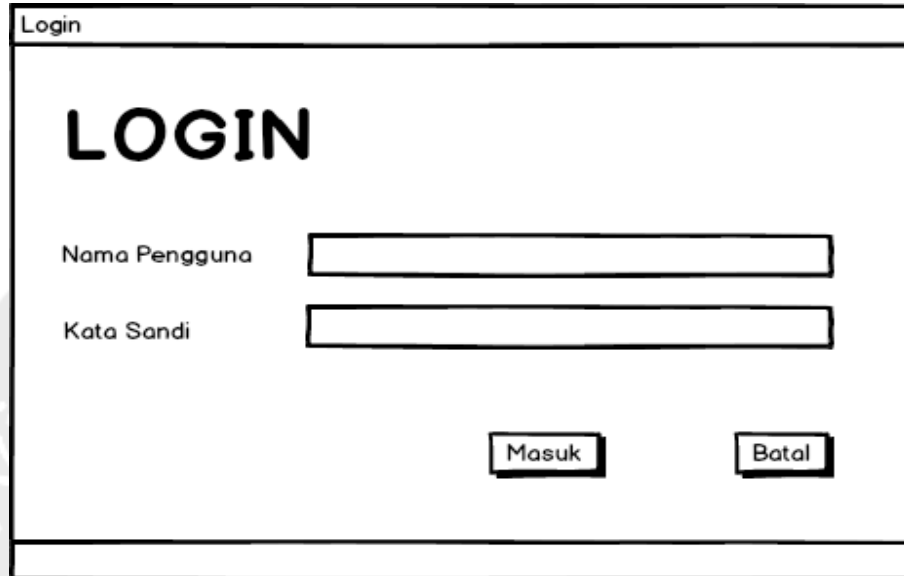
Gambar 3.1 Physical Data Model



Gambar 3.2 Physical Data Model (lanjutan)

4 Deskripsi Perancangan Antarmuka

4.1 Antarmuka Halaman Login

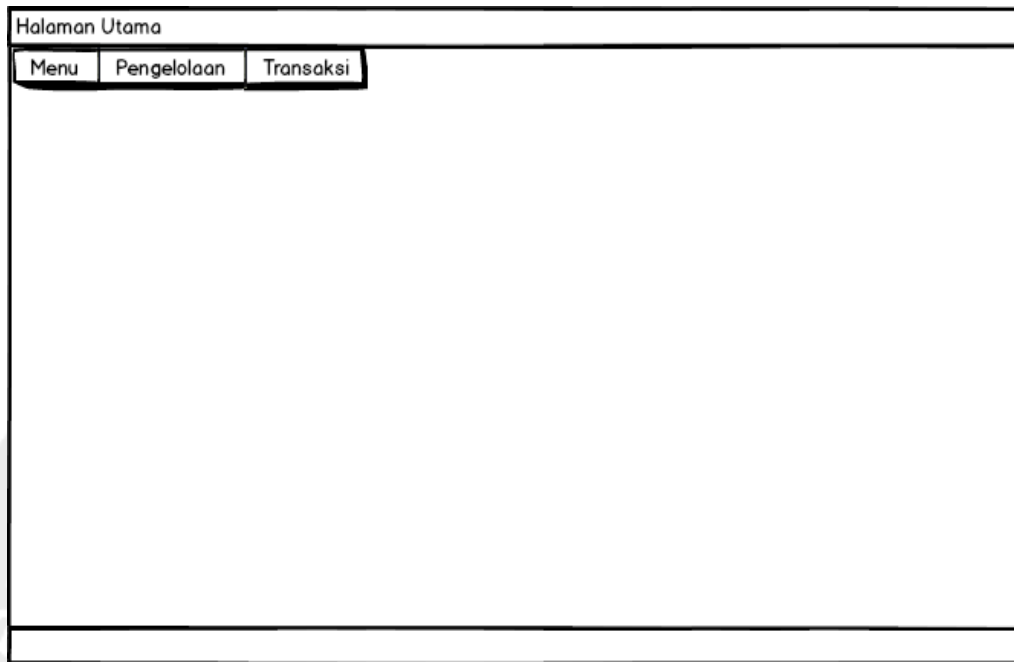


The image shows a login window titled "Login". Inside the window, the word "LOGIN" is displayed in large, bold, black letters. Below this, there are two text input fields. The first field is labeled "Nama Pengguna" and the second is labeled "Kata Sandi". At the bottom of the window, there are two buttons: "Masuk" (Login) and "Batal" (Cancel).

Gambar 4.1 Antarmuka Login

Rancangan antarmuka ini digunakan semua pengguna untuk melakukan login ke SITB. Untuk mendapat akses ke dalam sistem, pengguna harus memasukkan nama pengguna dan kata sandi pada textbox yang telah disediakan, kemudian menekan tombol Masuk. Jika nama pengguna dan kata sandi ditemukan di database, maka pengguna akan masuk dalam halaman utama sesuai dengan role. Sedangkan jika nama pengguna dan kata sandi tidak ditemukan data yang cocok di database, akan muncul pesan peringatan.

4.2 Antarmuka Halaman Utama



Gambar 4.2 Antarmuka Halaman Utama

Rancangan antarmuka ini merupakan antarmuka halaman utama yang berisi menu untuk masuk ke antarmuka-antarmuka yang lain. Pengguna dapat menggunakan menu bar yang berada di atas. Setelah login, masing-masing pengguna yang masuk mempunyai hak akses menu bar yang berbeda-beda. Menu berisi Dummy Stock, Backup Database, Logout dan Exit. Pengelolaan berisi pengelolaan user, customer, supplier, jenis customer, kategori barang, master barang, master jasa, lokasi, ketentuan harga, pencarian history stock, pencarian stock, dan cetak label barang. Transaksi berisi pembelian, retur pembelian, pembayaran hutang, penjualan, retur penjualan, pembayaran piutang, order, mutasi gudang dan reporting.

4.3 Antarmuka Halaman Pengelolaan User

The screenshot shows a web application interface for user management. At the top, it says 'Pengelolaan User' and 'PENGELOLAAN USER'. Below that is a search bar labeled 'Pencarian Pengguna'. A table with four columns: 'ID', 'Nama', 'Kata Sandi', and 'Role' is shown. Below the table is a form titled 'Data Pengguna' with fields for 'ID' (with a 'Label' next to it), 'Nama', 'Kata Sandi', and 'Role' (a 'ComboBox'). To the right of the form are four buttons: 'Tambah', 'Simpan', 'Ubah', and 'Hapus'.

Gambar 4.3 Antarmuka Pengelolaan User

Antarmuka pengelolaan user ini hanya dapat digunakan oleh owner. Antarmuka ini dapat digunakan untuk menambah data user, mengubah data user, menghapus data user maupun menampilkan data user. Keadaan awal saat antarmuka ini dipanggil groupbox data pengguna, button simpan, button hapus, button ubah adalah disable. Untuk melakukan penambahan data user dengan menekan tombol Tambah, lalu semua textbox akan aktif. Begitu juga dengan tombol Simpan akan aktif. User dapat melengkapi input data pada bidang isian yang ada. Jika input data sudah selesai dan tombol Simpan ditekan, maka sistem akan mengecek inputan user apakah sudah benar. Jika ada yang salah maka akan tampil pesan kesalahan.

User dapat melakukan ubah atau hapus data dengan cara mengklik salah satu baris data yang ada pada datagrid/ tabel yang ada. Kemudian untuk menghapus tinggal klik tombol Hapus, lalu akan muncul pesan konfirmasi penghapusan. Tekan tombol Yes, jika yakin untuk menghapus. Jika sistem mengecek dan ternyata data tidak dapat dihapus akan muncul pesan kesalahan. Untuk mengubah data, setelah klik salah satu baris data kemudian user hanya perlu melakukan input perubahan. Kemudian tekan Simpan bila sudah selesai. Lalu sistem kembali akan melakukan pengecekan.

4.4 Antarmuka Halaman Pengelolaan Customer

The screenshot displays a web application interface for customer management. At the top, it is titled 'Pengelolaan Customer' and 'PENGLOLAAN CUSTOMER'. Below the title is a search bar labeled 'Pencarian Customer'. A table with five columns (Kode, Nama, Alamat, No Telp, Jenis) is shown, with a greyed-out body. Below the table is a form titled 'Data Customer' with input fields for Kode, Nama, No Telp, and Jenis (a dropdown menu), and a larger text area for Alamat. At the bottom, there are four buttons: 'Tambah', 'Simpan', 'Ubah', and 'Hapus', along with a 'Print Data Customer' button.

Gambar 4.4 Antarmuka Pengelolaan Customer

Antarmuka pengelolaan customer ini dapat diakses user yang mempunyai role sebagai Owner maupun Manager. Melalui antarmuka ini user dapat melakukan penambahan data, pengubahan, penghapusan, dan mencetak semua data

yang ada. Untuk melakukan penambahan data, user harus menekan tombol Tambah terlebih dahulu lalu tombol Tambah terdisable dan groupbox untuk input data textbox akan aktif. Kemudian untuk menyimpan data yang telah diinputkan, user harus menekan tombol Simpan. Untuk melakukan ubah atau hapus data, user harus memilih salah satu data pada datagrid terlebih dahulu.

4.5 Antarmuka Halaman Pengelolaan Supplier

The screenshot shows a web application window titled "Pengelolaan Supplier". Inside, there is a sub-header "PENGLOLAAN SUPPLIER". Below this is a search section labeled "Pencarian Supplier" with a text input field. A data grid follows, with columns labeled "Kode", "Nama", "Alamat", and "No Telp". Below the grid is a form titled "Data Supplier" containing four input fields: "Kode", "Nama", "No Telp", and "Alamat". At the bottom of the window, there are five buttons: "Tambah", "Simpan", "Ubah", "Hapus", and "Print Data Supplier".

Gambar 4.5 Antarmuka Pengelolaan Supplier

Antarmuka ini dapat diakses oleh Owner. Melalui antarmuka ini user dapat melakukan penambahan data, pengubahan, penghapusan, dan mencetak semua data yang ada. Pada saat antarmuka ini dipanggil pada datagrid akan ditampilkan semua data supplier yang sudah tersimpan dalam database. Untuk melakukan penambahan data, user harus menekan tombol Tambah terlebih dahulu

lalu tombol Tambah terdisable dan groupbox untuk input data textbox akan aktif. Kemudian untuk menyimpan data yang telah diinputkan, user harus menekan tombol Simpan. Untuk melakukan ubah atau hapus data, user harus memilih salah satu data pada datagrid terlebih dahulu. Lalu untuk mencetak semua data supplier, user tekan tombol Print Data Supplier maka akan muncul report yang berisi data supplier dan siap untuk dicetak.

4.6 Antarmuka Halaman Pengelolaan Jenis Customer

The screenshot shows a software interface for managing customer types. It features a title bar, a main header, a data grid, an input section, and a set of action buttons.

Gambar 4.6 Antarmuka Pengelolaan Jenis Customer

Antarmuka ini dapat diakses oleh Owner. Melalui antarmuka ini user dapat melakukan penambahan data jenis customer, perubahan, penghapusan data jenis customer.

4.7 Antarmuka Halaman Pengelolaan Kategori Barang

The image shows a software interface for managing categories. It features a title bar 'Pengelolaan Kategori Barang', a main heading 'PENGELOLAAN KATEGORI BARANG', a table with columns 'ID' and 'Kategori', a search section 'Data Kategori Barang' with a search bar, and four action buttons: 'Tambah', 'Simpan', 'Ubah', and 'Hapus'.

Gambar 4.7 Antarmuka Pengelolaan Kategori Barang

Antarmuka ini hanya dapat diakses Owner. User dapat melakukan tambah kategori barang, ubah, hapus, cari, print data kategori. Untuk melakukan pencarian data kategori user hanya perlu memasukkan kata kunci pada textbox yang berada di paling atas. Pencarian akan dilakukan sistem dengan mencari data kategori berdasarkan deskripsi kategori. Lalu untuk mencetak semua data kategori, user tekan tombol Print Data Kategori maka akan muncul report yang berisi semua data kategori dan siap untuk dicetak.

4.8 Antarmuka Halaman Pengelolaan Barang

Pengelolaan Barang

PENGELOLAAN DATA BARANG

Pencarian Barang

Kode	Nama Barang	Harga Beli	Satuan	Kategori	Supplier
------	-------------	------------	--------	----------	----------

Data Barang

Barcode

Kode

Nama

Supplier

Satuan

Harga Beli

Kategori

Tambah

Simpan

Ubah

Hapus

Print Data Barang

Gambar 4.8 Antarmuka Pengelolaan Barang

Antarmuka pengelolaan barang ini hanya dapat digunakan oleh owner. Antarmuka ini dapat digunakan untuk menambah data barang, mengubah data barang, menghapus data barang, mencari data barang, menampilkan data barang maupun mencetak semua data barang. Keadaan awal saat antarmuka ini dipanggil groupbox data barang, button simpan, button hapus, button ubah adalah disable. Untuk melakukan penambahan data barang dengan menekan tombol Tambah, lalu semua textbox akan aktif. Begitu juga dengan tombol Simpan akan aktif. User dapat melengkapi input data pada bidang isian yang ada.

Ketika user telah menginputkan kode dan nama barang, pada picture box barcode akan muncul gambar barcode dari kode barang yang diinput. Kode barang tersebut bisa menjadi gambar barcode dengan menggunakan library eksternal milik ByteScout. Jika input data sudah selesai dan tombol Simpan ditekan, maka sistem akan mengecek inputan user apakah sudah benar. Jika ada yang salah maka akan tampil pesan kesalahan.

User dapat melakukan ubah atau hapus data dengan cara mengklik salah satu baris data yang ada pada datagrid/ tabel yang ada. Kemudian untuk menghapus tinggal klik tombol Hapus, lalu akan muncul pesan konfirmasi penghapusan. Tekan tombol Yes, jika yakin untuk menghapus. Jika sistem mengecek dan ternyata data tidak dapat dihapus akan muncul pesan kesalahan. Untuk mengubah data, setelah klik salah satu baris data kemudian user hanya perlu melakukan input perubahan. Kemudian tekan Simpan bila sudah selesai. Lalu sistem kembali akan melakukan pengecekan. User juga dapat mencetak data barang dengan klik tombol Print Data Barang. Lalu muncul report yang berisi semua data kategori dan siap untuk dicetak.

Program Studi Teknik Informatika	DPPL – SITB	104/ 122
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4.9 Antarmuka Halaman Pengelolaan Jasa

The screenshot shows a web application interface for service management. At the top, it says 'Pengelolaan Jasa' and 'PENGELOLAAN DATA JASA'. Below this is a search bar labeled 'Pencarian Jasa'. A table with three columns: 'Kode', 'Nama', and 'Harga' is shown, but it is currently empty. Below the table is a form titled 'Data Jasa' with three input fields: 'Kode Jasa', 'Nama', and 'Harga'. To the right of these fields are four buttons: 'Tambah', 'Simpan', 'Ubah', and 'Hapus'. At the bottom center of the form area is a button labeled 'Print Data Jasa'.

Gambar 4.9 Antarmuka Pengelolaan Jasa

Antarmuka ini dapat diakses oleh Owner. Melalui antarmuka ini user dapat melakukan penambahan data, pengubahan, penghapusan, dan mencetak semua data yang ada. Pada saat antarmuka ini dipanggil pada datagrid akan ditampilkan semua data master jasa yang sudah tersimpan dalam database. Untuk melakukan penambahan data, user harus menekan tombol Tambah terlebih dahulu lalu tombol Tambah terdisable dan groupbox untuk input data textbox akan aktif. Kemudian untuk menyimpan data yang telah diinputkan, user harus menekan tombol Simpan. Untuk melakukan ubah atau hapus data, user harus memilih salah satu data pada datagrid terlebih dahulu. Lalu untuk mencetak semua data jasa, user tekan tombol Print Data Jasa maka akan muncul report yang berisi semua data jasa dan siap untuk dicetak.

4.10 Antarmuka Halaman Pengelolaan Lokasi

The screenshot shows a web application interface for location management. At the top, there is a title bar 'Pengelolaan Lokasi' and a main header 'PENGLOLAAN LOKASI'. Below the header is a search bar labeled 'Pencarian Lokasi'. Underneath the search bar is a data grid with columns for 'Kode', 'Nama', 'Alamat', and 'Jenis'. Below the grid is a form titled 'Data Lokasi' with input fields for 'Kode', 'Nama', 'Alamat', and a 'Jenis' dropdown menu (labeled 'ComboBox'). At the bottom of the form are four buttons: 'Tambah', 'Simpan', 'Ubah', and 'Hapus'.

Gambar 4.10 Antarmuka Pengelolaan Lokasi

Antarmuka pengelolaan lokasi ini hanya dapat digunakan oleh owner. Antarmuka ini dapat digunakan untuk menambah data lokasi, mengubah data lokasi, menghapus data lokasi, mencari data lokasi, menampilkan data lokasi. Keadaan awal saat antarmuka ini dipanggil pada datagrid ditampilkan semua data lokasi yang sudah tersimpan pada database. Untuk melakukan penambahan data barang dengan menekan tombol Tambah, lalu semua textbox akan aktif. Begitu juga dengan tombol Simpan akan aktif. User melengkapi input data pada bidang isian yang ada. Jika input data sudah selesai dan tombol Simpan ditekan, maka sistem akan mengecek inputan user apakah sudah benar. Jika ada yang salah maka akan tampil pesan kesalahan.

4.11 Antarmuka Halaman Pengelolaan Ketentuan Harga

Pengelolaan Ketentuan Harga

PENGLOLAAN KETENTUAN HARGA

Pencarian Harga

Nama Barang	Satuan	Jumlah Bawah	Jumlah Atas	Harga	Jenis Customer
-------------	--------	--------------	-------------	-------	----------------

Barang

Kode Barang Cari

Nama Barang

Ketentuan Harga

ID Ketentuan Label

Jenis Customer

Jumlah Bawah

Jumlah Atas

Harga

Gambar 4.11 Antarmuka Pengelolaan Ketentuan Harga

Antarmuka ini digunakan oleh Owner untuk mengatur harga jual suatu barang. Harga jual ini dipengaruhi oleh 3 faktor yaitu barang itu sendiri, jenis customer yang membeli, dan jumlah yang dibeli. Pada saat antarmuka ini dipanggil, pada datagrid tampil semua ketentuan harga yang sudah ada. User dapat menambahkan ketentuan harga untuk suatu barang dengan klik tombol Cari maka akan muncul antarmuka pencarian barang. Setelah barang dipilih, klik tombol Tambah. Kemudian bidang isian pada groupbox ketentuan harga akan aktif. User memilih jenis customer dari combobox yang ada. Textbox jumlah bawah akan terisi secara otomatis sesuai dengan hasil pengecekan oleh sistem dengan data yang ada pada database. User hanya perlu menginputkan pada

textbox jumlah atas dan harga. Jika sudah selesai input, tekan tombol Simpan. Sistem akan mengecek inputan. Jika inputan sudah benar, maka akan muncul notifikasi data telah tersimpan. Kemudian akan muncul pertanyaan konfirmasi apakah user ingin menambahkan ketentuan harga lagi untuk barang yang sama. Hal ini ditujukan agar proses penambahan data ini dapat berlangsung lebih cepat.

User dapat juga melakukan perubahan data ketentuan harga dengan cara klik salah satu baris data yang ada pada tabel. Kemudian user input perubahan data, setelah selesai tinggal tekan tombol Simpan. User juga dapat menghapus data dengan cara yang sama seperti ubah data. Setelah data dipilih user tekan tombol Hapus. Untuk melakukan cetak data barang, tekan tombol Print Data.

4.12 Antarmuka Halaman Pencarian History Stock

Gambar 4.12 Antarmuka Pencarian History Stock

Antarmuka ini digunakan oleh Owner saja untuk melakukan pencarian data history stock. Pencarian history stock dapat dilakukan user berdasarkan kode barang dan range tanggal history. User memilih kode barang dari combobox yang ada, kemudian mengisi tanggal awal dan tanggal akhir dari datetimepicker. Setelah selesai tekan tombol Cari. Data hasil pencarian akan tampil pada datagrid. Apabila tidak ada data dari hasil pencarian, sistem tidak akan menampilkan pesan peringatan apapun.

4.13 Antarmuka Halaman Pencarian Stock

Gambar 4.13 Antarmuka Pencarian Stock

Antarmuka ini dapat digunakan oleh semua user untuk mencari data stock barang terakhir. Semua data stock yang ada pada database akan ditampilkan pada datagrid. Pencarian data stock dilakukan berdasarkan kode barang saja. User hanya perlu memilih kode barang dari combobox lalu tekan tombol Cari. User dapat mencetak data stock dengan tekan tombol Cetak.

4.14 Antarmuka Halaman Cetak Label Barang

Kode Barang	Nama Barang	Harga Jual
-------------	-------------	------------

Gambar 4.14 Antarmuka Cetak Label Barang

Antarmuka ini digunakan user untuk mencetak label barang yang diinginkan. User klik salah satu baris data yang ada pada tabel terlebih dahulu. Maka akan muncul preview hasil label di groupbox hasil label. Kemudian user memasukkan input jumlah label dan cetak mulai pada textbox yang tersedia. Cetak label ini sudah ditujukan untuk dicetak diatas kertas stiker label dengan jumlah 8 x 5 label. Sehingga jumlah label dibatasi maksimum 40 label dalam 1 kali cetak dan cetak mulai merupakan index cetak label akan dimulai.

4.15 Antarmuka Halaman Pembelian

Pembelian

PEMBELIAN

Data Pembelian

Nomor Faktur Supplier Jenis Pembayaran Cash Hutang

Tanggal Pembelian Cara Pembayaran

Kode Barang

Nama Barang

Satuan

Quantity

Harga Beli

Disc. Item

Subtotal

Gudang Simpan

Kode Rak

Jumlah

Disc. Total

Grand Total

Gambar 4.15 Antarmuka Pembelian

Antarmuka pembelian ini hanya dapat digunakan oleh Owner untuk melakukan transaksi pembelian. Untuk memulai transaksi pembelian, user perlu memasukkan nomor faktur lalu memilih supplier dan memilih jenis dan cara pembayaran. Kemudian tekan tombol Transaksi Baru. Sistem akan melakukan pengecekan terhadap data yang telah diinputkan terlebih dahulu. Jika tidak ada yang salah dengan inputan, maka user dapat menambahkan barang yang akan dibeli.

Selama proses menambahkan barang yang dibeli, user juga dapat menghapus barang yang dibeli dengan cara klik salah satu data yang ada pada datagrid lalu tekan tombol Hapus. Semua data yang telah diinputkan tidak akan tersimpan dalam database sebelum user menekan tombol Simpan Transaksi.

4.16 Antarmuka Halaman Retur Pembelian Keluar

Retur Pembelian Keluar

RETUR PEMBELIAN (Keluar)

Data Retur Pembelian

ID Retur	<input type="text"/>	<input type="button" value="Retur Baru"/>	Total XXXXXX
Tanggal Retur	<input type="text" value="/ /"/> <input type="button" value="📅"/>		

Data Barang Retur

Kode Barang	<input type="text"/>	<input type="button" value="Cari"/>	Jumlah	<input type="text"/>
Nama Barang	<input type="text"/>		Subtotal	<input type="text"/>
Supplier	<input type="text"/>		Keterangan	<input style="width: 100%; height: 100%;" type="text"/>
Kode Gudang	<input type="text"/>			
Kode Rak	<input type="text"/>			
Harga Beli	<input type="text"/>			
			<input type="button" value="Tambah"/> <input type="button" value="Hapus"/> <input type="button" value="Simpan"/>	

Gambar 4.16 Antarmuka Retur Pembelian Keluar

Antarmuka ini digunakan apabila ada transaksi retur pembelian keluar. Transaksi yang dimaksud adalah pengembalian barang yang telah dibeli kepada supplier. Untuk melakukan retur pembelian keluar, user perlu memilih tanggal retur dahulu. Saat antarmuka aktif, tanggal sudah terset pada tanggal sekarang. Kemudian tekan tombol Retur Baru, maka groupbox retur pembelian akan aktif. User dapat menambahkan barang yang diretur. Ketika user memilih barang, sistem akan mendapatkan data dari data stock barang. Sehingga jika jumlah yang dimasukkan user tidak mencukupi akan muncul peringatan. Setelah selesai input data user tekan tombol Tambah. Semua data yang telah diinputkan tidak akan tersimpan dalam database sebelum user menekan tombol Simpan.

4.17 Antarmuka Halaman Retur Pembelian Masuk

Retur Pembelian Keluar

RETUR PEMBELIAN (Masuk)

Data Retur Pembelian

Supplier

Tanggal Retur

Total XXXXXX

Data Barang Retur

Kode Barang

Nama Barang

Satuan

Harga Beli

Jumlah Sisa Keluar

Gudang Simpan

Jumlah Masuk

Subtotal

Keterangan

Gambar 4.17 Antarmuka Retur Pembelian Masuk

Antarmuka ini digunakan apabila ada transaksi retur pembelian masuk. Transaksi yang dimaksud adalah pengembalian barang dari supplier kepada user atas barang yang diretur pembelian keluar. Untuk melakukan retur pembelian masuk, tekan tombol Retur Baru maka groupbox retur pembelian akan aktif. User dapat menambahkan barang yang diretur dari banyak supplier. Sehingga user perlu memilih supplier dari combobox terlebih dahulu, baru kemudian memilih barang. Ketika user memilih barang, sistem akan mendapatkan data dari data barang yang sudah diretur pembelian keluar. Sehingga jika jumlah yang dimasukkan user tidak mencukupi akan muncul peringatan. Setelah selesai input data, user tekan tombol Tambah. Semua data yang telah diinputkan tidak akan tersimpan dalam database sebelum user menekan tombol Simpan.

4.18 Antarmuka Halaman Pembayaran Hutang

PEMBAYARAN HUTANG			
Data Hutang			
No Faktur	<input type="text"/>	Cicilan Ke	<input type="text"/>
Supplier	<input type="text"/>	Kekurangan	<input type="text"/>
Total Hutang	<input type="text"/>	Jumlah Bayar	<input type="text"/>
Uang Muka	<input type="text"/>	Tanggal Bayar	<input type="text"/>
Jatuh Tempo	<input type="text"/>		<input type="text"/>
Status	<input type="text"/>		<input type="text"/>
<input type="button" value="Simpan"/>			

Gambar 4.18 Antarmuka Pembayaran Hutang

Antarmuka pembayaran hutang digunakan untuk membayar hutang atas suatu pembelian, sehingga untuk bisa melakukan pembayaran hutang user harus memasukkan nomor faktur pembelian terlebih dahulu pada textbox no faktur kemudian tekan Enter. Sistem akan mengecek nomor faktur yang diinputkan tersebut pada database. Jika ditemukan bahwa nomor faktur tersebut masih hutang maka akan muncul semua data-data hutang tersebut. Sedangkan jika nomor faktur tersebut bukan hutang maka akan muncul peringatan. Melalui antarmuka ini user dapat melakukan pembayaran hutang secara cicilan dengan memasukkan jumlah pembayaran, kemudian tekan Simpan. Data pembayaran hutang akan tersimpan dalam database. Jika ditemukan bahwa hutang atas nomor faktur tersebut sudah berstatus pembayaran LUNAS, maka ketika user mencoba input jumlah pembayaran dan tekan tombol Simpan akan muncul pesan peringatan.

4.19 Antarmuka Halaman Penjualan

The screenshot displays a web application interface for sales transactions. At the top, the title 'PENJUALAN' is centered. Below it, the 'Data Penjualan' section includes input fields for 'Nomor Faktur', 'Tanggal Penjualan' (with a date picker), 'Kode Customer', and 'Nama Customer'. A 'Total' field on the right shows 'XXXXXX'. The 'Detil Penjualan' section contains fields for 'Kode Barang', 'Nama Barang', 'Satuan', 'Quantity', 'Harga Jual', and 'Disc. Item' (with radio buttons for 'Rp' and '%'). It also features 'Subtotal', 'Tambah', and 'Hapus' buttons. At the bottom, there are 'Jumlah', 'Disc. Total', and 'Total' fields, along with 'Simpan Transaksi' and 'Pembayaran' buttons. A large greyed-out area on the right represents a data grid for listing items.

Gambar 4.19 Antarmuka Penjualan

Antarmuka penjualan ini dapat digunakan oleh semua user untuk melakukan transaksi penjualan. Untuk memulai transaksi penjualan, user hanya perlu memasukkan kode customer kemudian tekan Enter. Pada saat antarmuka aktif, textbox kode customer terset default ke 0. Kode customer 0 menunjukkan bahwa customer bukan merupakan member. Setelah tekan Enter, sistem akan melakukan generate nomor faktur penjualan. Groupbox detil penjualan akan aktif dan user dapat menambahkan barang.

Selama proses menambahkan barang, user juga dapat menghapus barang dengan cara klik salah satu data yang ada pada datagrid lalu tekan tombol Hapus. Setelah semua data barang diinputkan, user perlu tekan tombol Bayar untuk input data pembayaran. Setelah input data pembayaran tekan tombol Simpan Transaksi agar data penjualan tersimpan ke dalam database.

4.20 Antarmuka Halaman Retur Penjualan

Retur Penjualan

RETUR PENJUALAN

Data Retur Penjualan

No Faktur

Tanggal Retur

Customer

Total
XXXXXX

Data Barang Retur

Kode Barang

Nama Barang

Harga Beli

Jumlah

Subtotal

Keterangan

Gambar 4.20 Antarmuka Retur Penjualan

Antarmuka ini dapat diakses oleh semua user dan digunakan untuk melakukan transaksi retur penjualan. Transaksi yang dimaksud adalah ketika customer mengembalikan barang yang sudah dibelinya karena alasan tertentu. Sehingga untuk memulai transaksi ini dibutuhkan nomor faktur penjualan/ pesanan. User menginputkan nomor faktur, kemudian tekan Enter. Sistem akan melakukan pengecekan apakah nomor faktur tersebut valid atau ada pada database. Jika valid, groupbox retur penjualan akan aktif dan sistem akan mengambil semua data penjualan dari nomor faktur tersebut. User tinggal memilih barang yang diretur dari combobox kemudian memasukkan jumlah. Apabila jumlah melebihi akan muncul peringatan. Jika selesai tekan tombol Tambah. Data retur penjualan yang diinput tidak akan tersimpan dalam database jika user belum menekan tombol Simpan.

4.21 Antarmuka Halaman Order Baru

Penjualan Barang dan Jasa		
PENJUALAN ORDER		
Data Penjualan Nomor Faktur <input type="text"/> Kode Customer <input type="text"/> Tanggal Penjualan <input type="text"/> / <input type="text"/> / <input type="text"/> Nama Customer <input type="text"/>		Total XXXXXX
Detil Barang <input type="button" value="Tambah Barang"/> <input type="button" value="Hapus Barang"/>		
Detil Order Kode Jasa <input type="text"/> <input type="button" value="Cari"/> Nama Jasa <input type="text"/> Quantity <input type="text"/> Harga Jasa <input type="text"/> Disc. Item <input type="text"/> Subtotal <input type="text"/> Keterangan <input type="text"/> <input type="button" value="Tambah"/> <input type="button" value="Hapus"/>		
Jumlah <input type="text"/> Disc. Total <input type="text"/> Total <input type="text"/>	<input type="button" value="Pembayaran"/> <input type="button" value="Simpan Transaksi"/>	

Gambar 4.21 Antarmuka Order Baru

Antarmuka order baru ini digunakan untuk penjualan. Penjualan ini khusus digunakan untuk penjualan barang dan penjualan jasa atau penjualan jasa saja. User hanya perlu memasukkan kode customer untuk memulai penjualan. Untuk menambahkan barang, pada groupbox detil barang tekan tombol tambah maka akan muncul antarmuka untuk menambahkan barang. Untuk menambahkan jasa, pada groupbox detil order pilih jasa dengan klik tombol Cari. Kemudian masukkan jumlahnya, diskon jika ada dan keterangan. Lalu tekan tombol Tambah. Jika sudah selesai menambahkan barang atau jasa, tekan tombol Bayar untuk melakukan pembayaran. Muncul antarmuka pembayaran lalu isi data pembayaran. Setelah input data pembayaran tekan tombol Simpan Transaksi agar data order tersimpan ke dalam database.

4.22 Antarmuka Halaman Proses Order

Proses Order

PROSES ORDER

Cari Data Order by No Pesanan _____

No Pesanan	Tanggal	Kode Customer	Status Pesanan

Data Order

No Pesanan

Tanggal Masuk

Nama Customer

Status Sekarang

Edit Status

Status
ComboBox

Batal Simpan

Gambar 4.22 Antarmuka Proses Order

Antarmuka ini dapat digunakan oleh semua user untuk memproses status pesanan dari sebuah order. Proses suatu order melalui 3 tahap status yaitu MULAI untuk order yang baru masuk, PROSES untuk order yang sudah dikerjakan, dan SELESAI untuk order yang sudah selesai dikerjakan. Untuk memproses status pesanan, user hanya perlu klik salah satu baris data yang ada pada datagrid. Sistem akan menampilkan data order yang dipilih pada groupbox data order. Kemudian user hanya perlu tekan tombol Simpan tanpa memilih pada combobox status karena sudah diset otomatis.

4.23 Antarmuka Halaman Pembatalan Order

Pembatalan Order

PEMBATALAN ORDER

Cari Data Order _____

No Pesanan	Tanggal	Status Pesanan	Kode Customer	Total Pesanan
------------	---------	----------------	---------------	---------------

Batal Order

Gambar 4.23 Antarmuka Pembatalan Order

Antarmuka ini digunakan untuk membatalkan sebuah order. Untuk membatalkan sebuah order, user hanya perlu klik pada salah satu baris data pada datagrid. Kemudian klik tombol BATAL ORDER dan akan muncul pertanyaan konfirmasi pembatalan order. Tekan tombol Yes untuk membatalkan order. Order yang bisa dibatalkan hanya yang mempunyai status pesanan MULAI. Sehingga jika order yang dipilih mempunyai status pesanan selain MULAI, akan muncul pesan peringatan.

4.24 Antarmuka Halaman Pembayaran Piutang

Pembayaran Piutang

PEMBAYARAN PIUTANG

Data Piutang

No Faktur/ Pesanan Cicilan Ke

Nama Customer Kekurangan

Total Piutang Jumlah Bayar

Uang Muka Tanggal Bayar

Jatuh Tempo

Status

Gambar 4.24 Antarmuka Pembayaran Piutang

Antarmuka pembayaran piutang digunakan untuk membayar piutang atas suatu penjualan, sehingga untuk bisa melakukan pembayaran piutang user harus memasukkan nomor faktur penjualan terlebih dahulu pada textbox no faktur/pesanan kemudian tekan Enter. Sistem akan mengecek nomor faktur yang diinputkan tersebut pada database. Jika ditemukan bahwa nomor faktur tersebut masih piutang maka akan muncul semua data-data piutang tersebut. Sedangkan jika nomor faktur tersebut bukan piutang maka akan muncul peringatan. Melalui antarmuka ini user dapat melakukan pembayaran piutang secara cicilan dengan memasukkan jumlah pembayaran, kemudian tekan Simpan. Data pembayaran piutang akan tersimpan dalam database. Jika ditemukan bahwa piutang atas nomor faktur tersebut sudah berstatus pembayaran LUNAS, maka ketika user mencoba input jumlah pembayaran dan tekan tombol Simpan akan muncul pesan peringatan.

4.25 Antarmuka Halaman Mutasi Gudang

MUTASI GUDANG							
Detail Asal				Detail Tujuan			
Kode Barang	<input type="text"/>	Cari		Gudang Tujuan	<input type="text" value="ComboBox"/>		
Nama Barang	<input type="text"/>			Kode Rak	<input type="text"/>		
Kode Gudang	<input type="text"/>			Simpan			
Kode Rak	<input type="text"/>						
Jumlah	<input type="text"/>						
ID Mutasi	Barang	Kode Gudang Asal	Kode Gudang Tujuan	Kode Rak Tujuan	Jenis Mutasi	Tanggal	Jumlah

Gambar 4.25 Antarmuka Mutasi Gudang

Antarmuka ini digunakan untuk melakukan transaksi mutasi yaitu transaksi pemindahan barang antar gudang. User yang dapat melakukan mutasi hanya Owner. User perlu memilih barang yang akan dimutasi terlebih dahulu dengan tekan tombol Cari. Kemudian sistem akan mengambil data barang dari data stock. User hanya perlu memasukkan jumlah, jika jumlah barang yang akan dimutasi tidak mencukupi akan muncul pesan peringatan. Setelah input jumlah, user hanya perlu menginput gudang tujuan dan kode raknya. Setelah selesai tekan tombol Simpan.

4.26 Antarmuka Halaman Reporting

The screenshot shows a web application interface for reporting. At the top, there is a title bar labeled "Reporting". Below it, the word "REPORTING" is centered. A horizontal menu contains several tabs: "Penjualan", "Detil Penjualan", "Retur Penjualan", "Pembelian", "Detil Pembelian", "Retur Pembelian", "Hutang", "Piutang", and "Mutasi". The "Penjualan" tab is currently selected. Below the menu is a table with a header row containing five columns: "No Faktur", "Nama Customer", "Tanggal Penjualan", "Jenis Pembayaran", and "Total Penjualan". The table body is currently empty and shaded grey. Below the table is a section titled "Reporting Data Penjualan" which contains a search filter. This section includes a "Kriteria Reporting" dropdown menu set to "ComboBox", a "Kata Kunci" text input field, and a "Tanggal Penjualan" section with two date pickers separated by "s/d". At the bottom of this section are two buttons: "Tampil" and "Cetak".

Gambar 4.26 Antarmuka Reporting

Antarmuka ini hanya bisa diakses oleh Owner dan Manager. Melalui antarmuka ini user dapat mengakses report penjualan, detil penjualan, retur penjualan, pembelian, detil pembelian, retur pembelian, hutang, piutang, dan mutasi dengan cara memilih tab yang ada di bagian atas. User juga dapat melihat report sesuai dengan kriteria sorting yang diinginkan dengan cara input kriteria lalu tekan tombol Tampil. Selain itu user juga dapat mencetak report dengan tekan tombol Cetak.