

BAB VI

PENUTUP

Pada bab penutup ini akan dijabarkan mengenai kesimpulan dan saran dari pembangunan sistem, daftar pustaka yang digunakan penulis dalam menyusun laporan ini, dan lampiran yang berisi SKPL (Spesifikasi Kebutuhan Perangkat Lunak) dan DPPL (Dokumen Perancangan Perangkat Lunak) dari perancangan pembangunan sistem.

6.1 Kesimpulan

Berdasarkan analisis, desain, dan implementasi perangkat lunak dan pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Tugas Akhir ini yaitu Sistem Informasi Distribusi dan Penjualan (SISIP) untuk Bulog Mart Divisi Regional D.I. Yogyakarta telah berhasil dibangun dengan menggunakan bahasa pemrograman *PHP* dan *framework CodeIgniter*.

6.2 Saran

Saran yang dapat diambil dari proses analisis sampai pada pembuatan Sistem Informasi SISIP di tugas akhir ini adalah sebagai berikut:

1. Menambahkan fitur pada sistem informasi distribusi dan penjualan (SISIP) untuk melakukan pembayaran ke Supplier.
2. Menambahkan fitur pada sistem informasi berbasis web untuk melakukan pembayaran pada transaksi penjualan menggunakan debit, tidak hanya pembayaran tunai.

3. Menambahkan modul keuangan pada sistem informasi dan distribusi seperti dokumen bukti keluar uang, dokumen bukti terima uang, laporan kas bank harian, dan lain sebagainya.



DAFTAR PUSTAKA

- Anderson, S. & Stanton, D., n.d. *How Barcodes Work : An Introduction to Code 128*. [Online] Available at: <http://courses.cs.washington.edu/courses/cse370/01au/minirproject/BarcodeBattlers/barcodes.html> [Accessed 25 April 2016].
- Angipora, M.P., 2002. *Dasar-Dasar Pemasaran*. 2nd ed. Jakarta: PT. Raja Grafindo Persada.
- Anisya, 2013. Pembangunan Aplikasi Persediaan Beras Bulog Pada Kantor Lurah Parak Laweh Pulau AIE NAN XX Kota Padang. *Jurnal TEKNOIF*, 1(2), pp.26-34.
- Arifudzaki, B., Somantri, M. & Rochim, A.F., 2010. Aplikasi Sistem Informasi Persediaan Barang pada Perusahaan Ekspor Hasil Laut Berbasis Web. *Jurnal Ilmiah Teknik Elektro*, 12(4), pp.138-44.
- Aryanto, A., Tjendrowasono, T.I. & Riasti, B.K., 2012. Pembangunan Sistem Penjualan Online Pada Toko Indah Jaya Furniture Surakarta. *Indonesian Journal on Computer Science Speed (IJCSS)*, 9(3), pp.55-61.
- Chao, S., LI, B.N. & Dong, M.C., 2006. Barcode Technology in Blood Bank Information Systems. *Journal of Medical Systems*, Springer, 30(6), pp.449-57.
- Computabel International, 2014. *Code 128*. [Online] Available at: <http://www.computabel.com/about128.htm> [Accessed 25 April 2016].

- Daulay, I.S.S., 2013. *Hubungan Barcode dengan Produk Industri Sebagai Standar Perdagangan Produk Industri Masa Kini*. [Online] Widyaiswara Madya Pusdiklat Industri, Kementerian Perindustrian Republik Indonesia Available at: <http://www.kemperin.go.id/download/6760/Hubungan-BARCODE-dengan-Produk-Industri-Sebagai-Standar-Perdagangan-Produk-Industri-Masa-Kini>. [Accessed 5 May 2016].
- Fatta, H., 2007. *Analisis & Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan & Organisasi Modern*. 1st ed. Yogyakarta: Andi.
- Februariyanti, H. & Zuliarso, E., 2012. Rancang Bangun Sistem Perpustakaan untuk Jurnal Elektronik. *Jurnal Teknologi Informasi Dinamik*, XVII(2), pp.124-32.
- Furness, K.A.O., 2000. Potential for two-dimensional codes in automated manufacturing. *Assembly Automation, Emerald Insight*, 20(1), pp.52-57.
- Handita, D.B. & Umar & Fadlillah, U., 2012. Sistem Informasi Penjualan Berbasis Web Pada Usantex. *Jurnal Emitter*, 12(1), pp.26-33.
- Hariadi, D., 2012. Pengaruh Produk, Harga, Promosi, dan Distribusi Terhadap Keputusan Pembelian Konsumen Pada Produk Projector Microvision. *Jurnal Ilmu dan Riset Manajemen*, 1(8), pp.1-21.
- Hidayat, H., 2011. *Cara Instan Menguasai Pemrograman Web*. 1st ed. Jakarta: Agogos Publishing.

- Kosasi, S., 2014. Perancangan Aplikasi Point of Sale dengan Arsitektur Client/Server Berbasis Linux dan Windows. *Creative Information Technology Journal*, 1(2), pp.116-27.
- Kosasi, S., 2015. Perancangan Sistem Informasi Integrasi Aplikasi Point Of Sale Mini Market. *Techsi : Jurnal Penelitian Teknik Informatika*, 6(1), pp.81-102.
- Ma'roep, M., 2009. Penerapan Sistem Informasi Akuntansi Penjualan pada PT Indomobil Surabaya. *Jurnal Ekonomi Bisnis*, 14(3), pp.214-23.
- Mulyadi, 2008. *Sistem Akuntansi*. Jakarta: Salemba Empat.
- Mulyadi, D., 2011. Pengembangan Sistem Logistik Yang Efisien dan Efektif Dengan Pendekatan Supply Chain Management. *Jurnal Riset Industri*, V(3), pp.275-82.
- Purnamasari, I., 2013. Perancangan Sistem Informasi Peminjaman Buku dan Komik Pada Taman Bacaan Fortune Baleharjo Pacitan. *Indonesian Journal on Computer Science*, 10(3), pp.102-06.
- Riyadi, A., Retnandi, E. & Deddy, A., 2012. Perancangan Sistem Informasi Berbasis Website Subsistem Guru di Sekolah Pesantren Persatuan Islam 99 Rancabango. *Jurnal Algoritma Sekolah Tinggi Teknologi Garut*, 9(40), pp.1-11.
- Rosdianingsih, F., Utama, I.G.A. & Prasetya, H.P., 2012. Rancang Bangun Sistem Informasi Pendistribusian Barang Movement Regional pada Perum

Bulog Divre Jawa Timur. *Jurnal Institut Bisnis dan Informatika Stikom Surabaya*, 1(2), pp.1-10.

Rusono, N. et al., 2013. Rencana Pembangunan Jangka Menengah Nasional (RPJMN) Bidang Pangan dan Pertanian 2015-2019. In Nasional, K.P.P., ed. *Nasional, Kementerian Perencanaan Pembangunan*. Jakarta Pusat, 2013. Direktorat Pangan dan Pertanian, Bappenas.

Sasmita, A., Bayupati & Manu, G., 2009. Sistem Transaksi Penjualan Kain Gringsing Berbasis Mobile (Mobile Commerce). *Jurnal Teknologi Elektro*, VIII(1), pp.91-96.

Satyawan, A., Hariadi, B. & Amelia, T., 2013. Sistem Informasi Penggajian Menggunakan Presensi Sidik Jari (Studi Kasus Pada PT. Kuda Inti Samudera Cabang Surabaya). *Jurnal Sistem Informasi*, 2(1), pp.60-65.

Sidik, B., 2012. *Framework CodeIgniter*. Bandung: Informatika.

Suryadharma, B., 2012. Perancangan dan Pembuatan Sistem Informasi Penjualan, Pembelian, dan Stock Gudang di PT. Mitra Setia Budi Sejahtera Berbasis Web. Surabaya, 2012. Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember.

Suwarno, H., 2006. Sembilan Fungsi Saluran Distribusi : Kunci Pelaksanaan Kegiatan Distribusi yang Efektif. *Jurnal Manajemen*, 6(1), pp.79-87.

Tjiptono, F., 2008. *Strategi Bisnis Pemasaran* (hal.185). Yogyakarta: Andi Offset.

Warman, I. & Zahni, A., 2013. Rekayasa Web untuk Pemesanan Handphone Berbasis JQuery pada Permata Cell. *Jurnal Momentum*, 15(2), pp.30-38.

Widayanto, A. & Wardati, I., 2013. Perancangan Sistem Informasi Penjualan Spare Part Mobil Pada Bengkel Samsi Motor Pacitan. *Indonesian Journal on Networking and Security*, 2(1), pp.1-7.

Winarno, E., Zaki, A. & Community, S., 2011. *Easy Web Programming with PHP plus HTML 5*. Semarang: PT. Elex Media Komputindo.

Yun, Y., 2014. Pengendalian Persediaan Terhadap Distribusi Beras Raskin Pada Perum Bulog Divre Jabar. In *Seminar Bisnis & Teknologi*. Bandar Lampung, 2014. Lembaga Pengembangan Pembelajaran, Penelitian & Pengabdian Kepada Masyarakat.

LAMPIRAN



SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

Sistem Informasi Distribusi dan Penjualan Berbasis Web

(Studi Kasus : Bulog Mart Divre Daerah
Istimewa Yogyakarta)

Untuk :


Tugas Akhir (Skripsi)

Dipersiapkan oleh:

Ancilla Alfionita Intan C. / 12 07 07051

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		SKPL-SISIP		1/76
	Fakultas Teknologi Industri	REVISI		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	AAIC							
Diperiksa oleh	IW & FIN							
Disetujui oleh	IW & FIN							

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	9
1.5 Deskripsi umum (Overview)	9
2. Deskripsi Kebutuhan.....	10
2.1 Perspektif produk	10
2.2 Fungsi Produk	11
2.3 Karakteristik Pengguna	25
2.4 Batasan-batasan	26
2.5 Asumsi dan Ketergantungan	26
3. Kebutuhan khusus.....	26
3.1 Kebutuhan antarmuka eksternal	26
4. Spesifikasi Rinci Kebutuhan.....	31
4.1 Spesifikasi Kebutuhan Fungsionalitas	31
Entity Relationship Diagram (ERD)	76

Daftar Gambar

Gambar 1. Arsitektur Perangkat lunak SISIP.....	11
Gambar 2. Use Case Diagram.....	30
Gambar 3. <i>Entity Relationship Diagram</i> (ERD).....	68



1. Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SISIP (Sistem Informasi Distribusi dan Penjualan) 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-SISIP ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak SISIP dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan yang terkait dengan cabang, kategori barang, barang, detil barang, supplier, dan barang supplier yang ada di Bulog Mart.
2. Menangani pengelolaan pemesanan dan penerimaan barang supplier, serta pemesanan dan penerimaan barang ke pusat atau antar cabang.
3. Menangani pengelolaan transaksi penjualan Bulog Mart.
4. Menangani pengelolaan *reporting* jumlah transaksi penjualan pada setiap cabang, jumlah pendapatan setiap cabang, histori stok barang, dan histori harga barang.

Dan aplikasi ini berjalan pada lingkungan dengan *web browser*.

Program Studi Teknik Informatika	SKPL – SISIP	6/76
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-SISIP-XXX	Kode yang merepresentasikan kebutuhan pada SISIP (Sistem Informasi Distribusi dan Penjualan) dimana XXX merupakan nomor fungsi produk.
SISIP	Perangkat lunak pengelolaan web.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Networkglobal</i> yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk <i>e-mail</i> , FTP, dan <i>World Wide Web</i> .
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Bulog Mart	Salah satu pengembangan usaha Perum Bulog berupa toko retail sembako untuk membantu masyarakat memenuhi kebutuhan pokok.
Penjualan	Suatu kegiatan yang dilakukan oleh penjual untuk mempengaruhi pembeli agar bersedia membeli barang atau jasa yang ditawarkan sesuai dengan kesepakatan harga dua belah pihak.
Distribusi	Suatu proses kegiatan pemasaran yang bertujuan untuk mempermudah kegiatan

	penyaluran barang atau jasa dari produsen ke konsumen.
<i>Barcode</i>	Kombinasi kode unik yang berbentuk garis atau batangan balok berwarna hitam putih dengan ketebalan berbeda yang mewakili data atau informasi tertentu.
<i>Barcode Scanner</i>	Alat pembaca barcode menggunakan sinar laser yang sensitif terhadap ketebalan garis, jarak antar baris. Kemudian data akan ditransfer ke komputer untuk diolah dan ditampilkan sebagai data yang dapat dibaca.
<i>Framework</i>	Suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks
<i>PHP</i>	<i>Perl Hypertext Preprocessor</i> merupakan bahasa skrip pemrograman yang dapat disisipkan ke dalam HTML dan sering dipakai untuk memprogram web dinamis.
<i>CodeIgniter</i>	Aplikasi <i>open source</i> yang berupa framework dengan model MVC (<i>Model, View, Controller</i>) untuk membangun web dinamis menggunakan PHP.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Irene Deandra Indarto, *Spesifikasi Kebutuhan Perangkat Lunak (SKPL) SISIP*, Universitas Atma Jaya Yogyakarta.
2. Yohanes Yanuar Adi Nugroho, *Spesifikasi Kebutuhan Perangkat Lunak (SKPL)SITB*, Universitas Atma Jaya Yogyakarta.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian utama 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 ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SISIP 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 SISIP tersebut.

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

2. Deskripsi Kebutuhan

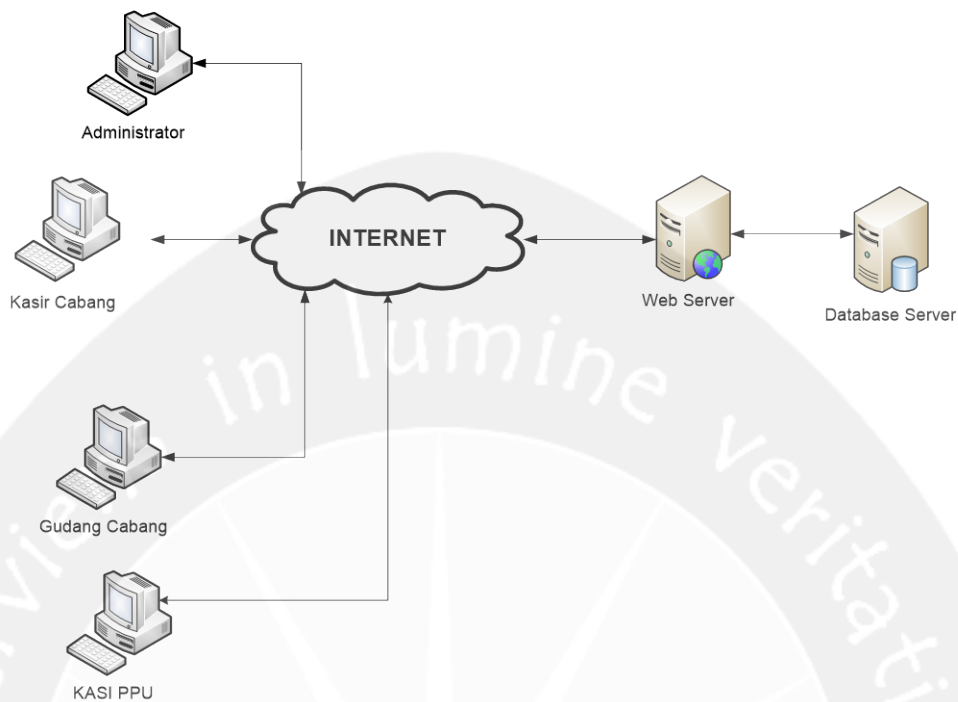
2.1 Perspektif produk

SISIP merupakan perangkat lunak yang dikembangkan untuk membantu proses bisnis terutama pada proses distribusi dan penjualan yang terjadi di Bulog Mart. Sistem ini dipergunakan oleh pihak internal Bulog Mart sendiri dan berupa aplikasi *web*.

Pada aplikasi *web*, sistem ini dapat digunakan oleh administrator, kasir, gudang, serta KASI PPU (Kepala Sie PPU) di setiap cabang dari Bulog Mart yang ada di Divre (Divisi Regional) Daerah Istimewa Yogyakarta. Melalui sistem tersebut administrator dapat melakukan pengelolaan data user, data cabang, data kategori barang, data barang, data detil barang, dan data supplier. Untuk pihak gudang dikhususkan untuk mengelola pemesanan dan penerimaan barang supplier ntuk gudang pusat, serta pemesanan dan penerimaan barang pusat ke cabang atau antar cabang. Selanjutnya kasir dapat melakukan pengelolaan transaksi penjualan yang terjadi di cabang Bulog Mart. Sedangkan KASI PPU dapat melakukan pengelolaan laporan jumlah penjualan, laporan jumlah pendapatan, histori stok barang, an histori harga barang.

Perangkat lunak *web* SISIP ini berjalan pada *web browser*, dan dikembangkan menggunakan bahasa pemrograman PHP dengan *framework CodeIgniter*. Selain itu, untuk lingkungan pemrogramannya menggunakan *Notepad++*, dan untuk *database* akan menggunakan *database MySQL*.

Program Studi Teknik Informatika	SKPL – SISIP	10/76
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 1. Arsitektur Perangkat lunak SISIP

2.2 Fungsi Produk

Fungsi produk perangkat lunak SISIP adalah sebagai berikut :

APLIKASI WEB

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

Merupakan fungsi yang digunakan oleh semua *user* untuk dapat masuk dalam sistem dan memperoleh hak akses sesuai dengan *role* yang dimiliki *user*.

2. Fungsi *Ubah Password* (**SKPL-SISIP-002**).

Merupakan fungsi yang digunakan untuk mengubah *password* dari *user*.

3. Fungsi *Pengelolaan Data User* (**SKPL-SISIP-003**).

Merupakan fungsi yang digunakan oleh *User* untuk mengelola data pribadi *user*. Kecuali pada bagian

Program Studi Teknik Informatika	SKPL – SISIP	11/76
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		

tambah user hanya bisa dilakukan oleh Administrator.

Fungsi Pengelolaan Data User mencakup :

- a. Fungsi *Tambah Data User* (**SKPL-SISIP-003-01**).
Merupakan fungsi yang digunakan untuk menambah data user dan hanya dapat dikelola oleh Administrator.
- b. Fungsi *Ubah Data User* (**SKPL-SISIP-003-02**).
Merupakan fungsi yang digunakan untuk mengubah data user.
- c. Fungsi *Hapus Data User* (**SKPL-SISIP-003-03**).
Merupakan fungsi yang digunakan untuk menghapus data user dan hanya dapat dikelola oleh Administrator.
- d. Fungsi *Tampil Data User* (**SKPL-SISIP-003-04**).
Merupakan fungsi yang digunakan untuk menampilkan data atau *profile user*.
- e. Fungsi *Cari Data User* (**SKPL-SISIP-003-05**)
Merupakan fungsi yang digunakan untuk mencari data user Bulog Mart.

4. Fungsi *Pengelolaan Data Cabang* (**SKPL-SISIP-004**)

Merupakan fungsi yang digunakan oleh Administrator untuk mengelola data cabang.

Fungsi Pengelolaan Data Cabang meliputi :

- a. Fungsi *Tambah Data Cabang* (**SKPL-SISIP-004-01**).
Merupakan fungsi yang digunakan untuk menambah data cabang yang baru.

Program Studi Teknik Informatika	SKPL – SISIP	12/76
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 *Ubah Data Cabang* (**SKPL-SISIP-004-02**).

Merupakan fungsi yang digunakan untuk mengubah data cabang.

c. Fungsi *Hapus Data Cabang* (**SKPLL-SISIP-004-03**).

Merupakan fungsi yang digunakan untuk menghapus data cabang.

d. Fungsi *Cari Data Cabang* (**SKPL-SISIP-004-04**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU dan Gudang untuk mencari data cabang Bulog Mart.

e. Fungsi *Tampil Data Cabang* (**SKPL-SISIP-004-05**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU, dan Gudang untuk menampilkan data cabang dari Bulog Mart.

5. Fungsi *Pengelolaan Data Supplier* (**SKPL-SISIP-005**)

Merupakan fungsi yang digunakan oleh Administrator untuk mengelola data supplier.

Fungsi Pengelolaan Data Supplier meliputi :

a. Fungsi *Tambah Data Supplier* (**SKPL-SISIP-005-01**).

Merupakan fungsi yang digunakan untuk menambah data supplier yang baru.

b. Fungsi *Ubah Data Supplier* (**SKPL-SISIP-005-02**).

Merupakan fungsi yang digunakan untuk mengubah data supplier.

c. Fungsi *Hapus Data Supplier* (**SKPLL-SISIP-005-03**).

Merupakan fungsi yang digunakan untuk menghapus data supplier.

d. Fungsi *Cari Data Supplier* (**SKPL-SISIP-005-04**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU dan Gudang untuk mencari data supplier Bulog Mart berdasarkan lokasinya.

e. Fungsi *Tampil Data Supplier* (**SKPL-SISIP-005-05**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU, dan Gudang untuk menampilkan data supplier dari Bulog Mart.

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

Merupakan fungsi yang digunakan oleh Administrator untuk mengelola data kategori barang.

Fungsi Pengelolaan Kategori Barang meliputi :

a. Fungsi *Tambah Kategori Barang* (**SKPL-SISIP-006-01**).

Merupakan fungsi yang digunakan untuk menambah kategori barang.

b. Fungsi *Ubah Kategori Barang* (**SKPL-SISIP-006-02**).

Merupakan fungsi yang digunakan untuk mengubah data kategori barang.

c. Fungsi *Hapus Kategori Barang* (**SKPL-SISIP-006-03**).

Merupakan fungsi yang digunakan untuk mengubah data kategori barang.

d. Fungsi *Cari Kategori Barang* (**SKPL-SISIP-006-04**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU dan Gudang.

e. Fungsi *Tampil Kategori Barang* (**SKPL-SISIP-006-05**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU dan Gudang untuk menampilkan kategori barang.

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

Merupakan fungsi yang digunakan oleh Administrator untuk mengelola data barang.

Fungsi Pengelolaan Data Barang meliputi :

a. Fungsi *Tambah Data Barang* (**SKPL-SISIP-007-01**).

Merupakan fungsi yang digunakan untuk menambah data barang yang baru.

b. Fungsi *Ubah Data Barang* (**SKPL-SISIP-007-02**).

Merupakan fungsi yang digunakan untuk mengubah data barang.

c. Fungsi *Hapus Data Barang* (**SKPL-SISIP-007-03**).

Merupakan fungsi yang digunakan untuk menghapus data barang.

Program Studi Teknik Informatika	SKPL – SISIP	15/76
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 *Cari Barang* (**SKPL-SISIP-007-04**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU, dan Gudang untuk menampilkan data barang berdasarkan nama barang, dan kategori.

e. Fungsi *Tampil Data Barang* (**SKPL-SISIP-007-05**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU, dan Gudang untuk menampilkan data barang.

8. Fungsi *Pengelolaan Data Detil Barang* (**SKPL-SISIP-008**)

Merupakan fungsi yang digunakan oleh Administrator untuk mengelola data detil barang.

Fungsi Pengelolaan Data Detil Barang meliputi :

a. Fungsi *Tambah Data Detil Barang* (**SKPL-SISIP-008-01**).

Merupakan fungsi yang digunakan untuk menambah data detil barang yang baru.

b. Fungsi *Ubah Data Detil Barang* (**SKPL-SISIP-008-02**).

Merupakan fungsi yang digunakan untuk mengubah data detil barang.

c. Fungsi *Hapus Data Detil Barang* (**SKPL-SISIP-008-03**).

Merupakan fungsi yang digunakan untuk menghapus data detil barang.

d. Fungsi *Cari Detil Barang* (**SKPL-SISIP-008-04**)

Program Studi Teknik Informatika	SKPL – SISIP	16/76
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 oleh Administrator, Kasir, KASI PPU, dan Gudang untuk menampilkan data detil barang.

e. Fungsi *Tampil Data Detil Barang* (**SKPL-SISIP-008-05**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU, dan Gudang untuk menampilkan data detil barang.

9. Fungsi *Pengelolaan Pemesanan Barang Supplier* (**SKPL-SISIP-009**)

Merupakan fungsi yang digunakan oleh Gudang untuk melakukan pencatatan pemesanan barang ke supplier.

Fungsi Pengelolaan Pemesanan Barang ke Supplier meliputi :

a. Fungsi *Generate Kode Pemesanan Barang Supplier* (**SKPL-SISIP-009-01**).

Merupakan fungsi yang digunakan untuk generate kode pemesanan barang ke supplier yang baru.

b. Fungsi *Tambah Pemesanan Barang Supplier* (**SKPL-SISIP-009-02**).

Merupakan fungsi yang digunakan untuk menambah data pemesanan barang ke supplier yang baru.

c. Fungsi *Batal Pemesanan Barang ke Supplier* (**SKPL-SISIP-009-03**).

Merupakan fungsi yang digunakan untuk membatalkan pemesanan barang ke supplier.

Program Studi Teknik Informatika	SKPL – SISIP	17/76
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 *Tampil Data Pemesanan Barang ke Supplier* (**SKPL-SISIP-009-04**)

Merupakan fungsi yang digunakan Gudang untuk menampilkan data pemesanan barang ke supplier.

e. Fungsi *Tambah Detil Pemesanan Barang ke Supplier* (**SKPL-SISIP-009-05**).

Merupakan fungsi yang digunakan untuk menambah data detil pemesanan barang ke supplier yang baru.

f. Fungsi *Ubah Detil Pemesanan Barang ke Supplier* (**SKPL-SISIP-009-06**).

Merupakan fungsi yang digunakan untuk mengubah data detil pemesanan barang ke supplier.

g. Fungsi *Hapus Detil Pemesanan Barang* (**SKPL-SISIP-009-07**).

Merupakan fungsi yang digunakan untuk menghapus data detil pemesanan barang ke supplier.

h. Fungsi *Tampil Detil Pemesanan Barang ke Supplier* (**SKPL-SISIP-009-08**).

Merupakan fungsi yang digunakan untuk menampilkan data detil pemesanan barang ke supplier.

i. Fungsi *Cetak Nota Dokumen Pemesanan untuk Supplier* (**SKPL-SISIP-009-09**).

Merupakan fungsi yang digunakan untuk mencetak dokumen pemesanan untuk supplier.

10. Fungsi *Pengelolaan Penerimaan Barang dari Supplier*
(**SKPL-SISIP-010**)

Merupakan fungsi yang digunakan oleh Gudang untuk mengelola data penerimaan barang dari supplier.

Fungsi *Pengelolaan Penerimaan Barang dari supplier* meliputi :

a. Fungsi *Tambah Penerimaan Barang dari supplier*
(**SKPL-SISIP-010-01**).

Merupakan fungsi yang digunakan untuk menambah data penerimaan barang dari supplier yang baru.

b. Fungsi *Ubah Detil Penerimaan Barang dari supplier* (**SKPL-SISIP-010-02**).

Merupakan fungsi yang digunakan untuk mengubah pengecekan data penerimaan barang dari supplier. Jika barang sudah diterima namun belum lengkap maka statusnya 'Barang Diterima Belum Lengkap', jika barang sudah diterima dan lengkap maka statusnya akan diganti menjadi 'Barang Diterima Lengkap'.

11. Fungsi *Pengelolaan Pemesanan Barang ke Pusat atau Antar Cabang* (**SKPL-SISIP-011**)

Merupakan fungsi yang digunakan oleh Gudang untuk melakukan pencatatan pemesanan barang ke gudang pusat atau antar cabang.

Fungsi *Pengelolaan Pemesanan Barang Pusat atau antar cabang* meliputi :

a. Fungsi *Generate Kode Pemesanan Barang Pusat atau Antar Cabang* (**SKPL-SISIP-011-01**).

Program Studi Teknik Informatika	SKPL – SISIP	19/76
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 generate kode pemesanan barang ke pusat atau antar cabang yang baru.

- b. Fungsi *Tambah Pemesanan Barang Pusat atau Antar Cabang (SKPL-SISIP-011-02)*.

Merupakan fungsi yang digunakan untuk menambah data pemesanan barang ke pusat atau antar cabang yang baru.

- c. Fungsi *Batal Pemesanan Barang Pusat atau Antar Cabang (SKPL-SISIP-011-03)*.

Merupakan fungsi yang digunakan untuk membatalkan pemesanan barang ke pusat atau antar cabang.

- d. Fungsi *Tampil Data Pemesanan Barang ke Pusat atau Antar Cabang (SKPL-SISIP-011-04)*

Merupakan fungsi yang digunakan oleh Gudang untuk menampilkan data pemesanan barang ke pusat atau antar cabang.

- e. Fungsi *Tambah Detil Pemesanan Barang ke Pusat atau Antar Cabang (SKPL-SISIP-011-05)*.

Merupakan fungsi yang digunakan untuk menambah data detil pemesanan barang ke pusat atau antar cabang yang baru.

- f. Fungsi *Ubah Detil Pemesanan Barang ke Pusat atau Antar Cabang (SKPL-SISIP-011-06)*.

Merupakan fungsi yang digunakan untuk mengubah data detil pemesanan barang ke pusat atau antar cabang.

Program Studi Teknik Informatika	SKPL – SISIP	20/76
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		

g. Fungsi *Hapus Detil Pemesanan Barang ke Pusat atau Antar Cabang (SKPL-SISIP-011-07)*.

Merupakan fungsi yang digunakan untuk menghapus data detil pemesanan barang ke pusat atau antar cabang.

h. Fungsi *Tampil Detil Pemesanan Barang ke Supplier (SKPL-SISIP-011-08)*.

Merupakan fungsi yang digunakan untuk menampilkan data detil pemesanan barang ke pusat atau antar cabang.

12. Fungsi *Pengelolaan Penerimaan Barang dari Pusat atau Antar Cabang (SKPL-SISIP-012)*

Merupakan fungsi yang digunakan oleh Gudang untuk mengelola data penerimaan barang dari pusat atau antar cabang.

Fungsi Pengelolaan Penerimaan Barang dari supplier meliputi :

a. Fungsi *Tambah Penerimaan Barang dari Pusat atau Antar Cabang (SKPL-SISIP-012-01)*.

Merupakan fungsi yang digunakan untuk menambah data penerimaan barang dari pusat atau antar cabang yang baru.

b. Fungsi *Ubah Detil Penerimaan Barang dari Pusat atau Antar Cabang (SKPL-SISIP-012-02)*.

Merupakan fungsi yang digunakan untuk mengubah pengecekan data penerimaan barang dari pusat atau antar cabang. Jika barang sudah diterima namun belum lengkap maka statusnya 'Barang

Program Studi Teknik Informatika	SKPL – SISIP	21/76
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		

Diterima Belum Lengkap', jika barang sudah diterima dan lengkap maka statusnya akan diganti menjadi 'Barang Diterima Lengkap'.

13. Fungsi *Pengelolaan Transaksi Penjualan* (**SKPL-SISIP-013**)

Merupakan fungsi yang digunakan oleh Kasir untuk melakukan pencatatan transaksi penjualan.

Fungsi *Pengelolaan Transaksi Penjualan* meliputi :

a. Fungsi *Generate No.Faktur Penjualan* (**SKPL-SISIP-013-01**).

Merupakan fungsi yang digunakan untuk *generate no.faktur* dari transaksi penjualan.

b. Fungsi *Tambah Transaksi Penjualan* (**SKPL-SISIP-013-02**).

Merupakan fungsi yang digunakan untuk menambah data transaksi penjualan yang baru.

c. Fungsi *Cetak Nota Transaksi Penjualan* (**SKPL-SISIP-013-03**).

Merupakan fungsi yang digunakan untuk mencetak nota transaksi penjualan.

d. Fungsi *Tampil Transaksi Penjualan* (**SKPL-SISIP-013-04**)

Merupakan fungsi yang digunakan oleh Kasir untuk menampilkan data transaksi penjualan.

e. Fungsi *Tambah Detil Transaksi Penjualan*(**SKPL-SISIP-013-05**).

Program Studi Teknik Informatika	SKPL – SISIP	22/76
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 menambah detail transaksi penjualan.

- f. Fungsi *Ubah Detil Transaksi Penjualan* (**SKPL-SISIP-013-06**).

Merupakan fungsi yang digunakan untuk mengubah data detail penjualan.

- g. Fungsi *Hapus Detil Transaksi Penjualan* (**SKPL-SISIP-013-07**).

Merupakan fungsi yang digunakan untuk menghapus data detail penjualan.

- h. Fungsi *Tampil Detil Transaksi Penjualan* (**SKPL-SISIP-013-08**).

Merupakan fungsi yang digunakan untuk tampil data detail penjualan.

14. Fungsi *Generate Laporan* (**SKPL-SISIP-014**)

Merupakan fungsi yang digunakan oleh manager atau dalam kasus ini adalah KASI PPU untuk men-generate laporan.

Fungsi Generate Laporan meliputi :

- a. Fungsi *Tampil Laporan Grafik dan Tabel Penjualan Berdasarkan Cabang dan per Periode Tahun* (**SKPL-SISIP-014-01**).

Merupakan fungsi yang digunakan untuk menampilkan laporan banyaknya transaksi penjualan yang terjadi berdasarkan cabang per periode tahun.

Program Studi Teknik Informatika	SKPL – SISIP	23/76
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 *Tampil Laporan Grafik dan Tabel Pendapatan Berdasarkan Cabang dan per Periode Tahun* (**SKPL-SISIP-014-02**).

Merupakan fungsi yang digunakan untuk menampilkan laporan jumlah pendapatan berdasarkan cabang per periode tahun.

c. Fungsi *Tampil History Stok Barang* (**SKPL-SISIP-014-02**).

Merupakan fungsi yang digunakan oleh KASI PPU untuk menampilkan data histori dari stok barang apabila terjadi perubahan stok baik saat pemesanan, penerimaan, maupun transaksi penjualan.

d. Fungsi *Tampil History Harga Barang* (**SKPL-SISIP-014-03**).

Merupakan fungsi yang digunakan oleh KASI PPU untuk menampilkan data histori dari harga barang apabila terjadi perubahan harga pada barang tertentu.

15. Fungsi *Pengelolaan Data Barang Supplier* (**SKPL-SISIP-015**)

Merupakan fungsi yang digunakan oleh Administrator untuk mengelola data barang supplier.

Fungsi Pengelolaan Data Supplier meliputi :

a. Fungsi *Tambah Data Barang Supplier* (**SKPL-SISIP-015-01**).

Merupakan fungsi yang digunakan untuk menambah data barang supplier yang baru.

Program Studi Teknik Informatika	SKPL – SISIP	24/76
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 *Ubah Data Barang Supplier* (**SKPL-SISIP-005-02**).

Merupakan fungsi yang digunakan untuk mengubah data barang supplier.

c. Fungsi *Hapus Data Barang Supplier* (**SKPL-SISIP-015-03**).

Merupakan fungsi yang digunakan untuk menghapus data barang supplier.

d. Fungsi *Cari Data Barang Supplier* (**SKPL-SISIP-015-04**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU dan Gudang untuk mencari data barang supplier Bulog Mart berdasarkan lokasinya.

e. Fungsi *Tampil Data Barang Supplier* (**SKPL-SISIP-015-05**)

Merupakan fungsi yang digunakan oleh Administrator, Kasir, KASI PPU, dan Gudang untuk menampilkan data barang supplier dari Bulog Mart.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SISIP adalah sebagai berikut :

1. Memahami pengoperasian Microsoft Windows.
2. Mengerti tentang *internet* dan *web*.
3. Memahami penggunaan aplikasi SISIP.

Program Studi Teknik Informatika	SKPL – SISIP	25/76
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		

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SISIP tersebut adalah :

1. Kebijaksanaan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SISIP.

2. Keterbatasan perangkat keras

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

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada aplikasi *web browser*.

3. Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak SISIP 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 halaman web.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SISIP adalah:

Program Studi Teknik Informatika	SKPL – SISIP	26/76
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. Perangkat *dekstop*.

Komponen	Minimal	Dianjurkan
Processor	Pentium III-class, 1GHz	Pentium 4 1,7 GHz
RAM	256 MB	512 MB
Hard Disk	1 GB	10 GB
Monitor	800x600 256 colors High Color 16-bit	1024x 768 256 colors High Color 32-bit
Mouse	Mouse Microsoft atau alat penunjuk yang compatible dengan OS	Mouse Microsoft atau alat penunjuk yang compatible dengan OS
Keyboard	Keyboard Microsoft atau alat pengetik yang compatible dengan OS	Keyboard Microsoft atau alat pengetik yang compatible dengan OS
Printer	Printer yang Compatible dengan OS	Printer yang Compatible 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.

2. Perangkat *Database Server*.

Komponen	Minimal	Dianjurkan
Memory	512 MB	1GB
Processor Speed	X86 1.0 GHz x64 1.4 GHZ	2.0 GHz
Processor type	x64 Processor: AMD Opteron, AMD	Intel Pentium IV with EM64T support

	Athlon 64, Intel Xeon with Intel EM64T support x86 Processor: Pentium III-compatible	
RAM	2GB	4GB

3. Perangkat Web Server.

CPU	2 core (3GHz or higher)
RAM	4GB
NICs	FrontNet, BackNet

3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SISIP adalah sebagai berikut :

1. Nama : MySQL

Sumber : Oracle

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

2. Nama : Windows XP/Vista/7

Sumber : Microsoft

Sebagai sistem operasi yang digunakan.

3. Nama : Apache

Sumber : Apache Software Foundation

Sebagai aplikasi *web server*.

4. Nama : Code Igniter
Sumber : EllisLab, Inc.
Sebagai *framework* untuk menjalankan aplikasi.

5. Nama : IE/Firefox/Chrome/Opera/dll
Sumber : Microsoft/Mozilla/Google/dll
Sebagai aplikasi *web browser* untuk membuka sistem *web*.

6. Nama : Notepad++
Sumber : Don Ho.
Sebagai aplikasi untuk editor program.

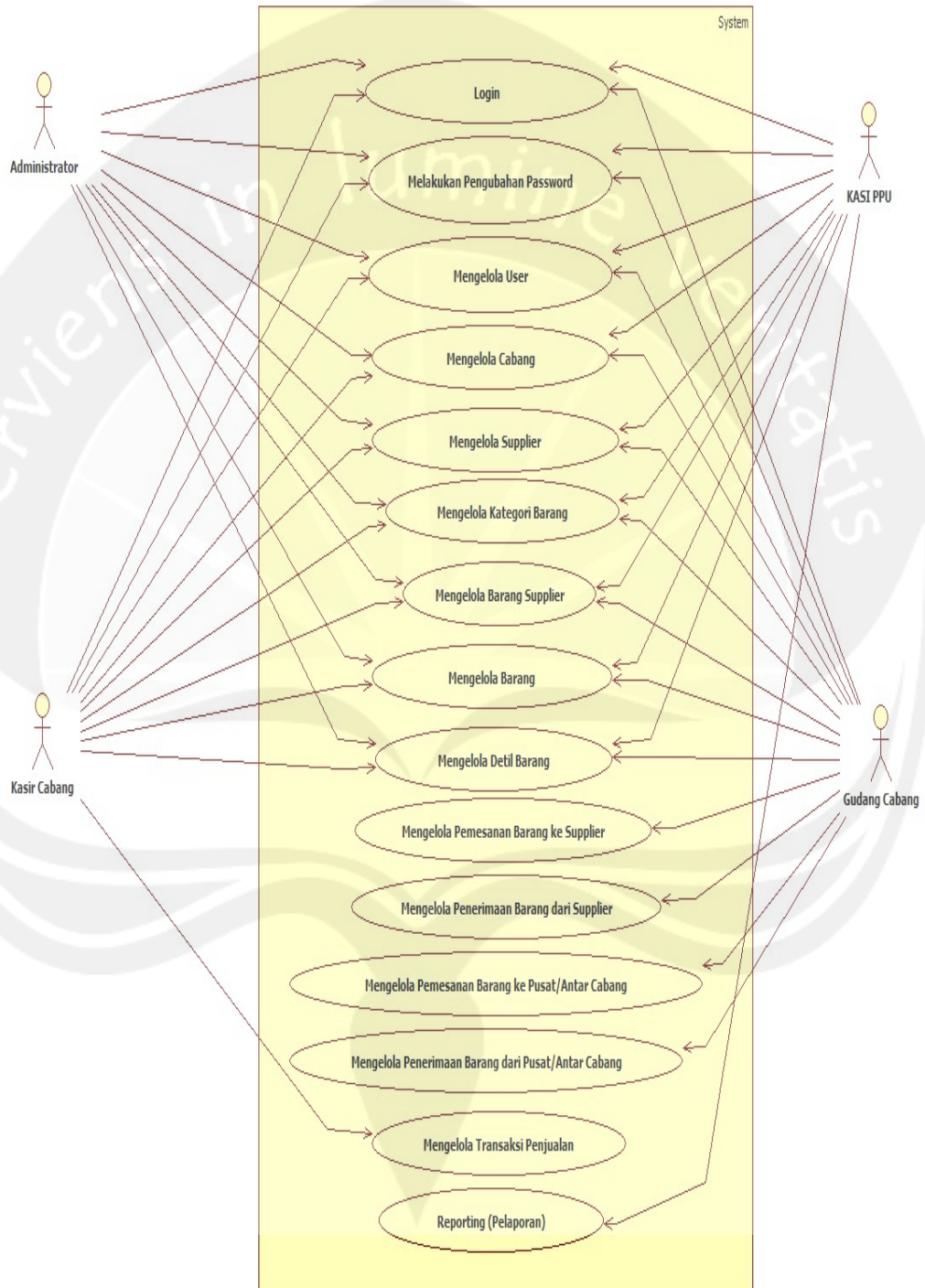
3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SISIP dengan aplikasi *web* menggunakan *protocol* HTTP.

3.2 Kebutuhan fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram

Aplikasi Web



Gambar 2. Use Case Diagram

4. Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

APLIKASI WEB

4.1.1 Use case Spesification : Login

1. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada sebuah id unik yaitu id user dan password yang berupa rangkaian karakter.

2. Primary Actor

1. Administrator
2. KASI PPU
3. Kasir
4. Gudang

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login
2. Sistem menampilkan antarmuka untuk login
3. Aktor memasukkan id dan password
4. Sistem memeriksa id dan password yang diinputkan aktor

E-1 Password atau id user tidak sesuai

5. Sistem memberikan akses ke aktor
6. Use Case ini selesai

5. Alternative Flow

1. None

Program Studi Teknik Informatika	SKPL – SISIP	31/76
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		

6. Error Flow

E-1 Password atau nama user tidak sesuai

1. Sistem menampilkan peringatan bahwa id user atau password tidak sesuai
2. Kembali ke Basic Flow langkah ke 3

7. PreConditions

None

8. PostConditions

Data User telah terupdate di database

4.1.2 Use case Spesification : Ubah Password

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengubah password.

2. Primary Actor

1. Administrator
2. KASI PPU
3. Kasir
4. Gudang

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan perubahan password.
2. Sistem menampilkan antarmuka untuk ubah password.
3. Aktor memasukkan password lama dan password baru
4. Sistem memeriksa apakah password lama dan password baru valid atau tidak

E-1 Password tidak sesuai

Program Studi Teknik Informatika	SKPL – SISIP	32/76
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 menyimpan password baru ke database

6. Use Case ini selesai

5. Alternative Flow

none

6. Error Flow

E-1 Password atau nama user tidak sesuai

1. Sistem menampilkan peringatan bahwa password tidak sesuai
2. Kembali ke Basic Flow langkah ke 3

7. PreConditions

1. Use case Login telah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Password telah diperbaharui di database, dan aktor dapat masuk ke sistem dengan password yang baru.

4.1.3 Use case Spesification : Pengelolaan Data User

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola datauser. *Primary* Aktor dapat melakukan Ubah data user, Tampil data user. Untuk tambah user hanya dapat dilakukan oleh Administrator.

2. Primary Actor

1. KASI PPU
2. Kasir
3. Gudang

3. Supporting Actor

1. Administrator

Program Studi Teknik Informatika	SKPL – SISIP	33/76
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 untuk melakukan pengelolaan data user.
2. Sistem memberikan pilihan untuk melakukan tambah data user (Administrator), ubah data user atau tampil data user.
3. *Supporting* Aktor memilih untuk melakukan Tambah data user
 - A-1 *Primary* Aktor memilih untuk melakukan Ubah data user
 - A-2 *Primary dan Supporting* Aktor memilih untuk melakukan Tampil data user
4. Sistem menampilkan data user.
5. *Supporting* Aktor menginputkan data user yang baru.
6. *Supporting* Aktor mengklik tombol Save untuk menyimpan data user yang telah diinputkan.
7. Sistem mengecek data user yang telah diubah
 - E-1 Data user yang diinputkan aktor kurang lengkap atau salah.
8. Sistem menyimpan data user ke database.
9. Use Case ini selesai.

4. Alternative Flow

- A-1 *Primary* Aktor memilih untuk melakukan Ubah user
 1. Sistem menampilkan data user
 2. *Primary* Aktor mengubah data user dengan inputan baru.
 3. *Primary* Aktor mengklik tombol Save untuk menyimpan perubahan data user ke dalam database.

Program Studi Teknik Informatika	SKPL – SISIP	34/76
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 data user yang telah diubah

E-2 Data user yang diinputkan aktor kurang lengkap atau salah.

5. Sistem menyimpan data user ke database.

6. Berlanjut ke Basic Flow langkah ke-9.

A-2 *Supporting dan Primary* Aktor memilih untuk melakukan Hapus user

1. Sistem menampilkan data pegawai yang ingin dihapus

2. *Primary* Aktor memilih untuk menghapus data pegawai

3. Sistem menampilkan pertanyaan konfirmasi apakah data user ingin dihapus ?

4. *Primary* Aktor mengkonfirmasi ingin menghapus data user

5. Sistem menyimpan data pegawai yang ada ke database

6. Berlanjut ke Basic Flow langkah 9

A-3 *Supporting dan Primary* Aktor memilih untuk melakukan Cari user

1. *Primary* Aktor mengetik data user yang ingin dicari

2. Sistem menampilkan data user yang dicari aktor

3. Berlanjut ke Basic Flow langkah ke 9

A-4 *Supporting dan Primary* Aktor memilih untuk melakukan Tampil user

1. Sistem menampilkan data user

2. Berlanjut ke Basic Flow langkah ke-9.

6. Error Flow

E-1 Data user yang diinputkan kurang lengkap atau salah

1. Sistem menampilkan peringatan bahwa data yang diinputkan kurang lengkap atau salah.
2. Sistem memberi penanda pada field yang salah.
3. Kembali ke Basic Flow langkah ke-5

E-2 Data user yang diinputkan kurang lengkap atau salah

1. Sistem menampilkan peringatan bahwa data yang diinputkan kurang lengkap atau salah.
2. Sistem memberi penanda pada field yang salah.
3. Kembali ke Basic Flow langkah ke-5

7. PreConditions

Aktor telah memasuki sistem

8. PostConditions

Data user di database telah terupdate.

4.1.4 Use case Spesification : Pengelolaan Cabang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data cabang. *Primary* Aktor dapat melakukan tambah data cabang, ubah data cabang, hapus data cabang, dan cari data cabang. Sedangkan *Supporting* Aktor dapat melakukan tampil data cabang.

2. Primary Actor

1. Administrator

Program Studi Teknik Informatika	SKPL – SISIP	36/76
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. Supporting Actor

1. KASI PPU
2. Kasir
3. Gudang

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data cabang.
2. *Primary* Aktor memilih untuk melakukan tambah data cabang
 - A-1 *Primary* Aktor memilih untuk melakukan ubah data cabang
 - A-2 *Primary* Aktor memilih untuk melakukan hapus data cabang
 - A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data cabang
 - A-4 *Primary* dan *Supporting* Aktor memilih untuk melakukan tampil data cabang
3. *Primary* Aktor menginputkan data cabang
4. *Primary* Aktor mengklik tombol Save untuk menyimpan data cabang yang telah diinputkan
5. Sistem mengecek data cabang yang telah diinputkan
 - E-1 Data data cabang yang diinputkan aktor salah atau kurang lengkap
6. Sistem menyimpan data cabang ke database
7. Use Case selesai

5. Alternative Flow

- A-1 *Primary* Aktor memilih untuk melakukan ubah data cabang
 1. Sistem menampilkan data cabang

Program Studi Teknik Informatika	SKPL – SISIP	37/76
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. *Primary* Aktor mengubah data cabang yang sudah ditampilkan

3. *Primary* Aktor meminta sistem untuk menyimpan data cabang yang telah diubah

4. Sistem melakukan pengecekan terhadap data cabang yang telah diubah

E-2 data cabang yang telah diubah salah atau kurang lengkap

5. Sistem menyimpan data cabang yang telah diubah ke database

6. Berlanjut ke Basic Flow langkah ke 7

A-2 Aktor memilih untuk melakukan hapus data cabang

1. Sistem menampilkan data cabang yang ingin dihapus

2. *Primary* Aktor memilih untuk menghapus data cabang

3. Sistem menampilkan pertanyaan konfirmasi apakah data cabang ingin dihapus ?

4. *Primary* Aktor mengkonfirmasi ingin menghapus data cabang

5. Sistem menyimpan data cabang yang ada ke database

6. Berlanjut ke Basic Flow langkah 7

A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data cabang

1. *Primary* dan *Supporting* Aktor mengetik data cabang yang ingin dicari

2. Sistem menampilkan data cabang yang dicari aktor

3. Berlanjut ke Basic Flow langkah ke-7

Program Studi Teknik Informatika	SKPL – SISIP	38/76
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-4 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data cabang

1. Sistem menampilkan data cabang
2. Berlanjut ke Basic Flow langkah ke-7

6. Error Flow

E-1 data cabang yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

E-2 data cabang yang diinputkan aktor salah

4. Sistem menampilkan pesan peringatan bahwa data yang diubah salah atau kurang lengkap
5. Sistem memberi penanda pada field yang salah
6. Kembali ke Alternative Flow A-1 Langkah ke 2

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data cabang di database telah terupdate

4.1.5 Use case Spesification : Pengelolaan Supplier

1. Brief Description

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

Program Studi Teknik Informatika	SKPL – SISIP	39/76
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		

cari data supplier. Sedangkan *Supporting* Aktor dapat melakukan tampil data supplier.

2. Primary Actor

1. Administrator

3. Supporting Actor

1. KASI PPU
2. Kasir
3. Gudang

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data supplier.
2. *Primary* Aktor memilih untuk melakukan tambah data supplier
 - A-1 *Primary* Aktor memilih untuk melakukan ubah data supplier
 - A-2 *Primary* Aktor memilih untuk melakukan hapus data supplier
 - A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data supplier
 - A-4 *Primary* dan *Supporting* Aktor memilih untuk melakukan tampil data supplier
3. *Primary* Aktor menginputkan data supplier
4. *Primary* Aktor mengklik tombol Save untuk menyimpan data supplier yang telah diinputkan
5. Sistem mengecek data supplier yang telah diinputkan
 - E-1 Data data supplier yang diinputkan aktor salah atau kurang lengkap
6. Sistem menyimpan data cabang ke database
7. Use Case selesai

Program Studi Teknik Informatika	SKPL – SISIP	40/76
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 *Primary* Aktor memilih untuk melakukan ubah data supplier

1. Sistem menampilkan data supplier
2. *Primary* Aktor mengubah data supplier yang sudah ditampilkan
3. *Primary* Aktor meminta sistem untuk menyimpan data supplier yang telah diubah
4. Sistem melakukan pengecekan terhadap data supplier yang telah diubah

E-2 data supplier yang telah diubah salah atau kurang lengkap

5. Sistem menyimpan data supplier yang telah diubah ke database
 6. Berlanjut ke Basic Flow langkah ke 7
- A-2 Aktor memilih untuk melakukan hapus data supplier

1. Sistem menampilkan data supplier yang ingin dihapus
2. *Primary* Aktor memilih untuk menghapus data supplier
3. Sistem menampilkan pertanyaan konfirmasi apakah data supplier ingin dihapus ?
4. *Primary* Aktor mengkonfirmasi ingin menghapus data supplier
5. Sistem menyimpan data supplier yang ada ke database
6. Berlanjut ke Basic Flow langkah 7

A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data supplier

1. *Primary* dan *Supporting* Aktor mengetik data supplier yang ingin dicari
 2. Sistem menampilkan data supplier yang dicari aktor
 3. Berlanjut ke Basic Flow langkah ke-7
- A-4 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data supplier
1. Sistem menampilkan data supplier
 2. Berlanjut ke Basic Flow langkah ke-7

6. Error Flow

E-1 data supplier yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

E-2 data supplier yang diinputkan aktor salah

1. Sistem menampilkan pesan peringatan bahwa data yang diubah salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Alternative Flow A-1 Langkah ke 2

9. PreConditions

1. Aktor telah memasuki sistem

10. PostConditions

1. Data supplier di database telah terupdate

Program Studi Teknik Informatika	SKPL – SISIP	42/76
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		

4.1.6 Use case Spesification : Pengelolaan Kategori Barang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data kategori barang. *Primary* Aktor dapat melakukan tambah data kategori barang, ubah data kategori barang, dan hapus data kategori barang. Sedangkan untuk tampil data dapat dilakukan oleh *Primary* dan *Supporting* aktor.

2. Primary Actor

1. Administrator

3. Supporting Actor

1. KASI PPU
2. Kasir
3. Gudang

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data kategori barang.
2. *Primary* Aktor memilih untuk melakukan tambah data kategori barang
 - A-1 *Primary* Aktor memilih untuk melakukan ubah data kategori barang
 - A-2 *Primary* Aktor memilih untuk melakukan hapus data kategori barang
 - A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data kategori barang
 - A-4 *Primary* dan *Supporting* Aktor memilih untuk melakukan tampil data kategori barang
3. *Primary* Aktor menginputkan data kategori barang

Program Studi Teknik Informatika	SKPL – SISIP	43/76
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. *Primary* Aktor mengklik tombol Save untuk menyimpan data kategori barang yang telah diinputkan
5. Sistem mengecek data kategori barang yang telah diinputkan
 - E-1 Data data kategori barang yang diinputkan aktor salah atau kurang lengkap
6. Sistem menyimpan data kategori barang ke database
7. Use Case selesai

5. Alternative Flow

A-1 *Primary* Aktor memilih untuk melakukan ubah data kategori barang

1. Sistem menampilkan data kategori barang
2. *Primary* Aktor mengubah data kategori barang yang sudah ditampilkan
3. *Primary* Aktor meminta sistem untuk menyimpan data kategori barang yang telah diubah
4. Sistem melakukan pengecekan terhadap data kategori barang yang telah diubah

E-2 data kategori barang yang telah diubah salah atau kurang lengkap

5. Sistem menyimpan data kategori barang yang telah diubah ke database
6. Berlanjut ke Basic Flow langkah ke 7

A-2 *Primary* Aktor memilih untuk melakukan hapus data kategori barang

1. Sistem menampilkan data kategori barang yang ingin dihapus

2. *Primary* Aktor memilih untuk menghapus data kategori barang
3. Sistem menampilkan pertanyaan konfirmasi apakah data kategori barang ingin dihapus ?
4. *Primary* Aktor mengkonfirmasi ingin menghapus data kategori barang
5. Sistem menyimpan data kategori barang yang ada ke database
6. Berlanjut ke Basic Flow langkah 7

A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data kategori barang

1. *Primary* dan *Supporting* Aktor mengetik data kategori barang yang ingin dicari
2. Sistem menampilkan data kategori barang yang dicari aktor
3. Berlanjut ke Basic Flow langkah ke-7

A-4 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data kategori barang

1. Sistem menampilkan data kategori barang
2. Berlanjut ke Basic Flow langkah ke-7

6. Error Flow

E-1 data kategori barang yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

E-2 data kategori barang yang diinputkan aktor salah

4. Sistem menampilkan pesan peringatan bahwa data yang diubah salah atau kurang lengkap
5. Sistem memberi penanda pada field yang salah
6. Kembali ke Alternative Flow A-1 Langkah ke 2

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data kategori barang di database telah terupdate

4.1.7 Use case Spesification : Pengelolaan Barang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data barang. *Primary* Aktor dapat melakukan tambah data barang, ubah data barang, dan hapus data barang. *Primary* dan *Supporting* aktor dapat melakukan tampil data barang.

2. Primary Actor

1. Administrator
2. Gudang

3. Supporting Actor

1. KASI PPU
2. Kasir

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data barang.
2. *Primary* Aktor memilih untuk melakukan tambah data barang

Program Studi Teknik Informatika	SKPL – SISIP	46/76
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 *Primary* Aktor memilih untuk melakukan ubah data barang

A-2 *Primary* Aktor memilih untuk melakukan hapus data barang

A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data barang

A-4 *Primary* dan *Supporting* Aktor memilih untuk melakukan tampil data barang

3. *Primary* Aktor menginputkan data barang
4. *Primary* Aktor mengklik tombol Save untuk menyimpan data barang yang telah diinputkan
5. Sistem mengecek data barang yang telah diinputkan

E-1 Data data barang yang diinputkan aktor salah atau kurang lengkap

6. Sistem menyimpan data barang ke database
7. Use Case selesai

5. Alternative Flow

A-1 *Primary* Aktor memilih untuk melakukan ubah data barang

1. Sistem menampilkan data barang
2. *Primary* Aktor mengubah data barang yang sudah ditampilkan
3. *Primary* Aktor meminta sistem untuk menyimpan data barang yang telah diubah
4. Sistem melakukan pengecekan terhadap data barang yang telah diubah

E-2 data barang yang telah diubah salah atau kurang lengkap

5. Sistem menyimpan data barang yang telah diubah ke database

Program Studi Teknik Informatika	SKPL – SISIP	47/76
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		

6. Berlanjut ke Basic Flow langkah ke 7

A-2 *Primary* Aktor memilih untuk melakukan hapus data barang

1. Sistem menampilkan data barang yang ingin dihapus
2. *Primary* Aktor memilih untuk menghapus data barang
3. Sistem menampilkan pertanyaan konfirmasi apakah data barang ingin dihapus ?
4. *Primary* Aktor mengkonfirmasi ingin menghapus data barang
5. Sistem menyimpan data barang yang ada ke database
6. Berlanjut ke Basic Flow langkah 7

A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data barang

1. *Primary* dan *Supporting* Aktor mengetik data barang yang ingin dicari
2. Sistem menampilkan data barang yang dicari aktor
3. Berlanjut ke Basic Flow langkah ke-7

A-4 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data barang

1. Sistem menampilkan data barang
2. Berlanjut ke Basic Flow langkah ke-7

6. Error Flow

E-1 data barang yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap

Program Studi Teknik Informatika	SKPL – SISIP	48/76
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 memberi penanda pada field yang salah

3. Kembali ke Basic Flow Langkah ke 3

E-2 data barang yang diinputkan aktor salah

4. Sistem menampilkan pesan peringatan bahwa data yang diubah salah atau kurang lengkap

5. Sistem memberi penanda pada field yang salah

6. Kembali ke Alternative Flow A-1 Langkah ke 2

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data barang di database telah terupdate

4.1.8 Use case Spesification : Pengelolaan Detil Barang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data detil barang. *Primary* Aktor dapat melakukan tambah data detil barang, ubah data detil barang, dan hapus data detil barang. *Primary* dan *Supporting* aktor dapat melakukan tampil data detil barang.

2. Primary Actor

1. Administrator
2. Gudang

3. Supporting Actor

1. KASI PPU
2. Kasir

4. Basic Flow

Program Studi Teknik Informatika	SKPL – SISIP	49/76
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. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data detail barang.
2. *Primary* Aktor memilih untuk melakukan tambah data detail barang
 - A-1 *Primary* Aktor memilih untuk melakukan ubah data detail barang
 - A-2 *Primary* Aktor memilih untuk melakukan hapus data detail barang
 - A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data detail barang
 - A-4 *Primary* dan *Supporting* Aktor memilih untuk melakukan tampil data detail barang
3. *Primary* Aktor menginputkan data detail barang
4. *Primary* Aktor mengklik tombol Save untuk menyimpan data detail barang yang telah diinputkan
5. Sistem mengecek data detail barang yang telah diinputkan
 - E-1 Data data detail barang yang diinputkan aktor salah atau kurang lengkap
6. Sistem menyimpan data detail barang ke database
7. Use Case selesai

5. Alternative Flow

- A-1 *Primary* Aktor memilih untuk melakukan ubah data detail barang
1. Sistem menampilkan data detail barang
 2. *Primary* Aktor mengubah data detail barang yang sudah ditampilkan

Program Studi Teknik Informatika	SKPL – SISIP	50/76
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. *Primary* Aktor meminta sistem untuk menyimpan data detil barang yang telah diubah
4. Sistem melakukan pengecekan terhadap data detil barang yang telah diubah

E-2 data detil barang yang telah diubah salah atau kurang lengkap

5. Sistem menyimpan data detil barang yang telah diubah ke database
6. Berlanjut ke Basic Flow langkah ke 7

A-2 *Primary* Aktor memilih untuk melakukan hapus data detil barang

1. Sistem menampilkan data detil barang yang ingin dihapus
2. *Primary* Aktor memilih untuk menghapus data detil barang
3. Sistem menampilkan pertanyaan konfirmasi apakah data detil barang ingin dihapus ?
4. *Primary* Aktor mengkonfirmasi ingin menghapus data detil barang
5. Sistem menyimpan data detil barang yang ada ke database
6. Berlanjut ke Basic Flow langkah 7

A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data detil barang

1. *Primary* dan *Supporting* Aktor mengetik data detil barang yang ingin dicari
2. Sistem menampilkan data detil barang yang dicari aktor
3. Berlanjut ke Basic Flow langkah ke-7

A-4 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data detail barang

1. Sistem menampilkan data detail barang
2. Berlanjut ke Basic Flow langkah ke-7

6. Error Flow

E-1 data detail barang yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

E-2 data barang yang diinputkan aktor salah

4. Sistem menampilkan pesan peringatan bahwa data yang diubah salah atau kurang lengkap
5. Sistem memberi penanda pada field yang salah
6. Kembali ke Alternative Flow A-1 Langkah ke 2

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data detail barang di database telah terupdate

4.1.9 Use case Spesification : Pengelolaan Barang Supplier

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data barang supplier. *Primary* Aktor dapat melakukan tambah data barang supplier, ubah data barang supplier,

Program Studi Teknik Informatika	SKPL – SISIP	52/76
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		

dan hapus data barang supplier. *Primary* dan *Supporting* aktor dapat melakukan tampil data barang supplier.

2. Primary Actor

1. Administrator

3. Supporting Actor

1. KASI PPU
2. Kasir
3. Gudang

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data barang supplier.
2. *Primary* Aktor memilih untuk melakukan tambah data barang supplier
 - A-1 *Primary* Aktor memilih untuk melakukan ubah data barang supplier
 - A-2 *Primary* Aktor memilih untuk melakukan hapus data barang supplier
 - A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data barang supplier
 - A-4 *Primary* dan *Supporting* Aktor memilih untuk melakukan tampil data barang supplier
3. *Primary* Aktor menginputkan data barang supplier
4. *Primary* Aktor mengklik tombol Save untuk menyimpan data barang supplier yang telah diinputkan
5. Sistem mengecek data barang supplier yang telah diinputkan

- E-1 Data data barang supplier yang diinputkan aktor salah atau kurang lengkap
6. Sistem menyimpan data barang supplier ke database
 7. Use Case selesai

5. Alternative Flow

A-1 *Primary* Aktor memilih untuk melakukan ubah data barang supplier

1. Sistem menampilkan data barang supplier
2. *Primary* Aktor mengubah data barang supplier yang sudah ditampilkan
3. *Primary* Aktor meminta sistem untuk menyimpan data barang supplier yang telah diubah
4. Sistem melakukan pengecekan terhadap data barang supplier yang telah diubah

E-2 data barang supplier yang telah diubah salah atau kurang lengkap

5. Sistem menyimpan data barang supplier yang telah diubah ke database
6. Berlanjut ke Basic Flow langkah ke 7

A-2 *Primary* Aktor memilih untuk melakukan hapus data barang supplier

1. Sistem menampilkan data barang supplier yang ingin dihapus
2. *Primary* Aktor memilih untuk menghapus data barang supplier
3. Sistem menampilkan pertanyaan konfirmasi apakah data barang supplier ingin dihapus ?

4. *Primary* Aktor mengkonfirmasi ingin menghapus data barang supplier
 5. Sistem menyimpan data barang supplier yang ada ke database
 6. Berlanjut ke Basic Flow langkah 7
- A-3 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data barang supplier
1. *Primary* dan *Supporting* Aktor mengetik data barang supplier yang ingin dicari
 2. Sistem menampilkan data barang supplier yang dicari aktor
 3. Berlanjut ke Basic Flow langkah ke-7
- A-4 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data barang supplier
1. Sistem menampilkan data barang supplier
 2. Berlanjut ke Basic Flow langkah ke-7

6. Error Flow

- E-1 data barang supplier yang diinputkan aktor salah atau kurang lengkap
1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
 2. Sistem memberi penanda pada field yang salah
 3. Kembali ke Basic Flow Langkah ke 3
- E-2 data barang yang diinputkan aktor salah
4. Sistem menampilkan pesan peringatan bahwa data yang diubah salah atau kurang lengkap
 5. Sistem memberi penanda pada field yang salah

6. Kembali ke Alternative Flow A-1 Langkah ke
2

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data barang supplier di database telah terupdate

4.1.10 Use case Spesification : Pengelolaan
Pemesanan Barang ke Supplier

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola pemesanan barang ke supplier. *Primary* Aktor dapat melakukan tambah data barang, dan membatalkan pemesanan barang, cari dan tampil data pemesanan barang ke supplier.

2. Primary Actor

1. Gudang

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data pemesanan barang ke supplier.
2. *Primary* Aktor memilih untuk melakukan tambah pemesanan barang ke supplier
 - A-1 *Primary* Aktor memilih untuk membatalkan pemesanan barang
 - A-2 *Primary* dan *Supporting* Aktor memilih untuk cari pemesanan barang
 - A-3 *Primary* dan *Supporting* Aktor memilih tampil menampilkan data pemesanan barang

Program Studi Teknik Informatika	SKPL – SISIP	56/76
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-4 *Primary* Aktor memilih untuk melakukan tambah detail pemesanan barang

A-5 *Primary* Aktor memilih untuk melakukan ubah detail pemesanan barang

A-6 *Primary* Aktor memilih untuk melakukan hapus detail pemesanan barang

A-7 *Primary* Aktor memilih untuk menampilkan detail pemesanan barang

3. Sistem melakukan *generate* kode dan tanggal pemesanan secara otomatis

4. *Primary* Aktor menginputkan pemesanan barang yang baru

5. *Primary* Aktor mengklik tombol Save untuk menyimpan data pemesanan barang yang telah diinputkan

6. Sistem mengecek data pemesanan barang yang telah diinputkan

E-1 Data barang yang diinputkan aktor salah atau kurang lengkap

7. Sistem menyimpan data pemesanan barang ke database dengan status pesan yaitu "Ter kirim ke Supplier"

8. Sistem menampilkan dokumen pemesanan untuk supplier dalam bentuk pdf

9. Use Case selesai

5. Alternative Flow

A-1 *Primary* Aktor memilih untuk membatalkan pemesanan barang

1. Sistem menampilkan data pemesanan barang

2. *Primary* Aktor membatalkan data pemesanan barang

Program Studi Teknik Informatika	SKPL – SISIP	57/76
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 menghapus data pemesanan dan detail pemesanan dalam database

4. Berlanjut ke Basic Flow langkah ke-9

A-2 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data pemesanan barang

1. *Primary* dan *Supporting* Aktor mengetik data pemesanan barang yang ingin dicari

2. Sistem menampilkan data pemesanan barang yang dicari aktor

3. Berlanjut ke Basic Flow langkah ke-9

A-3 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data pemesanan barang

1. Sistem menampilkan data pemesanan barang

2. Berlanjut ke Basic Flow langkah ke-9

A-4 *Primary* Aktor memilih untuk melakukan tambah detail data pemesanan barang

1. Aktor menginputkan detail pemesanan barang

2. Aktor mengklik tombol Save untuk menyimpan detail pemesanan barang yang telah diinputkan

3. Sistem mengecek detail pemesanan barang yang telah diinputkan

E-2 Data detail pemesanan barang yang diinputkan aktor salah atau kurang lengkap

4. Sistem menyimpan detail pemesanan barang ke database

5. Berlanjut ke Basic Flow langkah ke-9

A-5 *Primary* Aktor memilih untuk melakukan ubahdata detail pemesanan barang

1. Sistem menampilkan detail pemesanan barang

2. Aktor mengubah detail pemesanan barang yang sudah ditampilkan
 3. Aktor meminta sistem untuk menyimpan detail pemesanan barang yang telah diubah
 4. Sistem melakukan pengecekan terhadap detail pemesanan barang yang telah diubah
 - E-3 detail pemesanan barang yang telah diubah salah atau kurang lengkap.
 5. Sistem menyimpan detail pemesanan barang yang telah diubah ke database
 6. Berlanjut ke Basic Flow langkah ke-9
- A-6 *Primary* Aktor memilih untuk melakukan hapusdata detail pemesanan barang
1. Sistem menampilkan detail pemesanan barang yang ingin dihapus
 2. Aktor memilih untuk menghapus detail pemesanan barang
 3. Sistem menampilkan pertanyaan konfirmasi apakah detail pemesanan barang ingin dihapus ?
 4. Aktor mengkonfirmasi ingin menghapus detail pemesanan barang
 5. Sistem menyimpan detail pemesanan barang yang ada ke database
 6. Berlanjut ke Basic Flow langkah ke-9
- A-7 Aktor memilih untuk menampilkan detail pemesanan barang
1. Sistem menampilkan detail pemesanan barang
 2. Berlanjut ke Basic Flow langkah ke-9

6. Error Flow

Program Studi Teknik Informatika	SKPL – SISIP	59/76
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 barang yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

E-2 detail pemesanan barang yang telah diubah salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

E-3 detail pemesanan barang yang telah diubah salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

2. PreConditions

1. Aktor telah memasuki system

3. PostConditions

1. Data pemesanan barang di database telah terupdate

Program Studi Teknik Informatika	SKPL – SISIP	60/76
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		

4.1.11 Use case Spesification : Pengelolaan Penerimaan Barang dari Supplier

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data penerimaan barang dari supplier. Aktor dapat melakukan tambah data penerimaan barang, dan ubah status penerimaan barang.

2. Primary Actor

1. Gudang

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan penerimaan barang.
2. Sistem menampilkan daftar pemesanan barang
3. Aktor memilih pemesanan barang berdasarkan kode pesan yang diterima
4. Sistem menampilkan detail pemesanan barang
5. Aktor memilih untuk melakukan pengecekan kuantitas barang yang diterima
A-1 Aktor memilih untuk melakukan ubah detail penerimaan barang dari supplier
6. Aktor mengklik tombol Save untuk menyimpan data penerimaan barang yang telah diinputkan
7. Sistem mengecek data penerimaan barang yang telah diinputkan, jika masih ada kekurangan pada kuantitas barang yang diterima maka sistem akan membuat status "Barang Diterima Belum Lengkap", jika tidak ada kekurangan kuantitas terima barang maka sistem akan

membuat status terima menjadi "Barang Diterima Lengkap"

8. Sistem menyimpan data penerimaan barang dari supplier ke database
9. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan ubah detil penerimaan barang dari supplier

1. Sistem menampilkan data penerimaan barang
2. Aktor memilih data pemesanan barang berdasarkan kodenya
3. Sistem menampilkan detil pemesanan barang dari supplier berdasarkan kode pemesanan
4. Aktor menginputkan jumlah kekurangan atau mencek *checkbox* lengkap
6. Sistem menyimpan data detil penerimaan barang yang telah diubah ke database
7. Sistem akan membuat status "Barang diterima Lengkap" jika tidak ada kekurangan, jika masih ada maka statusnya "Barang diterima belum Lengkap"
7. Berlanjut ke Basic Flow langkah ke-9

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data Penerimaan barang di database telah terupdate

Program Studi Teknik Informatika	SKPL – SISIP	62/76
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		

4.1.12 Use case Spesification : Pengelolaan Pemesanan Barang ke Pusat atau Antar Cabang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola pemesanan barang ke pusat atau antar cabang. *Primary* Aktor dapat melakukan tambah data barang, dan membatalkan pemesanan barang, cari dan tampil data pemesanan barang ke pusat atau antar cabang.

2. Primary Actor

1. Gudang

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data pemesanan barang ke pusat atau antar cabang.
2. *Primary* Aktor memilih untuk melakukan tambah pemesanan barang ke pusat atau antar cabang
 - A-1 *Primary* Aktor memilih untuk membatalkan pemesanan barang
 - A-2 *Primary* dan *Supporting* Aktor memilih untuk cari pemesanan barang
 - A-3 *Primary* dan *Supporting* Aktor memilih tampil menampilkan data pemesanan barang
 - A-4 *Primary* Aktor memilih untuk melakukan tambah detail pemesanan barang
 - A-5 *Primary* Aktor memilih untuk melakukan ubah detail pemesanan barang
 - A-6 *Primary* Aktor memilih untuk melakukan hapus detail pemesanan barang

Program Studi Teknik Informatika	SKPL – SISIP	63/76
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-7 *Primary* Aktor memilih untuk menampilkan detail pemesanan barang
3. Sistem melakukan *generate* kode dan tanggal pemesanan secara otomatis
 4. *Primary* Aktor menginputkan pemesanan barang yang baru
 5. *Primary* Aktor mengklik tombol *Save* untuk menyimpan data pemesanan barang yang telah diinputkan
 6. Sistem mengecek data pemesanan barang yang telah diinputkan
 - E-1 Data barang yang diinputkan aktor salah atau kurang lengkap
 7. Sistem menyimpan data pemesanan barang ke database dengan status pesan yaitu "Ter kirim ke Cabang"
 8. Use Case selesai

5. Alternative Flow

- A-1 *Primary* Aktor memilih untuk membatalkan pemesanan barang
1. Sistem menampilkan data pemesanan barang
 2. *Primary* Aktor membatalkan data pemesanan barang
 3. Sistem menghapus data pemesanan dan detail pemesanan dalam database
 4. Berlanjut ke Basic Flow langkah ke-8
- A-2 *Primary* dan *Supporting* Aktor memilih untuk melakukan cari data pemesanan barang
1. *Primary* dan *Supporting* Aktor mengetik data pemesanan barang yang ingin dicari

2. Sistem menampilkan data pemesanan barang yang dicari aktor

3. Berlanjut ke Basic Flow langkah ke-8

A-3 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data pemesanan barang

1. Sistem menampilkan data pemesanan barang

2. Berlanjut ke Basic Flow langkah ke-8

A-4 *Primary* Aktor memilih untuk melakukan tambah detail data pemesanan barang

1. Aktor menginputkan detail pemesanan barang

2. Aktor mengklik tombol Save untuk menyimpan detail pemesanan barang yang telah diinputkan

3. Sistem mengecek detail pemesanan barang yang telah diinputkan

E-2 Data detail pemesanan barang yang diinputkan aktor salah atau kurang lengkap

4. Sistem menyimpan detail pemesanan barang ke database

5. Berlanjut ke Basic Flow langkah ke-8

A-5 *Primary* Aktor memilih untuk melakukan ubah data detail pemesanan barang

1. Sistem menampilkan detail pemesanan barang

2. Aktor mengubah detail pemesanan barang yang sudah ditampilkan

3. Aktor meminta sistem untuk menyimpan detail pemesanan barang yang telah diubah

4. Sistem melakukan pengecekan terhadap detail pemesanan barang yang telah diubah

E-3 detail pemesanan barang yang telah diubah salah atau kurang lengkap.

5. Sistem menyimpan detail pemesanan barang yang telah diubah ke database
 6. Berlanjut ke Basic Flow langkah ke-8
- A-6 *Primary* Aktor memilih untuk melakukan hapusdata detail pemesanan barang
1. Sistem menampilkan detail pemesanan barang yang ingin dihapus
 2. Aktor memilih untuk menghapus detail pemesanan barang
 3. Sistem menampilkan pertanyaan konfirmasi apakah detail pemesanan barang ingin dihapus ?
 4. Aktor mengkonfirmasi ingin menghapus detail pemesanan barang
 5. Sistem menyimpan detail pemesanan barang yang ada ke database
 6. Berlanjut ke Basic Flow langkah ke-8
- A-7 Aktor memilih untuk menampilkan detail pemesanan barang
1. Sistem menampilkan detail pemesanan barang
 2. Berlanjut ke Basic Flow langkah ke-8

6. Error Flow

- E-1 data barang yang diinputkan aktor salah atau kurang lengkap
1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
 2. Sistem memberi penanda pada field yang salah

Program Studi Teknik Informatika	SKPL – SISIP	66/76
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. Kembali ke Basic Flow Langkah ke 3
E-2detil pemesanan barang yang telah diubah salah atau kurang lengkap

2. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap

3. Sistem memberi penanda pada field yang salah

4. Kembali ke Basic Flow Langkah ke 3
E-3detil pemesanan barang yang telah diubah salah atau kurang lengkap

4. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap

5. Sistem memberi penanda pada field yang salah

6. Kembali ke Basic Flow Langkah ke 3

7. PreConditions

1. Aktor telah memasuki system

8. PostConditions

1. Data pemesanan barang di database telah terupdate

4.1.13 Use case Spesification : Pengelolaan Penerimaan Barang dari Pusat atau Antar Cabang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data penerimaan barang dari pusat atau antar cabang. Aktor dapat melakukan tambah data penerimaan barang, dan ubah status penerimaan barang.

Program Studi Teknik Informatika	SKPL – SISIP	67/76
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. Primary Actor

1. Gudang

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan penerimaan barang.
2. Sistem menampilkan daftar pemesanan barang
3. Aktor memilih pemesanan barang berdasarkan kode pesan yang diterima
4. Sistem menampilkan detail pemesanan barang
5. Aktor memilih untuk melakukan pengecekan kuantitas barang yang diterima
 - A-1 Aktor memilih untuk melakukan ubah detail penerimaan barang dari pusat atau antar cabang
6. Aktor mengklik tombol Save untuk menyimpan data penerimaan barang yang telah diinputkan
7. Sistem mengecek data penerimaan barang yang telah diinputkan, jika masih ada kekurangan pada kuantitas barang yang diterima maka sistem akan membuat status "Barang Diterima Belum Lengkap", jika tidak ada kekurangan kuantitas terima barang maka sistem akan membuat status terima menjadi "Barang Diterima Lengkap"
8. Sistem menyimpan data penerimaan barang dari supplier ke database
9. Use Case selesai

5. Alternative Flow

Program Studi Teknik Informatika	SKPL – SISIP	68/76
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 melakukan ubah detail penerimaan barang dari pusat atau antar cabang

1. Sistem menampilkan data penerimaan barang
2. Aktor memilih data pemesanan barang berdasarkan kodenya
3. Sistem menampilkan detail pemesanan barang dari supplier berdasarkan kode pemesanan
4. Aktor menginputkan jumlah kekurangan atau mencek *checkbox* lengkap
6. Sistem menyimpan data detail penerimaan barang yang telah diubah ke database
7. Sistem akan membuat status "Barang diterima Lengkap" jika tidak ada kekurangan, jika masih ada maka statusnya "Barang diterima belum Lengkap"
7. Berlanjut ke Basic Flow langkah ke-9

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Data Penerimaan barang di database telah terupdate

4.1.14 Use case Spesification : Pengelolaan Transaksi Penjualan

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola transaksi penjualan. *Primary* Aktor dapat melakukan tambah data transaksi penjualan, dan cetak nota

Program Studi Teknik Informatika	SKPL – SISIP	69/76
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		

transaksi penjualan. *Primary* dan *Supporting* aktor dapat menampilkan data transaksi penjualan.

2. Primary Actor

1. Kasir

3. Supporting Actor

1. Administrator
2. KASI PPU

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data transaksi penjualan.
2. *Primary* Aktor memilih untuk melakukan tambahtransaksi penjualan
 - A-1 *Primary* dan *Supporting* Aktor memilih untuk melakukan tampil transaksi penjualan
 - A-2 *Primary* Aktor memilih untuk melakukan tambah detail transaksi penjualan
 - A-3 *Primary* Aktor memilih untuk melakukan ubah detail transaksi penjualan
 - A-4 *Primary* Aktor memilih untuk melakukan hapus detail transaksi penjualan
 - A-5 *Primary* Aktor memilih untuk melakukan tampil detail transaksi penjualan
 - A-6 *Primary* Aktor memilih untuk melakukan cetak nota transaksi penjualan
3. Sistem melakukan *generate* kode dan tanggal transaksi penjualan secara otomatis
4. Aktor menginputkan data transaksi penjualan yang baru
5. Sistem mengecek transaksi penjualan yang telah diinputkan

Program Studi Teknik Informatika	SKPL – SISIP	70/76
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 transaksi penjualan yang diinputkan aktor salah atau kurang lengkap
6. Sistem menyimpan data transaksi penjualan ke database
 7. Use Case selesai

5. Alternative Flow

A-1 *Primary* dan *Supporting* Aktor memilih untuk menampilkan data transaksi penjualan

1. Sistem menampilkan data transaksi penjualan
2. Berlanjut ke Basic Flow langkah ke-7

A-2 *Primary* Aktor memilih untuk melakukan tambah detail transaksi penjualan

1. Aktor menginputkan detail transaksi penjualan
2. Aktor mengklik tombol tambah untuk menyimpan detail transaksi penjualan yang telah diinputkan
3. Sistem mengecek detail transaksi penjualan yang telah diinputkan

E-2 Data detail transaksi penjualan yang diinputkan aktor salah atau kurang lengkap

4. Sistem menyimpan detail transaksi penjualan ke database
5. Berlanjut ke Basic Flow langkah ke-7

A-3 *Primary* Aktor memilih untuk melakukan ubah detail transaksi penjualan

1. Sistem menampilkan detail transaksi penjualan
2. Aktor mengubah detail transaksi penjualan yang sudah ditampilkan

Program Studi Teknik Informatika	SKPL – SISIP	71/76
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 meminta sistem untuk menyimpan detail transaksi penjualan yang telah diubah
4. Sistem melakukan pengecekan terhadap detail transaksi penjualan yang telah diubah
- E-3 detail transaksi penjualan yang telah diubah salah atau kurang lengkap
5. Sistem menyimpan detail transaksi penjualan yang telah diubah ke database
6. Berlanjut ke Basic Flow langkah ke-7

A-4 *Primary* Aktor memilih untuk melakukan hapus detail transaksi penjualan

1. Sistem menampilkan detail transaksi penjualan yang ingin dihapus
2. Aktor memilih untuk menghapus detail transaksi penjualan
3. Sistem menampilkan pertanyaan konfirmasi apakah detail transaksi penjualan ingin dihapus ?
4. Aktor mengkonfirmasi ingin menghapus detail transaksi penjualan
5. Sistem menyimpan detail transaksi penjualan yang ada ke database
6. Berlanjut ke Basic Flow langkah ke-7

A-5 *Primary* dan *Supporting* Aktor memilih untuk menampilkan detail transaksi penjualan

1. Sistem menampilkan detail transaksi penjualan
2. Berlanjut ke Basic Flow langkah ke-7

A-6 *Primary* Aktor memilih untuk mencetak nota transaksi penjualan

1. Sistem menampilkan nota transaksi penjualan

Program Studi Teknik Informatika	SKPL – SISIP	72/76
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. Berlanjut ke Basic Flow langkah ke-7

6. Error Flow

E-1 data transaksi penjualan yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

E-2 data detail transaksi penjualan yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

E-2 data detail transaksi penjualan yang diinputkan aktor salah atau kurang lengkap

1. Sistem menampilkan pesan peringatan bahwa data yang diinputkan salah atau kurang lengkap
2. Sistem memberi penanda pada field yang salah
3. Kembali ke Basic Flow Langkah ke 3

2. PreConditions

1. Aktor telah memasuki sistem

3. PostConditions

1. Data transaksi penjualan di database telah terupdate

Program Studi Teknik Informatika	SKPL – SISIP	73/76
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		

4.1.15 Use case Spesification : Pengelolaan Reporting (Laporan)

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola *reporting* atau laporan. Aktor dapat melihat laporan penjualan, laporan pendapatan, laporan histori stok barang, dan laporan histori harga barang.

2. Primary Actor

1. KASI PPU

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *reporting* atau laporan
2. Aktor memilih untuk melihat laporan penjualan secara berdasarkan cabang dan periode tahun
 - A-1 Aktor memilih untuk melihat laporan pendapatan per periode tahun
 - A-2 Aktor memilih untuk melihat laporan histori stok barang
 - A-3 Aktor memilih untuk melihat laporan histori harga barang
3. Aktor memilih cabang dan tahun yang akan ditampilkan
4. Sistem menampilkan data laporan penjualan
5. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melihat laporan pendapatan per periode tahun

Program Studi Teknik Informatika	SKPL – SISIP	74/76
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. Aktor memilih cabang dan tahun yang akan ditampilkan
 2. Sistem menampilkan data laporan pendapatan
 3. Berlanjut ke Basic Flow langkah ke 5
- A-2 Aktor memilih untuk melihat laporan histori stok barang
1. Sistem menampilkan laporan berupa grafik penjualan
 2. Berlanjut ke Basic Flow langkah ke 5
- A-3 Aktor memilih untuk melihat laporan histori harga barang
1. Sistem menampilkan laporan berupa grafik penjualan
 2. Berlanjut ke Basic Flow langkah ke 5

6. Error Flow

none

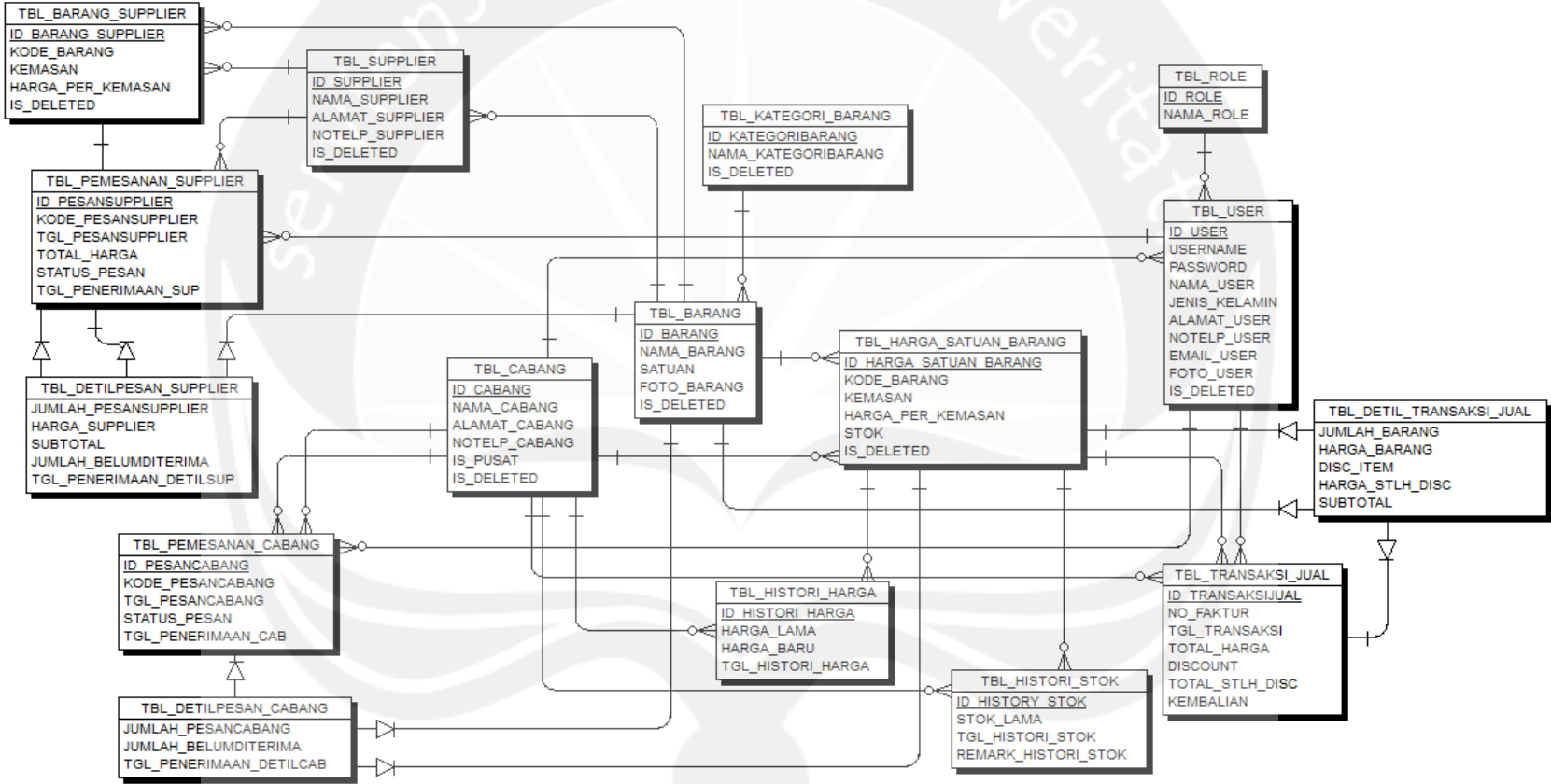
7. PreConditions

1. Aktor telah memasuki sistem

8. Post Conditions

1. Aktor telah dapat melihat laporan penjualan secara umum, laporan detil penjualan, laporan pendapatan, laporan daftar barang habis, melihat grafik penjualan, dan grafik pendapatan.

Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

**Sistem Informasi Distribusi dan Penjualan
Berbasis Web**

**(Studi Kasus : Bulog Mart Divre Daerah
Istimewa Yogyakarta)**

Untuk :


Tugas Akhir (Skripsi)

Dipersiapkan oleh:

Ancilla Alfionita Intan C. / 12 07 07051

**Program Studi Teknik Informatika - Fakultas Teknologi
Industri**

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL-SISIP		
		REVISI		

DAFTAR PERUBAHAN

REVISI	DESKRIPSI
A	
B	
C	
D	
E	
F	

INDEKSTGL	-	A	B	C	D	E	F
DITULISOLEH	AAIC						
DIPERIKSAOLEH	IW & FIN						
DISETUJUIOLEH	IW & FIN						

DAFTAR HALAMAN PERUBAHAN

Halaman	Revisi	Halaman	Revisi

DAFTAR ISI

1. Pendahuluan.....	11
1.1 Tujuan.....	11
1.2 Lingkup Masalah.....	11
1.3 Definisi, Akronim dan Singkatan.....	12
1.4 Referensi.....	13
2 Perancangan Sistem.....	14
2.1 Perancangan Asitektur.....	14
2.2 Perancangan Rinci.....	15
2.2.1 Pengelolaan Login.....	15
2.2.2 Ubah Password.....	16
2.2.3 Pengelolaan Data User.....	17
2.2.3.1 Tambah Data User	17
2.2.3.2 Ubah Data User	17
2.2.3.3 Cari Data User	18
2.2.3.4 Tampil Data User	18
2.2.4 Mengelola Data Cabang.....	19
2.2.4.1 Tambah Data Cabang	19
2.2.4.2 Ubah Data Cabang	19
2.2.4.3 Hapus Data Cabang	20
2.2.4.4 Cari Data Cabang	20
2.2.4.5 Tampil Data Cabang	21
2.2.5 Mengelola Data Supplier.....	21
2.2.5.1 Tambah Data Supplier	21
2.2.5.2 Ubah Data Supplier	22
2.2.5.3 Hapus Data Supplier	22
2.2.5.4 Cari Data Supplier	23
2.2.5.5 Tampil Data Supplier	23
2.2.6 Mengelola Data Kategori Barang.....	24
2.2.6.1 Tambah Data Kategori Barang	24
2.2.6.2 Ubah Data Kategori Barang	24
2.2.6.3 Hapus Data Kategori Barang	25
2.2.6.4 Cari Data Kategori Barang	25
2.2.6.5 Tampil Data Kategori Barang	26
2.2.7 Mengelola Data Barang.....	26
2.2.7.1 Tambah Data Barang	26
2.2.7.2 Ubah Data Barang	27
2.2.7.3 Hapus Data Barang	27
2.2.7.4 Cari Data Barang	28
2.2.7.5 Tampil Data Barang	28
2.2.8 Mengelola Data Detil Barang.....	29
2.2.8.1 Tambah Data Detil Barang	29
2.2.8.2 Ubah Data Detil Barang	30
2.2.8.3 Hapus Data Detil Barang	31
2.2.8.4 Cari Data Detil Barang	31
2.2.8.5 Tampil Data Detil Barang	32
2.2.9 Mengelola Barang Supplier.....	32

2.2.9.1	Tambah Data Barang Supplier	32
2.2.9.2	Ubah Data Barang Supplier	33
2.2.9.3	Cari Data Barang Supplier	34
2.2.9.4	Tampil Data Barang Supplier	34
2.2.10	Mengelola Data Pemesanan Barang ke Supplier.....	35
2.2.10.1	Generate Kode Pemesanan Barang ke Supplier	35
2.2.10.2	Tambah Data Pemesanan Barang ke Supplier	36
2.2.10.3	Batal Pemesanan Barang ke Supplier	37
2.2.10.4	Cari Pemesanan Barang ke Supplier	37
2.2.10.5	Ubah Detil Pemesanan Barang ke Supplier	38
2.2.10.6	Hapus Detil Pemesanan Barang ke Supplier	38
2.2.10.7	Tampil Pemesanan Barang ke Supplier	39
2.2.10.8	Cetak Dokumen Pemesanan Barang ke Supplier	39
2.2.11	Mengelola Data Pemesanan Barang ke Pusat/Antar Cabang....	40
2.2.11.1	Generate Kode Pemesanan Barang ke Pusat/Antar Cabang 40	
2.2.11.2	Tambah Data Pemesanan Barang ke Pusat/Antar Cabang .	41
2.2.11.3	Batal Pemesanan Barang ke Pusat/Antar Cabang	42
2.2.11.4	Cari Pemesanan Barang ke Pusat/Antar Cabang	42
2.2.11.5	Ubah Detil Pemesanan Barang ke Pusat/Antar Cabang ..	43
2.2.11.6	Hapus Detil Pemesanan Barang ke Pusat/Antar Cabang .	43
2.2.11.7	Tampil Pemesanan Barang ke Pusat/Supplier	44
2.2.11.8	Cetak Dokumen Pemesanan Barang ke Pusat/Antar Cabang 44	
2.2.12	Mengelola Penerimaan Barang dari Supplier.....	45
2.2.12.1	Tambah Penerimaan Barang dari Supplier	45
2.2.12.1	Ubah Detil Penerimaan Barang dari Supplier	46
2.2.13	Mengelola Penerimaan Barang dari Pusat/Antar Cabang.....	47
2.2.13.1	Tambah Penerimaan Barang dari Pusat/Antar Cabang ...	47
2.2.13.2	Ubah Detil Penerimaan Barang dari Pusat/Antar Cabang 48	
2.2.14	Mengelola Transaksi Penjualan.....	49
2.2.14.1	Generate No Faktur Transaksi Penjualan	49
2.2.14.2	Tambah Transaksi Penjualan	50
2.2.14.3	Batal Transaksi Penjualan	50
2.2.14.4	Cetak Nota Transaksi Penjualan	51
2.2.14.5	Tampil Transaksi Penjualan	51
2.2.14.6	Ubah Detil Transaksi Penjualan	52
2.2.14.7	Hapus Detil Transaksi Penjualan	52
2.2.15	Reporting (Pelaporan).....	53
2.2.15.1	Laporan Penjualan	53
2.2.15.2	Laporan Pendapatan	53
2.2.15.3	Laporan Histori Stok Barang	54
2.2.15.4	Histori Harga Barang	54
2.2.16	Class Diagram.....	55
2.2.17	Class Diagram Specific Descriptions.....	57
2.2.17.1	Specific Design Class LoginDekstopUI	57
2.2.17.2	Specific Design Class ChangePassUI	57
2.2.17.3	Specific Design Class insert_userUI	57

2.2.17.4	Specific Design Class	UserUI	58
2.2.17.5	Specific Design Class	CabangUI	59
2.2.17.6	Specific Design Class	KategoriBarangUI	60
2.2.17.7	Specific Design Class	BarangUI	61
2.2.17.8	Specific Design Class	detilBarangUI	61
2.2.17.9	Specific Design Class	barangSupplierUI	63
2.2.17.10	Specific Design Class	pemesananSupplierUI	64
2.2.17.10	Specific Design Class	detail_pemesananSupplierUI	65
2.2.17.11	Specific Design Class	pemesananCabangUI	66
2.2.17.11	Specific Design Class	detail_pemesananCabangUI	67
2.2.17.12	Specific Design Class	PenerimaanSupplierUI	67
2.2.17.13	Specific Design Class	PenerimaanCabangUI	68
2.2.17.14	Specific Design Class	TransPenjualanUI	69
2.2.17.15	Specific Design Class	detailtrans_penjualanUI	70
2.2.17.16	Specific Design Class	lapPenjualanUI()	70
2.2.17.17	Specific Design Class	lapPendapatanUI()	71
2.2.17.18	Specific Design Class	historiStokUI()	71
2.2.17.19	Specific Design Class	historiHargaUI()	71
2.2.17.20	Specific Design Class	loginCtrl()	72
2.2.17.21	Specific Design Class	userCtrl()	72
2.2.17.22	Specific Design Class	cabangCtrl()	73
2.2.17.23	Specific Design Class	kategori_barangCtrl()	74
2.2.17.24	Specific Design Class	barangCtrl()	75
2.2.17.25	Specific Design Class	detilbarangCtrl()	76
2.2.17.26	Specific Design Class	barangSupplierCtrl()	77
2.2.17.27	Spesific Design Class	pemesananSupplierCtrl	78
2.2.17.28	Spesific Design Class	pemesananCabangCtrl	79
2.2.17.29	Spesific Design Class	trans_penjualanCtrl	81
2.2.17.30	Spesific Design Class	laporanCtrl	82
2.2.17.31	Spesific Design Class	userModel	83
2.2.17.32	Specific Design Class	roleModel	85
2.2.17.33	Specific Design Class	supplierModel	85
2.2.17.34	Specific Design Class	cabangModel	86
2.2.17.35	Specific Design Class	kategori_barangModel	87
2.2.17.36	Spesific Design Class	barangModel	88
2.2.17.1	Spesific Design Class	detilBarangModel	89
2.2.17.1	Spesific Design Class	barangSupplierModel	90
2.2.17.2	Spesific Design Class	histori_hargaModel	91
2.2.17.3	Specific Design Class	histori_stokModel	92
2.2.17.4	Specific Design Class	hargaSatuan_barangModel	93
2.2.17.5	Specific Design Class	barangSupplierModel	93
2.2.17.6	Spesific Design Class	pemesananSupplierModel	94
2.2.17.7	Specific Design Class	detilPemesananSupplierModel	96
2.2.17.8	Spesific Design Class	pemesananCabangModel	97
2.2.17.9	Specific Design Class	detilPemesananCabangModel	98
2.2.17.10	Specific Design Class	trans_penjualanModel	99
2.2.17.11	Specific Design Class	detilTrans_penjualanModel	100
3	PERANCANGAN DATA		101
3.1	Dekomposisi Data		101
3.1.1	Deskripsi Entitas User		101
3.1.2	Deskripsi Entitas Role		102
3.1.3	Deskripsi Entitas Cabang		102

3.1.4	Deskripsi Entitas Supplier.....	102
3.1.5	Deskripsi Entitas Kategori Barang.....	103
3.1.6	Deskripsi Entitas Barang.....	103
3.1.7	Deskripsi Entitas Harga Satuan Barang.....	103
3.1.8	Deskripsi Entitas Harga Satuan Barang.....	104
3.1.9	Deskripsi Entitas Pemesanan ke Supplier.....	105
3.1.10	Deskripsi Entitas Detil Pemesanan ke Supplier.....	106
3.1.11	Deskripsi Entitas Pemesanan ke Pusat/Antar Cabang.....	106
3.1.12	Deskripsi Entitas Detil Pemesanan ke Pusat/Antar Cabang.....	107
3.1.13	Deskripsi Entitas Transaksi Penjualan.....	108
3.1.14	Deskripsi Entitas Detil Transaksi Penjualan.....	108
3.1.15	Deskripsi Entitas Histori Stok Barang.....	109
3.1.16	Deskripsi Entitas Histori Harga Barang.....	109
3.1.17	Deskripsi Entitas Bulan.....	110
3.2	Physical Data Model.....	111
4	Perancangan Antarmuka.....	112
4.1	Antarmuka Halaman Login Web.....	112
4.2	Antarmuka Halaman Beranda Web untuk Administrator.....	113
4.3	Antarmuka Halaman Beranda Web untuk KASI PPU.....	114
4.4	Antarmuka Halaman Beranda Web untuk Gudang.....	115
4.5	Antarmuka Halaman Beranda Web untuk Kasir.....	116
4.6	Antarmuka Halaman Pengelolaan Data User.....	117
4.7	Antarmuka Halaman Pengelolaan Cabang.....	120
4.8	Antarmuka Halaman Kategori Barang.....	123
4.9	Antarmuka Halaman Pengelolaan Barang.....	126
4.10	Antarmuka Halaman Detil Barang.....	129
4.11	Antarmuka Halaman Pengelolaan Supplier.....	132
4.12	Antarmuka Halaman View Stok Barang oleh Gudang.....	135
4.13	Antarmuka Halaman Pengelolaan Pemesanan Barang ke Supplier	136
4.14	Antarmuka Halaman Pengelolaan Pemesanan Barang Cabang ke Pusat	139
4.15	Antarmuka Halaman Pengelolaan Penerimaan Barang dari Supplier	141
4.16	Antarmuka Halaman Pengelolaan Penerimaan Barang dari Pusat ke Cabang	143
4.17	Antarmuka Transaksi Penjualan.....	146
4.18	Antarmuka Laporan Penjualan.....	148
4.19	Antarmuka Laporan Pendapatan.....	149
4.20	Antarmuka Laporan Histori Stok Barang.....	150
	Gambar 4.36 Rancangan Antarmuka Laporan Histori Stok Barang	150
4.21	Antarmuka Laporan Histori Harga Barang.....	151

DAFTAR GAMBAR

Gambar 2. 1	Perancangan Arsitekur Kelas SISIP.....	14
Gambar 2. 2	Sequence Diagram Login.....	15
Gambar 2. 3	Sequence Diagram Ubah Password.....	16
Gambar 2. 4	Sequence Diagram Tambah Data User.....	17
Gambar 2. 5	Sequence Diagram Ubah Data User.....	17
Gambar 2. 6	Sequence Diagram Cari Data User.....	18
Gambar 2. 7	Sequence Diagram Tampil Data User.....	18
Gambar 2. 8	Sequence Diagram Tambah Data Cabang.....	19
Gambar 2. 9	Sequence Diagram Ubah Data Cabang.....	19
Gambar 2. 10	Sequence Diagram Hapus Data Cabang.....	20
Gambar 2. 11	Sequence Diagram Cari Data Cabang.....	20
Gambar 2. 12	Sequence Diagram Tampil Data Cabang.....	21
Gambar 2. 13	Sequence Diagram Tambah Data Supplier.....	21
Gambar 2. 14	Sequence Diagram Ubah Data Supplier.....	22
Gambar 2. 15	Sequence Diagram Hapus Data Supplier.....	22
Gambar 2. 16	Sequence Diagram Cari Data Supplier.....	23
Gambar 2. 17	Sequence Diagram Tampil Data Supplier.....	23
Gambar 2. 18	Sequence Diagram Tambah Data Kategori Barang.....	24
Gambar 2. 19	Sequence Diagram Ubah Data Kategori Barang.....	24
Gambar 2. 20	Sequence Diagram Hapus Data Kategori Barang.....	25
Gambar 2. 21	Sequence Diagram Cari Data Kategori Barang.....	25
Gambar 2. 22	Sequence Diagram Tampil Data Kategori Barang.....	26
Gambar 2. 23	Sequence Diagram Tambah Data Barang.....	26
Gambar 2. 24	Sequence Diagram Ubah Data Barang.....	27
Gambar 2. 25	Sequence Diagram Hapus Data Kategori Barang.....	27
Gambar 2. 26	Sequence Diagram Cari Data Barang.....	28
Gambar 2. 27	Sequence Diagram Tampil Data Barang.....	28
Gambar 2. 28	Sequence Diagram Tambah Data Detil Barang.....	29
Gambar 2. 29	Sequence Diagram Ubah Data Detil Barang.....	30
Gambar 2. 30	Sequence Diagram Hapus Data Kategori Barang.....	31
Gambar 2. 31	Sequence Diagram Cari Data Barang.....	31
Gambar 2. 32	Sequence Diagram Tampil Data Detil Barang.....	32
Gambar 2. 33	Sequence Diagram Generate Kode Pemesanan Barang Supplier....	35
Gambar 2. 34	Sequence Diagram Tambah Data Pemesanan Barang Supplier.....	36
Gambar 2. 35	Sequence Diagram Batal Pemesanan Barang ke Supplier.....	37
Gambar 2. 36	Sequence Diagram Cari Data Pemesanan Barang ke Supplier.....	37
Gambar 2. 37	Sequence Diagram Ubah Detil Pemesanan Barang.....	38
Gambar 2. 38	Sequence Diagram Hapus Detil Pemesanan Barang.....	38
Gambar 2. 39	Sequence Diagram Tampil Data Pemesanan Barang ke Supplier...39	
Gambar 2. 40	Sequence Diagram Cetak Dokumen Pemesanan Barang ke Supplier.39	
Gambar 2. 41	Sequence Diagram Generate Kode Pemesanan Barang ke Pusat/Antar Cabang.....	40
Gambar 2. 42	Sequence Diagram Tambah Data Pemesanan Barang ke Pusat/Antar Cabang.....	41
Gambar 2. 43	Sequence Diagram Batal Pemesanan Barang ke Pusat/Antar Cabang	42
Gambar 2. 44	Sequence Diagram Cari Data Pemesanan Barang ke Pusat/Antar Cabang.....	42

Gambar 2. 45	Sequence Diagram Ubah Detil Pemesanan Barang.....	43
Gambar 2. 46	Sequence Diagram Hapus Detil Pemesanan Barang.....	43
Gambar 2. 47	Sequence Diagram Tampil Data Pemesanan Barang ke Pusat/Antar Cabang.....	44
Gambar 2. 48	Sequence Diagram Tampil Data Pemesanan Barang ke Pusat/Antar Cabang.....	44
Gambar 2. 49	Sequence Diagram Tambah Data Penerimaan Barang dari Supplier	45
Gambar 2. 50	Sequence Diagram Ubah Data Penerimaan Barang dari Supplier..	46
Gambar 2. 51	Sequence Diagram Tambah Data Penerimaan Barang dari Pusat/Antar Cabang.....	47
Gambar 2. 52	Sequence Diagram Ubah Data Penerimaan Barang dari Pusat/Antar Cabang.....	48
Gambar 2. 53	Sequence Diagram Generate No. Faktur Transaksi Penjualan....	49
Gambar 2. 54	Sequence Diagram Tambah Transaksi Penjualan.....	50
Gambar 2. 55	Sequence Diagram Batal Transaksi Penjualan.....	50
Gambar 2. 56	Sequence Diagram Cetak Nota Transaksi Penjualan.....	51
Gambar 2. 57	Sequence Diagram Tampil Transaksi Penjualan.....	51
Gambar 2. 58	Sequence Diagram Ubah Detil Transaksi Penjualan.....	52
Gambar 2. 59	Sequence Diagram Hapus Detil Transaksi Penjualan.....	52
Gambar 2. 60	Sequence Diagram Laporan Penjualan.....	53
Gambar 2. 61	Sequence Diagram Laporan Pendapatan.....	53
Gambar 2. 62	Sequence Diagram Laporan Histori Stok Barang.....	54
Gambar 2. 63	Sequence Diagram Laporan Histori Harga Barang.....	54
Gambar 2. 64	Class Diagram SISIP.....	56
Gambar 2. 65	Physical Data Model SISIP.....	111
Gambar 4. 1	Rancangan Antarmuka <i>Login Web</i>	112
Gambar 4.2	Rancangan Antarmuka Halaman Beranda Web untuk Administrator..	113
Gambar 4.3	Rancangan Antarmuka Halaman Beranda Web untuk KASI PPU.....	114
Gambar 4.4	Rancangan Antarmuka Halaman Beranda Web untuk Gudang.....	115
Gambar 4.5	Rancangan Antarmuka Halaman Beranda Web untuk Kasir.....	116
Gambar 4.6	Rancangan Antarmuka Halaman User.....	117
Gambar 4.7	Rancangan Antarmuka Tambah User.....	118
Gambar 4.8	Rancangan Antarmuka Ubah Data User.....	119
Gambar 4.9	Rancangan Antarmuka Halaman Cabang.....	120
Gambar 4.10	Rancangan Antarmuka Halaman Tambah Cabang.....	121
Gambar 4.11	Rancangan Antarmuka Halaman Ubah Data Cabang.....	122
Gambar 4.12	Rancangan Antarmuka Pengelolaan Kategori Barang.....	123
Gambar 4.13	Rancangan Antarmuka Tambah Kategori Barang.....	124
Gambar 4.14	Rancangan Antarmuka Ubah Data Kategori Barang.....	125
Gambar 4.15	Rancangan Antarmuka Pengelolaan Barang.....	126
Gambar 4.16	Rancangan Antarmuka Tambah Barang.....	127
Gambar 4.17	Rancangan Antarmuka Ubah Data Barang.....	128
Gambar 4.18	Rancangan Antarmuka Pengelolaan Detil Barang.....	129
Gambar 4.19	Rancangan Antarmuka Tambah Detil Barang.....	130
Gambar 4.20	Rancangan Antarmuka Ubah Data Detil Barang.....	131
Gambar 4.21	Rancangan Antarmuka Pengelolaan <i>Supplier</i>	132
Gambar 4.22	Rancangan Antarmuka Tambah <i>Supplier</i>	133
Gambar 4.23	Rancangan Antarmuka Ubah Data <i>Supplier</i>	134
Gambar 4.24	Rancangan Antarmuka Lihat Stok Barang.....	135
Gambar 4.25	Rancangan Antarmuka Pengelolaan Pemesanan Barang ke <i>upplier</i>	136

Gambar 4.26 Rancangan Antarmuka Tambah Pemesanan Barang ke <i>Supplier</i>	137
Gambar 4.27 Rancangan Antarmuka Pengelolaan Pemesanan Barang Cabang ke Pusat.....	139
Gambar 4.28 Rancangan Antarmuka Tambah Pemesanan Barang Cabang ke Pusat.	140
Gambar 4.29 Rancangan Antarmuka Pengelolaan Penerimaan dari <i>Supplier</i>	141
Gambar 4.30 Rancangan Antarmuka Checking Barang dari <i>Supplier</i> dan Ubah Status Terima.....	142
Gambar 4.31 Rancangan Antarmuka Pengelolaan Penerimaan Pusat ke Cabang..	143
Gambar 4.32 Rancangan Antarmuka Tambah Penerimaan Pusat ke Cabang.....	144
Gambar 4.33 Rancangan Antarmuka Transaksi Penjualan.....	146
Gambar 4.34 Rancangan Antarmuka Transaksi Penjualan.....	148
Gambar 4.35 Rancangan Antarmuka Transaksi Penjualan.....	149
Gambar 4.36 Rancangan Antarmuka Laporan Histori Stok Barang.....	150
Gambar 4.37 Rancangan Antarmuka Laporan Histori Harga Barang.....	151



1. Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen tersebut akan digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap berikutnya.

1.2 Lingkup Masalah

Perangkat Lunak SISIP dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan yang terkait dengan cabang, kategori barang, barang, detil barang, supplier, dan barang supplier yang ada di Bulog Mart.
2. Menangani pengelolaan pemesanan dan penerimaan barang supplier, serta pemesanan dan penerimaan barang ke pusat atau antar cabang.
3. Menangani pengelolaan transaksi penjualan Bulog Mart.
4. Menangani pengelolaan *reporting* jumlah transaksi penjualan pada setiap cabang, jumlah pendapatan setiap cabang, histori stok barang, dan histori harga barang.

Dan aplikasi ini berjalan pada lingkungan dengan *web browser*.

Program Studi Teknik Informatika	DPPL-SISIP	11/151
Dokumen ini adalah 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 spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
DPPL-SISIP-XXX	Kode yang merepresentasikan kebutuhan pada SISIP(Sistem Informasi Distribusi dan Penjualan) dimana XXX merupakan nomor fungsi produk.
SISIP	Perangkat lunak pengelolaan web.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Bulog Mart	Salah satu pengembangan usaha Perum Bulog berupa toko retail sembako untuk membantu masyarakat memenuhi kebutuhan pokok.
Nota Transaksi	Nota yang berisi detail transaksi penjualan dari pembelian barang di Bulog

Penjualan	Mart.
Terkirim ke Supplier	Status dimana barang yang yang dipesan sudah diproses ke supplier
Terkirim ke Cabang	Status dimana barang yang dipesan belum diproses ke cabang.
Belum Diterima	Status di mana barang yang belum diterima oleh gudang.
Barang Diterima Belum Lengkap	Status di mana barang yang diterima oleh gudang cabang belum lengkap
Barang Diterima Lengkap	Status di mana barang yang diterima oleh gudang cabang lengkap
Hapus semu	Sebenarnya tidak menghapus data, hanya membuat data tidak di tampilkan, dengan mengubah nilai is_deleted menjadi satu.

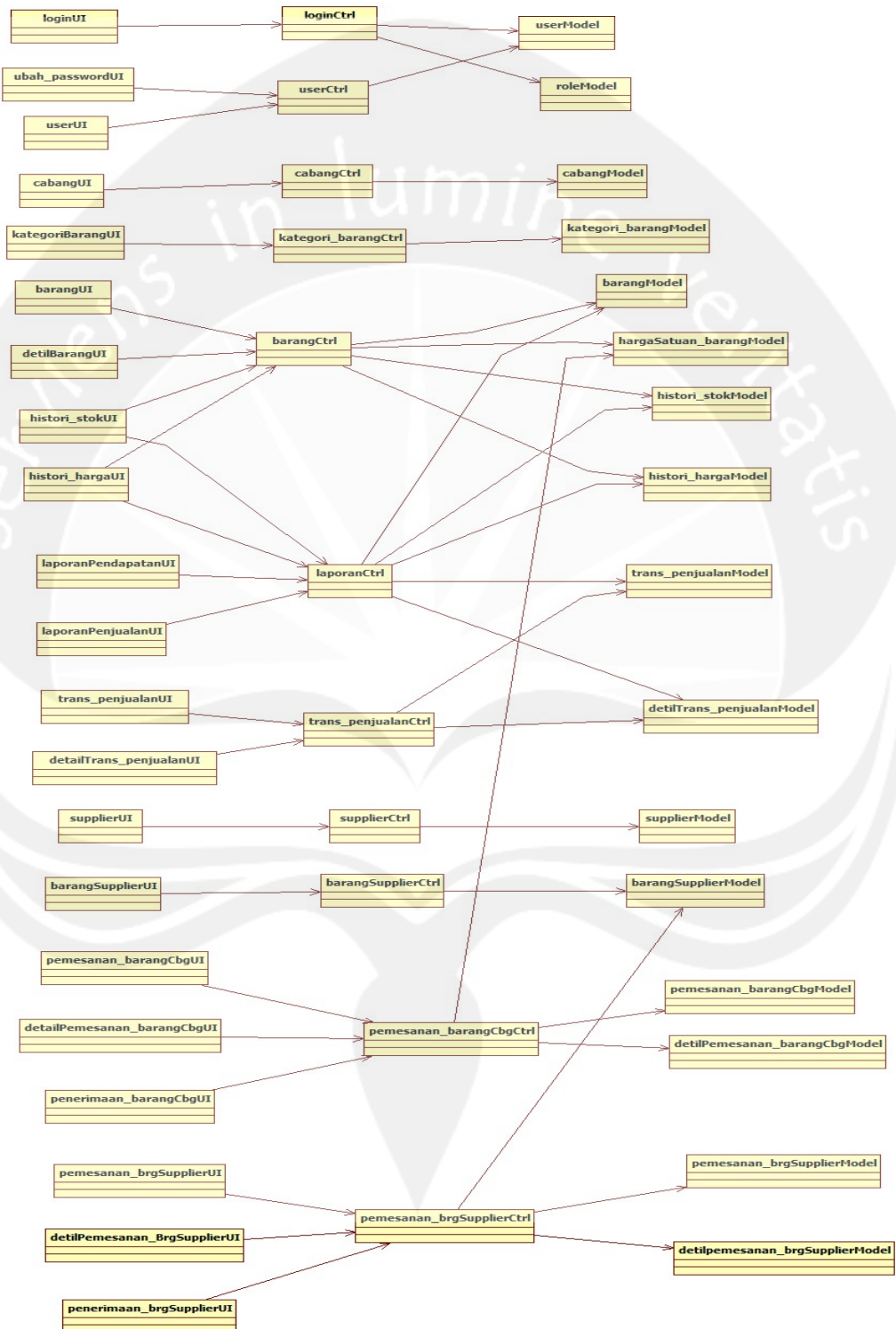
1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Irene Deandra Indarto, *Deskripsi Perancangan Perangkat Lunak (DPPL) SISIP*, Universitas Atma Jaya Yogyakarta.
2. Yohanes Yanuar Adi Nugroho, *Deskripsi Perancangan Perangkat Lunak (DPPL) SITB*, Universitas Atma Jaya Yogyakarta.

2 Perancangan Sistem

2.1 Perancangan Asitektur

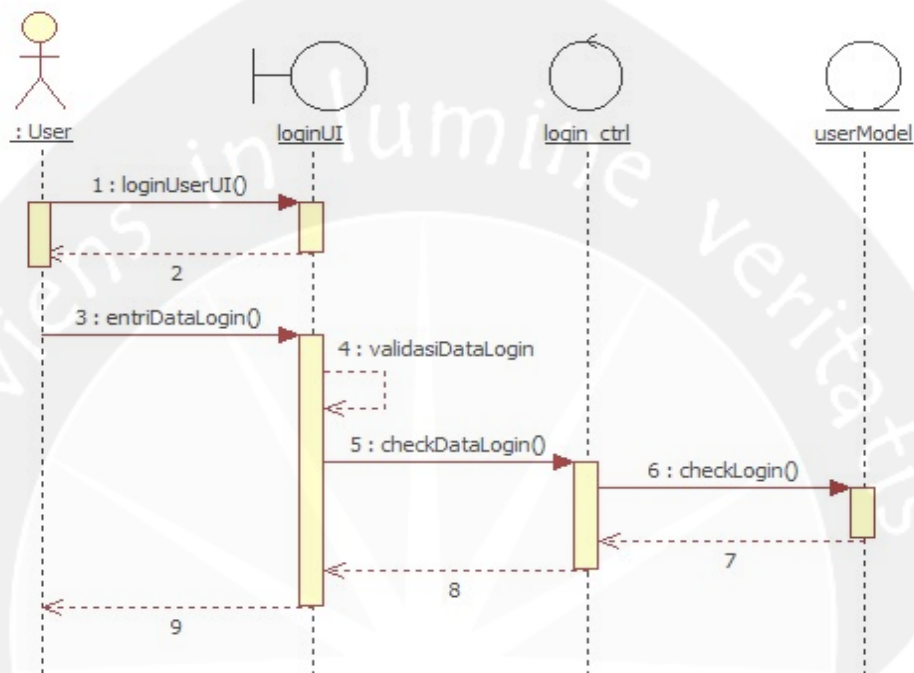


Gambar 2. 1 Perancangan Arsitektur Kelas SISIP

2.2 Perancangan Rinci

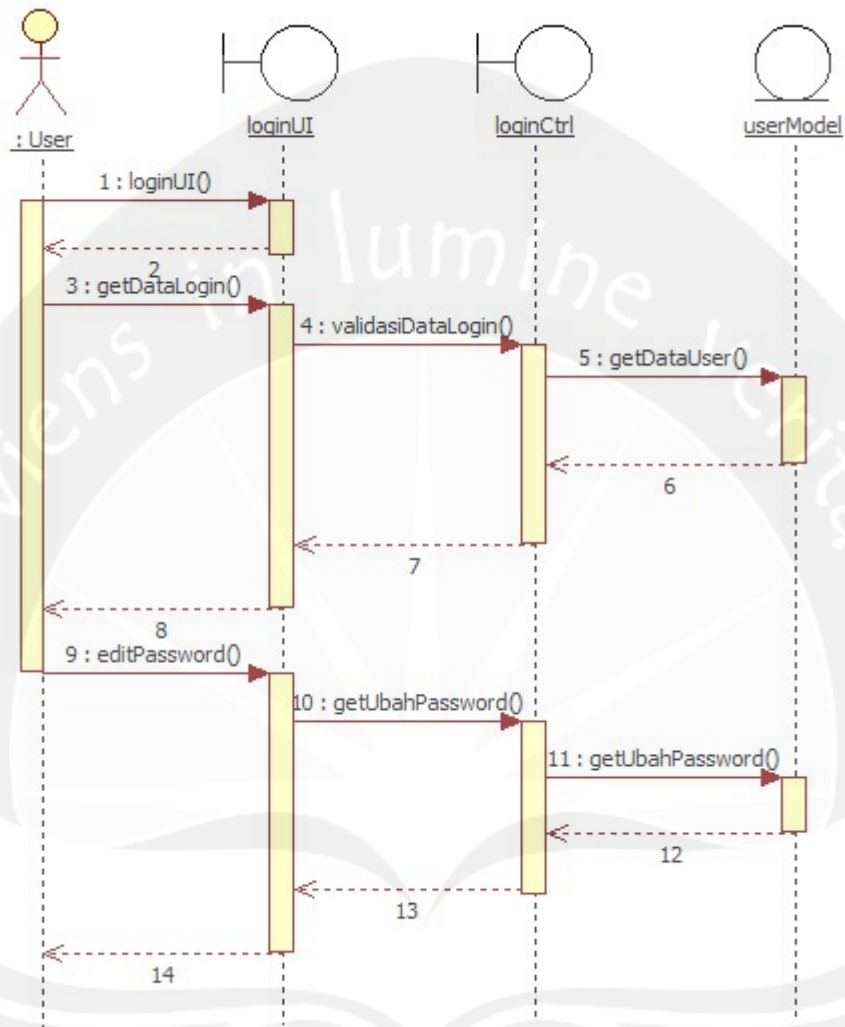
2.2.1 Pengelolaan Login

2.2.1.1 Login



Gambar 2. 2 Sequence Diagram Login

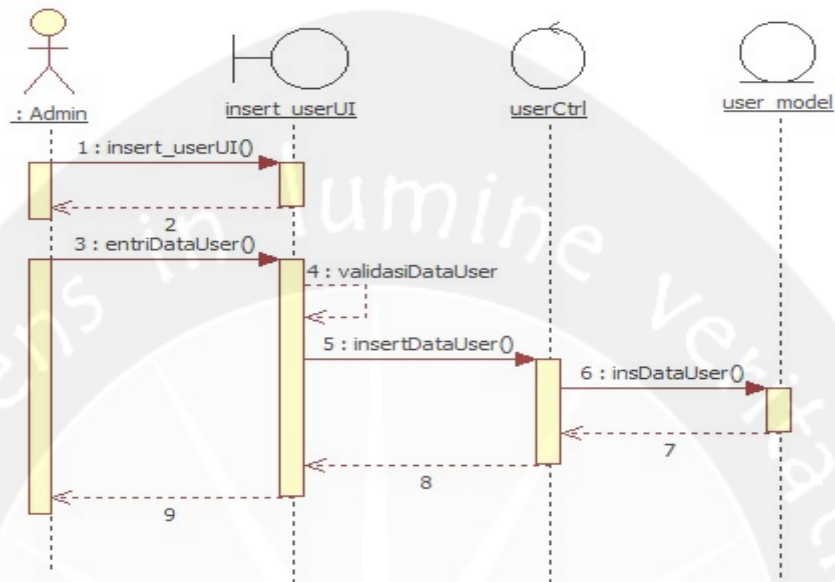
2.2.2 Ubah Password



Gambar 2. 3 Sequence Diagram Ubah Password

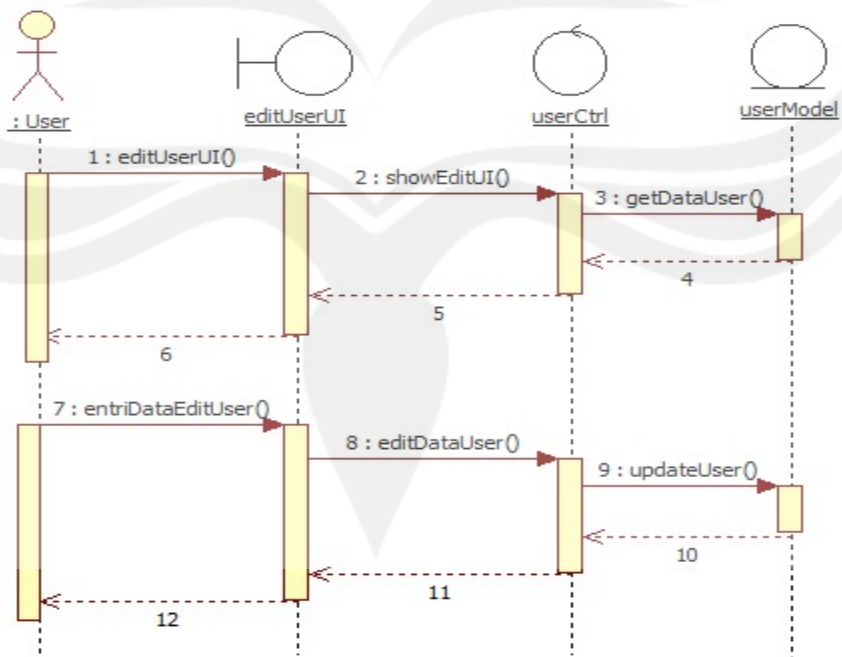
2.2.3 Pengelolaan Data User

2.2.3.1 Tambah Data User



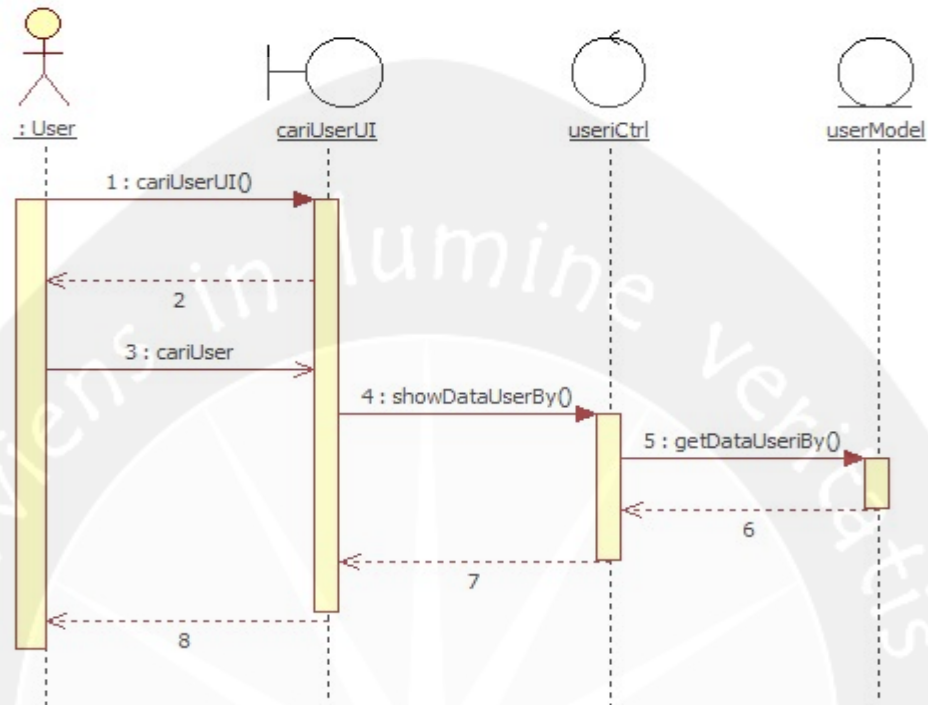
Gambar 2. 4 Sequence Diagram Tambah Data User

2.2.3.2 Ubah Data User



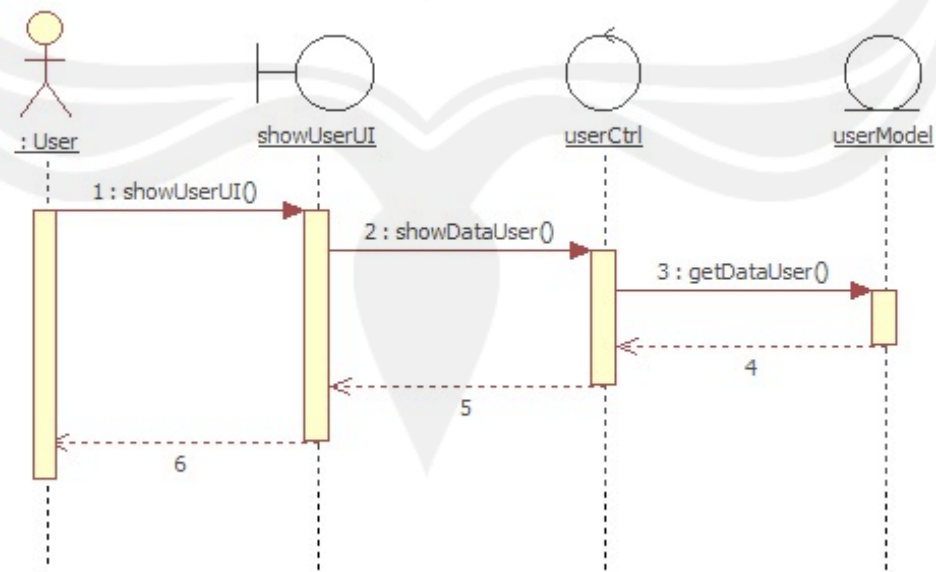
Gambar 2. 5 Sequence Diagram Ubah Data User

2.2.3.3 Cari Data User



Gambar 2. 6 Sequence Diagram Cari Data User

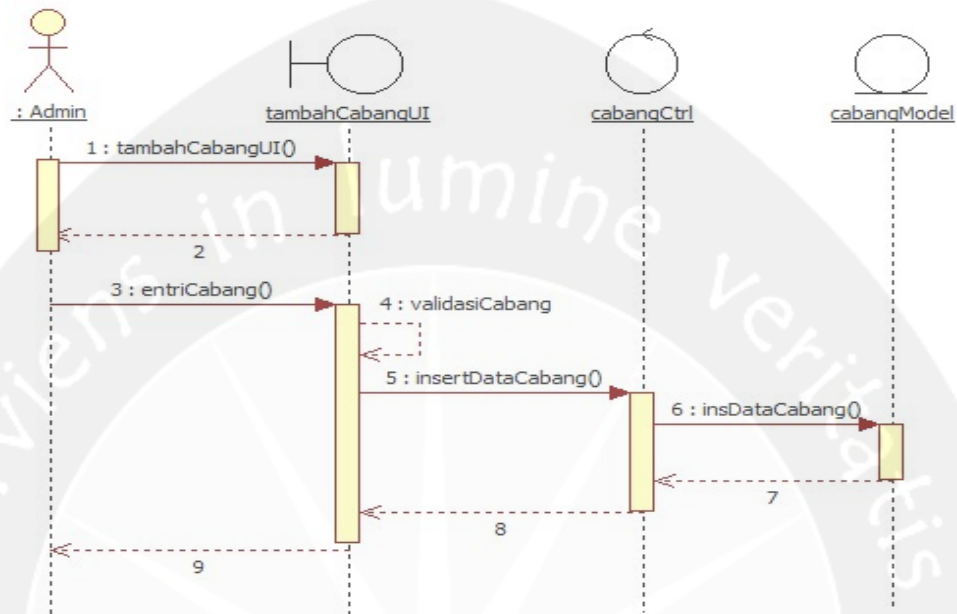
2.2.3.4 Tampil Data User



Gambar 2. 7 Sequence Diagram Tampil Data User

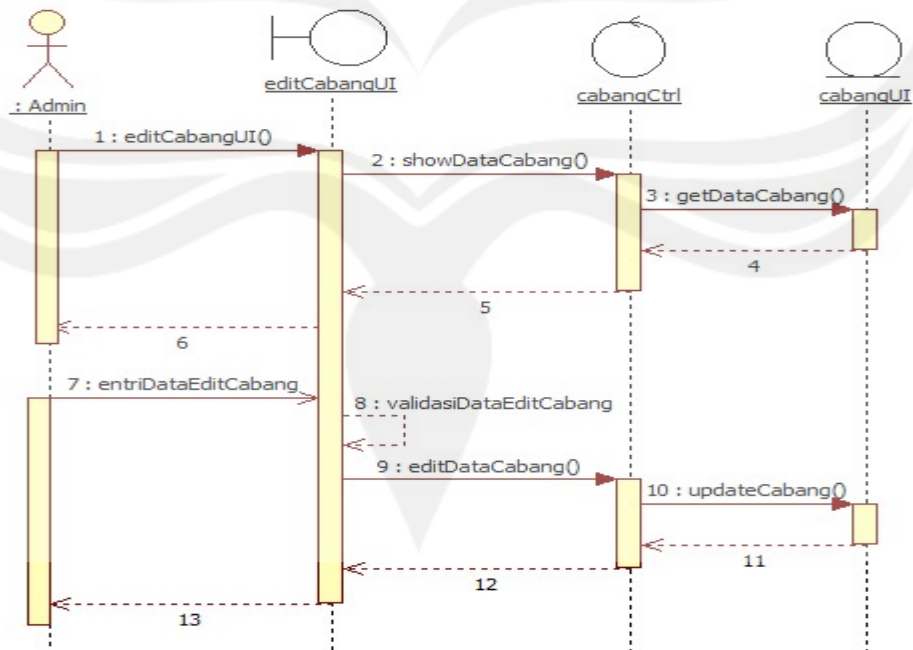
2.2.4 Mengelola Data Cabang

2.2.4.1 Tambah Data Cabang



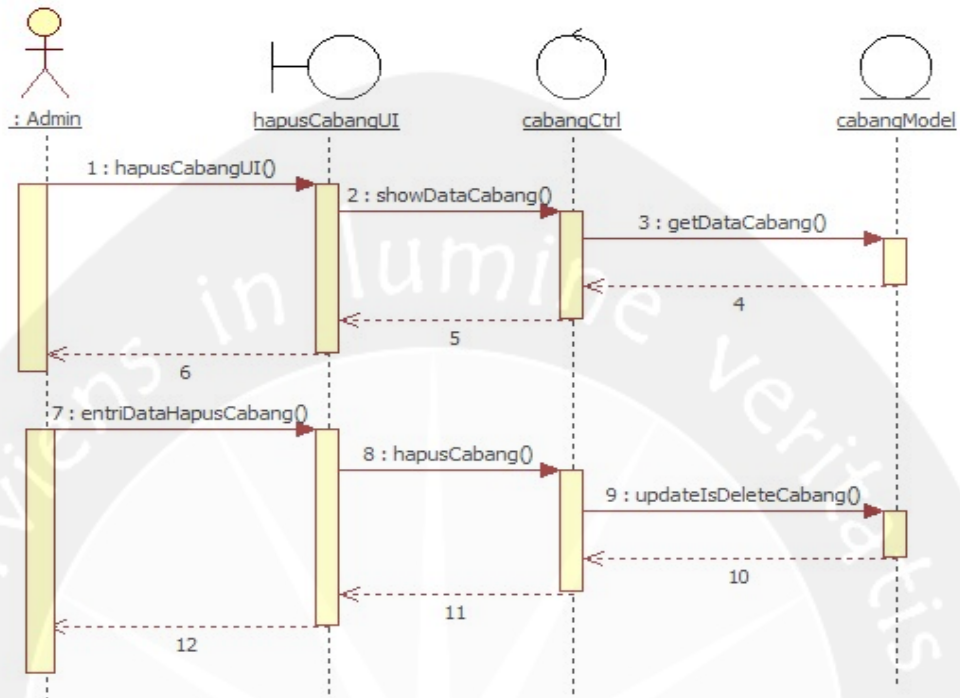
Gambar 2. 8 Sequence Diagram Tambah Data Cabang

2.2.4.2 Ubah Data Cabang



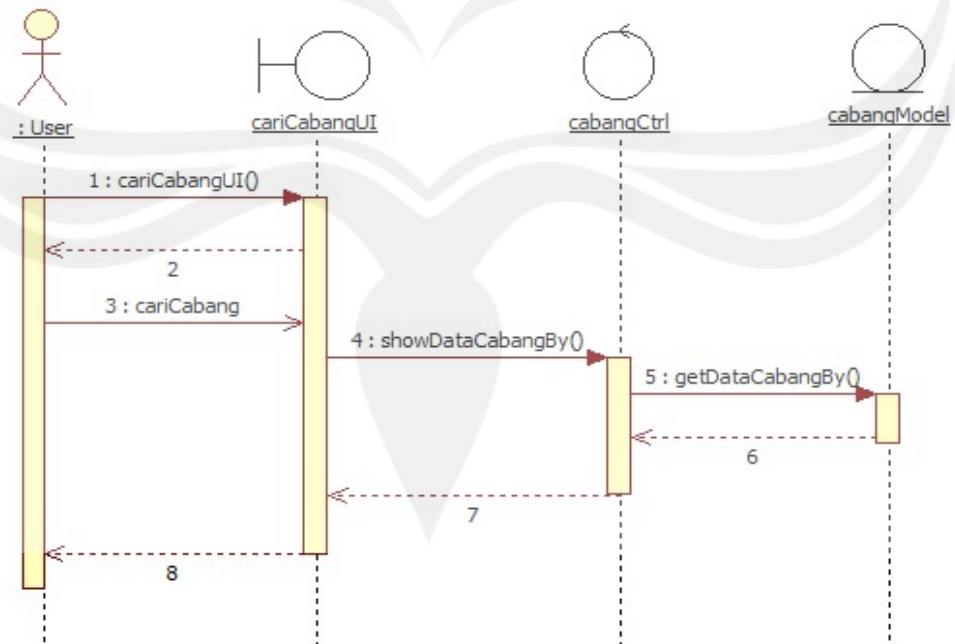
Gambar 2. 9 Sequence Diagram Ubah Data Cabang

2.2.4.3 Hapus Data Cabang



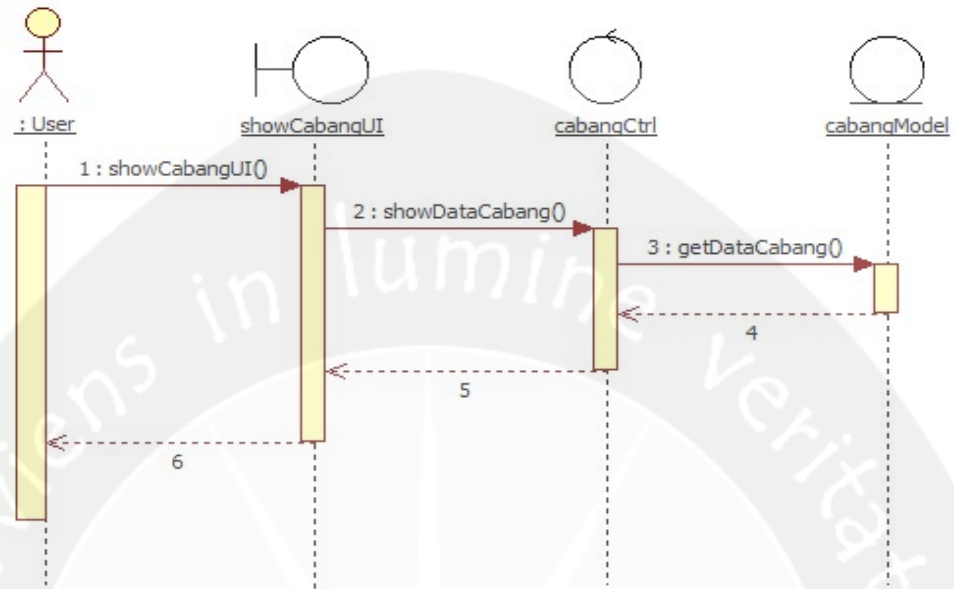
Gambar 2. 10 Sequence Diagram Hapus Data Cabang

2.2.4.4 Cari Data Cabang



Gambar 2. 11 Sequence Diagram Cari Data Cabang

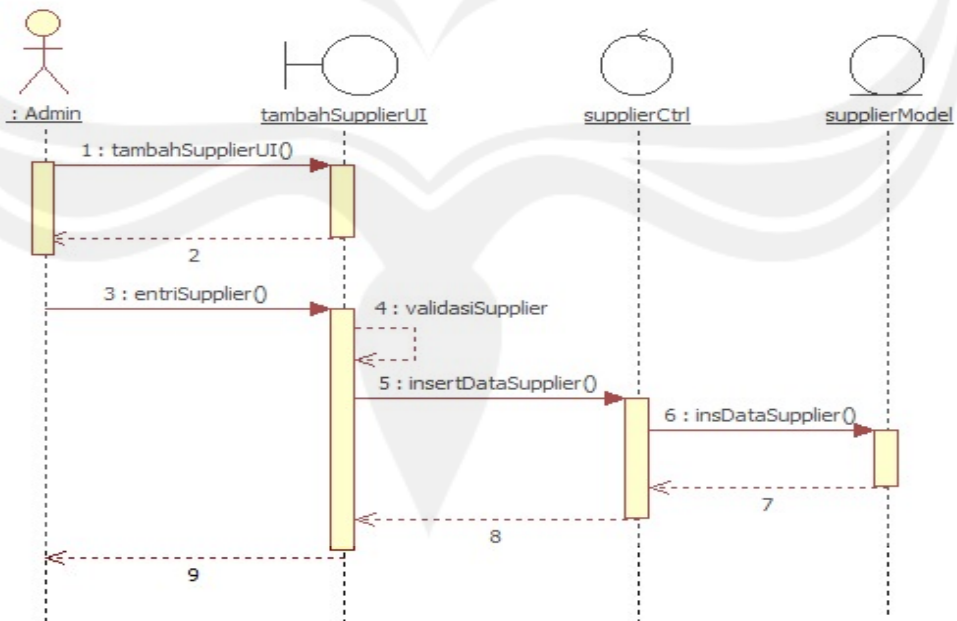
2.2.4.5 Tampil Data Cabang



Gambar 2. 12 Sequence Diagram Tampil Data Cabang

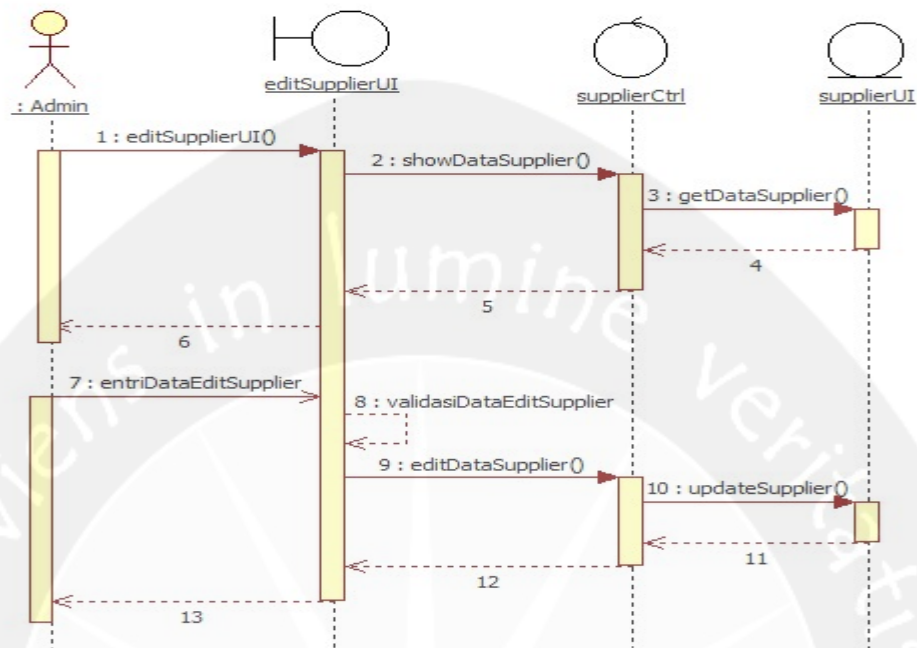
2.2.5 Mengelola Data Supplier

2.2.5.1 Tambah Data Supplier



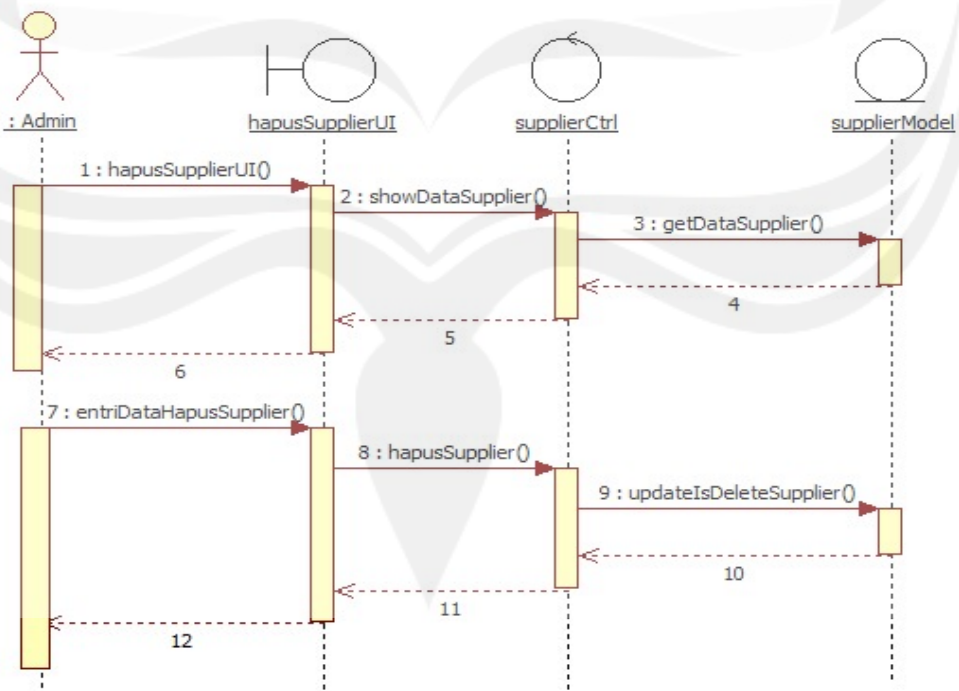
Gambar 2. 13 Sequence Diagram Tambah Data Supplier

2.2.5.2 Ubah Data Supplier



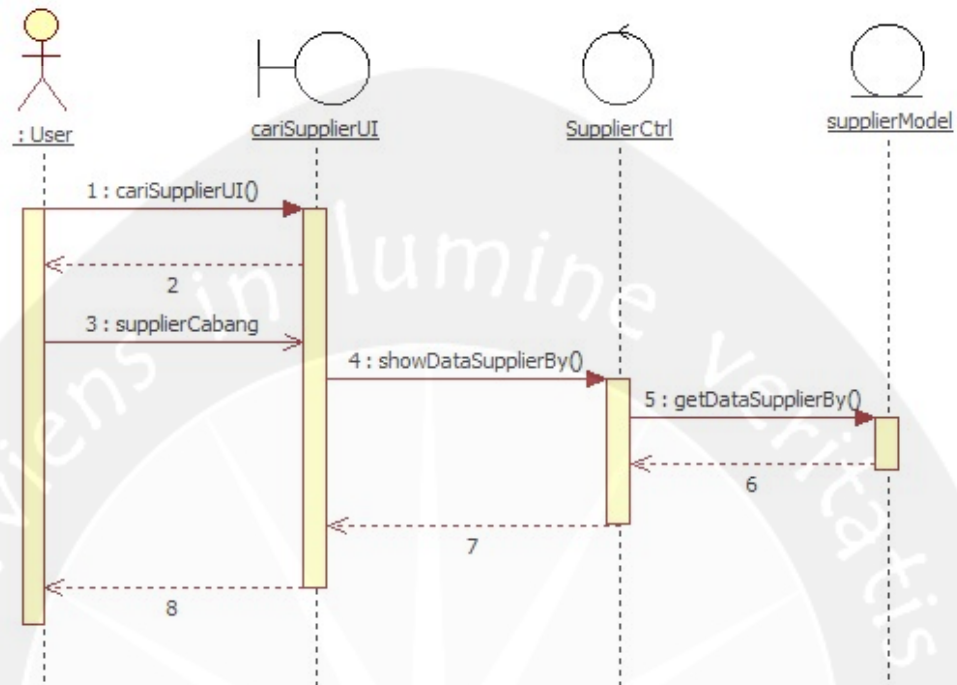
Gambar 2. 14 Sequence Diagram Ubah Data Supplier

2.2.5.3 Hapus Data Supplier



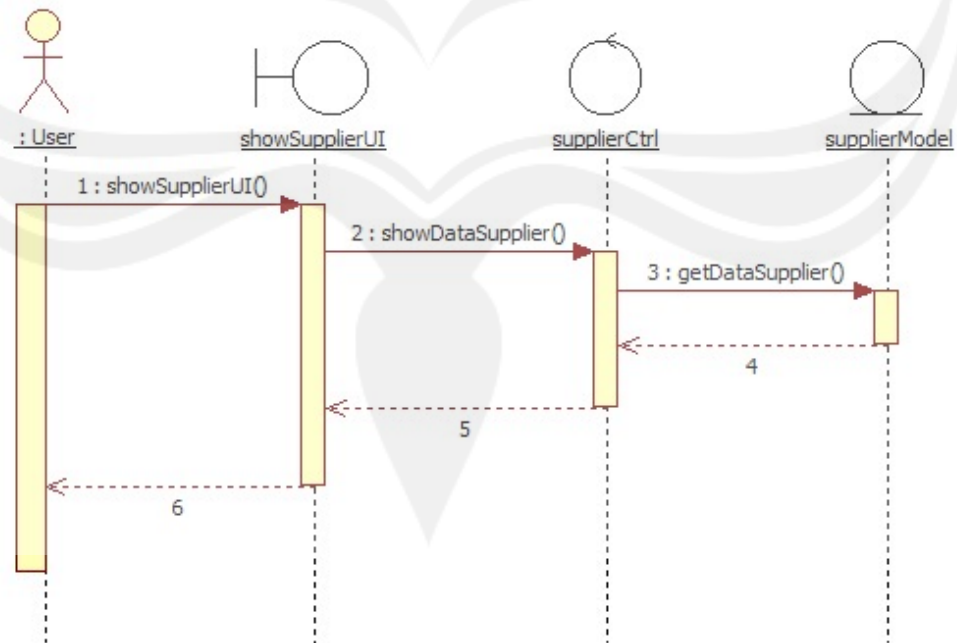
Gambar 2. 15 Sequence Diagram Hapus Data Supplier

2.2.5.4 Cari Data Supplier



Gambar 2. 16 Sequence Diagram Cari Data Supplier

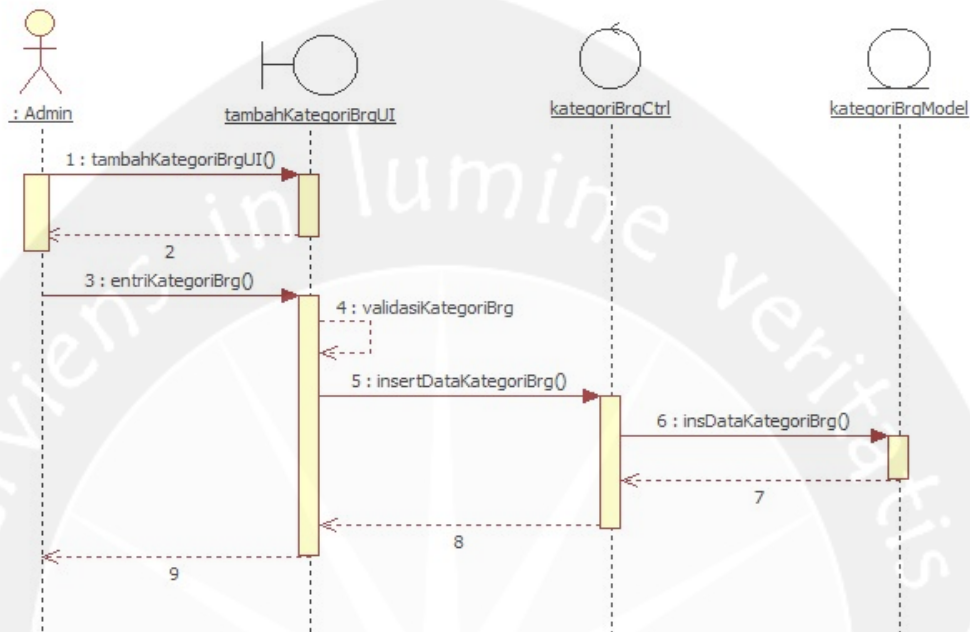
2.2.5.5 Tampil Data Supplier



Gambar 2. 17 Sequence Diagram Tampil Data Supplier

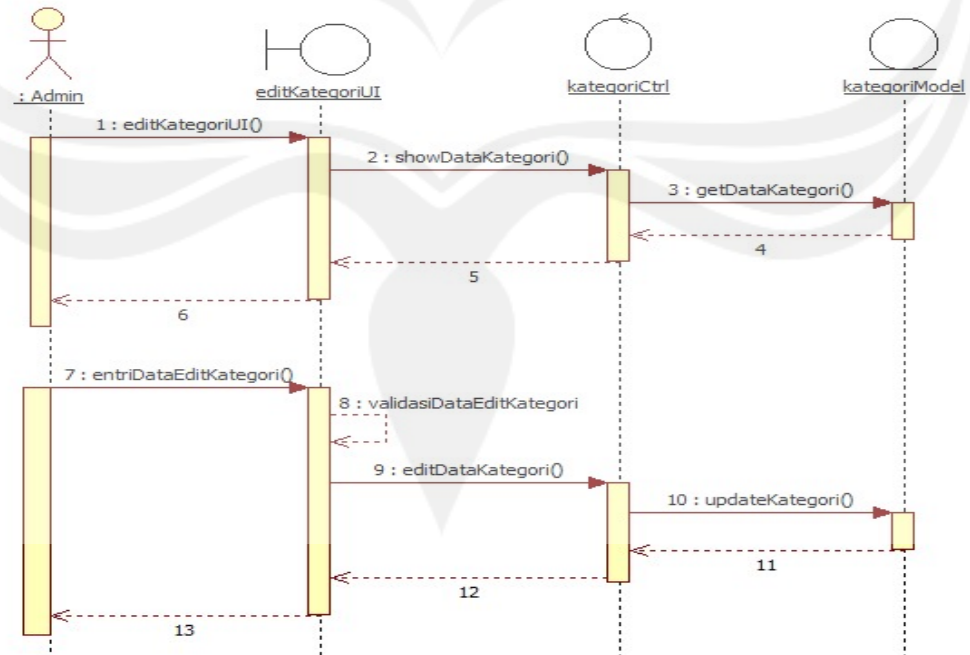
2.2.6 Mengelola Data Kategori Barang

2.2.6.1 Tambah Data Kategori Barang



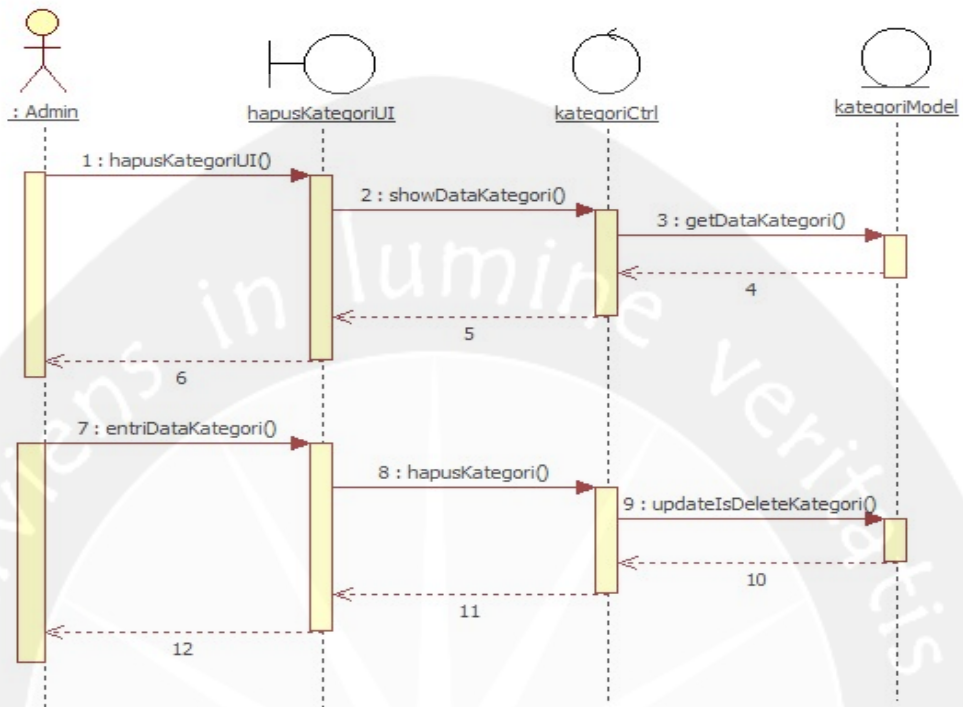
Gambar 2. 18 Sequence Diagram Tambah Data Kategori Barang

2.2.6.2 Ubah Data Kategori Barang



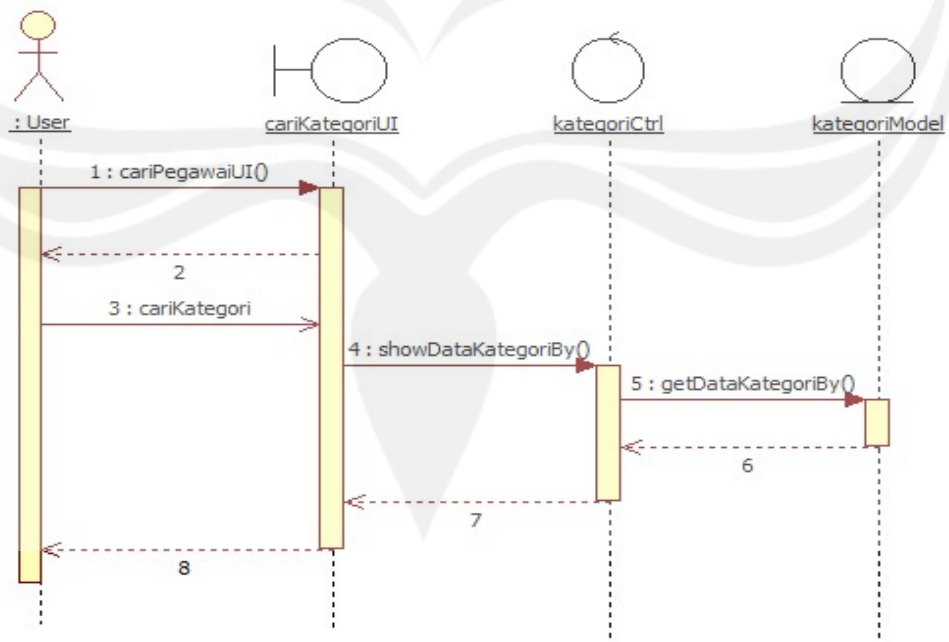
Gambar 2. 19 Sequence Diagram Ubah Data Kategori Barang

2.2.6.3 Hapus Data Kategori Barang



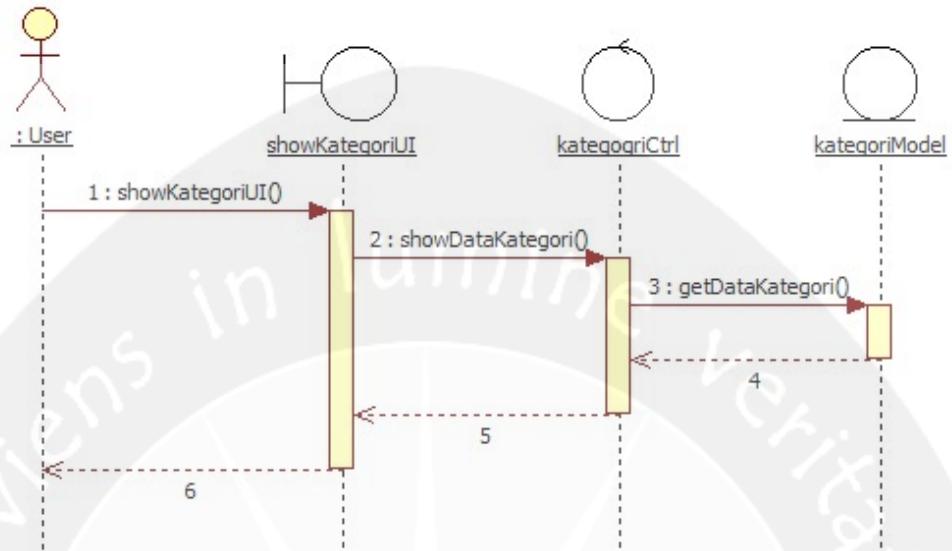
Gambar 2. 20 Sequence Diagram Hapus Data Kategori Barang

2.2.6.4 Cari Data Kategori Barang



Gambar 2. 21 Sequence Diagram Cari Data Kategori Barang

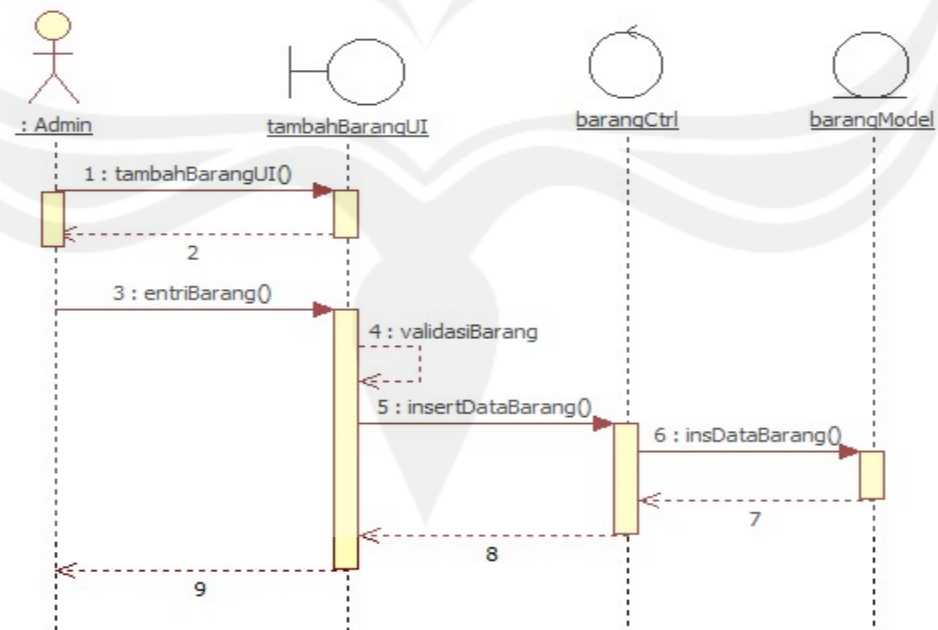
2.2.6.5 Tampil Data Kategori Barang



Gambar 2. 22 Sequence Diagram Tampil Data Kategori Barang

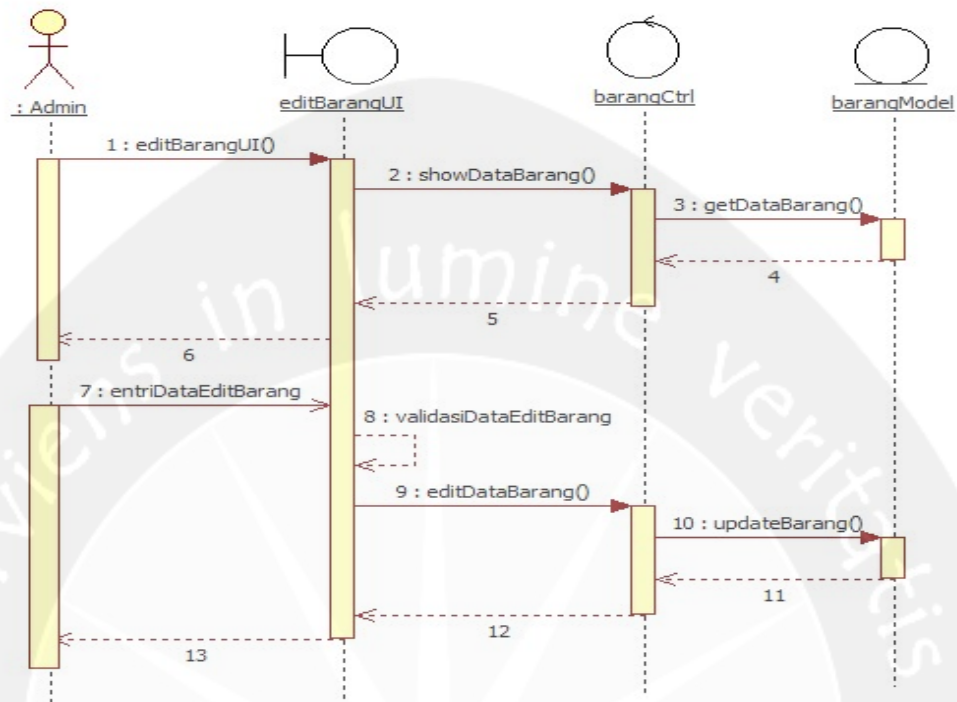
2.2.7 Mengelola Data Barang

2.2.7.1 Tambah Data Barang



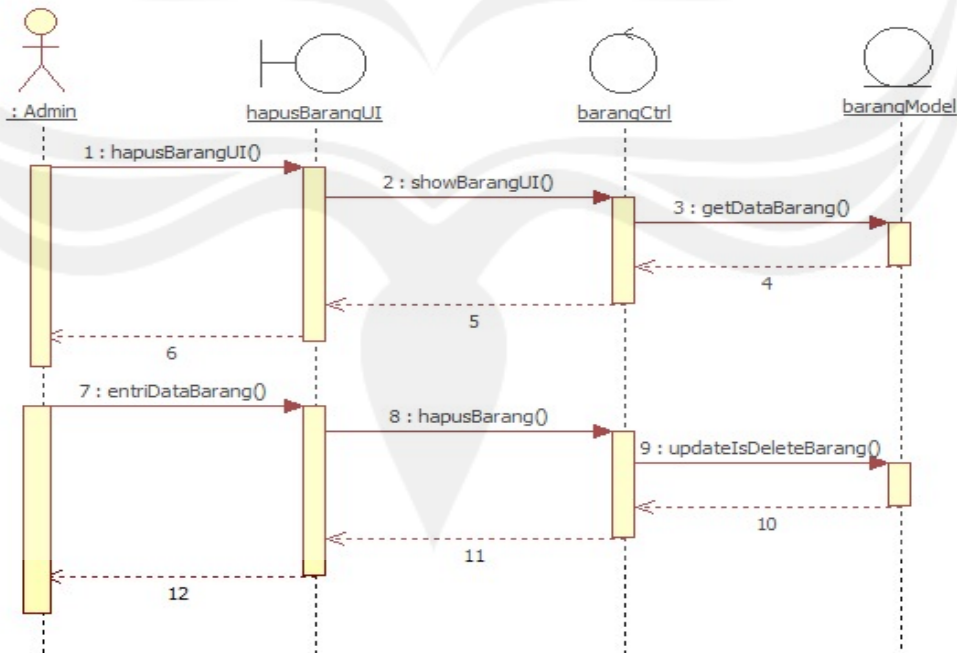
Gambar 2. 23 Sequence Diagram Tambah Data Barang

2.2.7.2 Ubah Data Barang



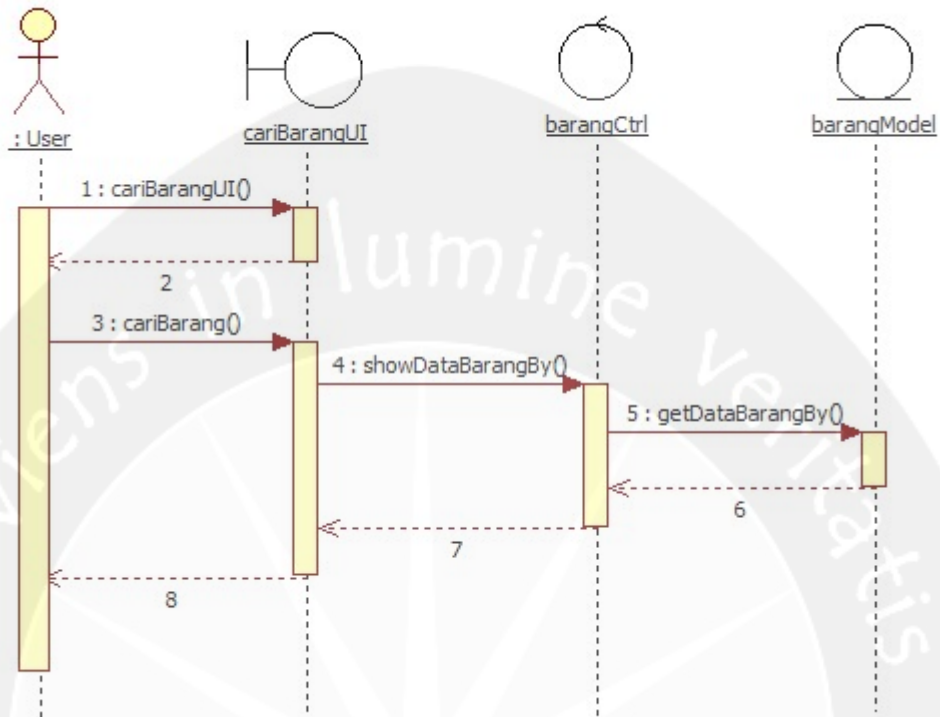
Gambar 2. 24 Sequence Diagram Ubah Data Barang

2.2.7.3 Hapus Data Barang



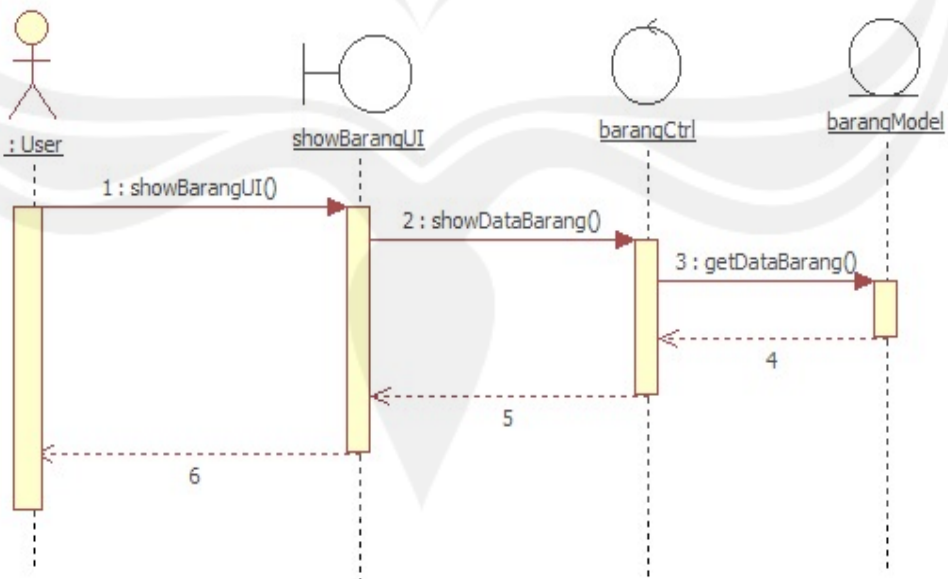
Gambar 2. 25 Sequence Diagram Hapus Data Kategori Barang

2.2.7.4 Cari Data Barang



Gambar 2. 26 Sequence Diagram Cari Data Barang

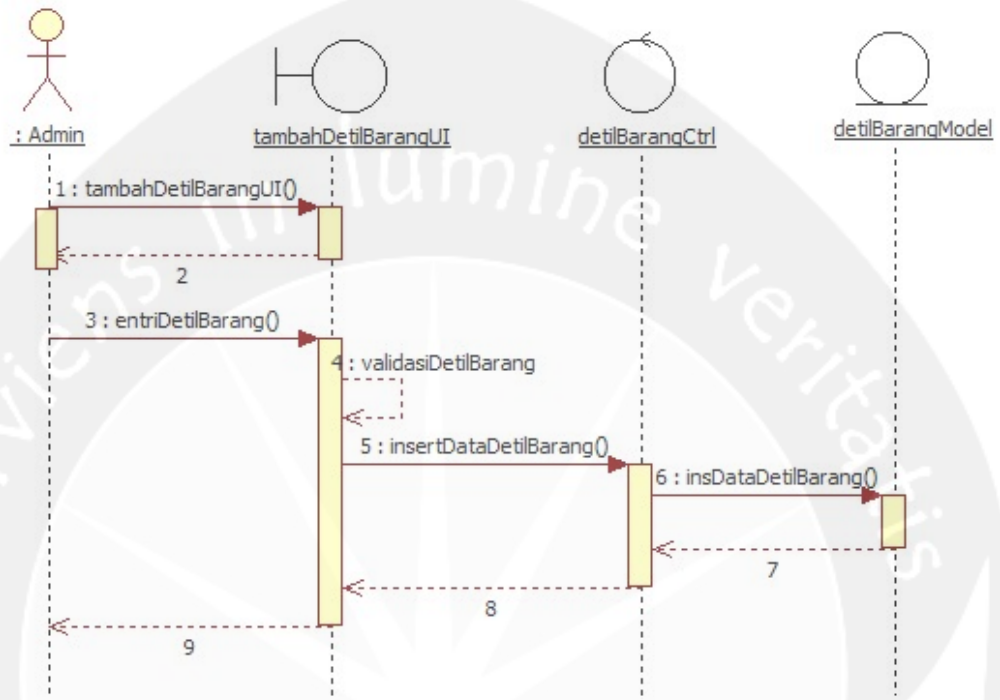
2.2.7.5 Tampil Data Barang



Gambar 2. 27 Sequence Diagram Tampil Data Barang

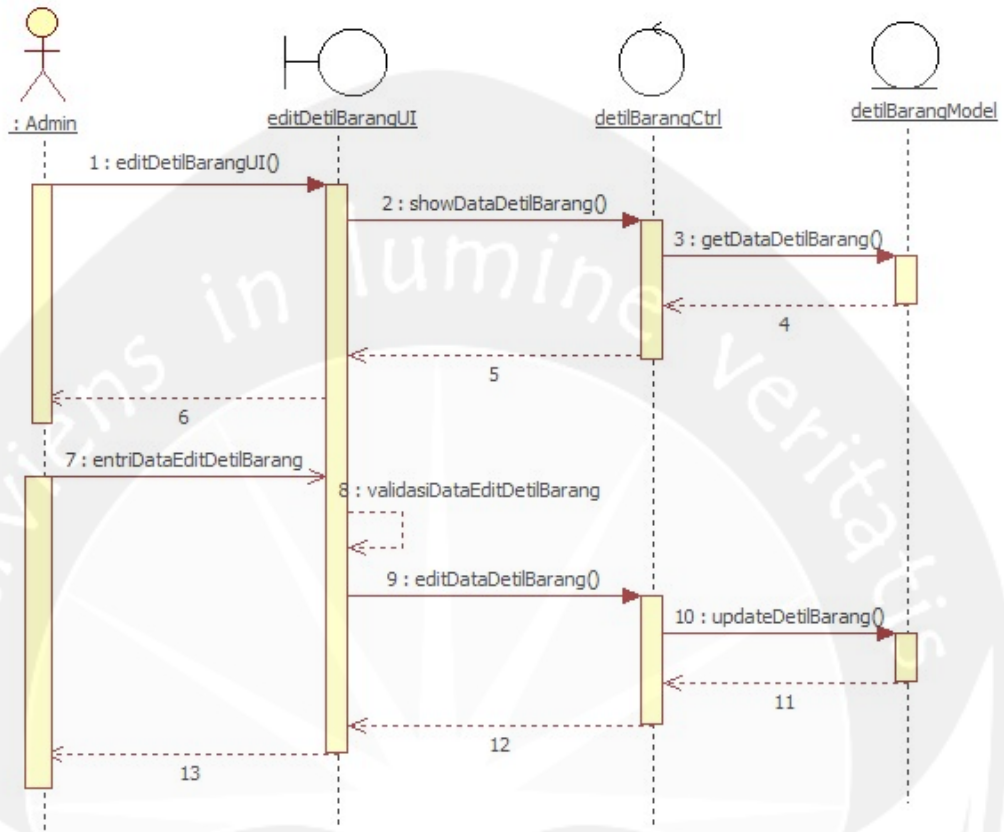
2.2.8 Mengelola Data Detil Barang

2.2.8.1 Tambah Data Detil Barang



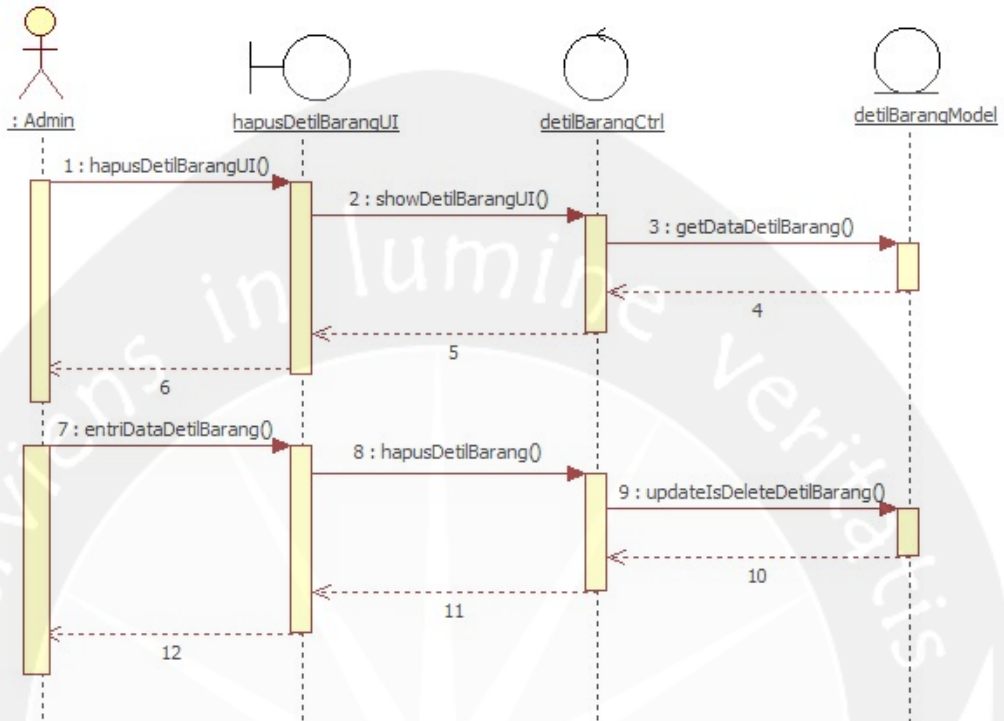
Gambar 2. 28 Sequence Diagram Tambah Data Detil Barang

2.2.8.2 Ubah Data Detil Barang



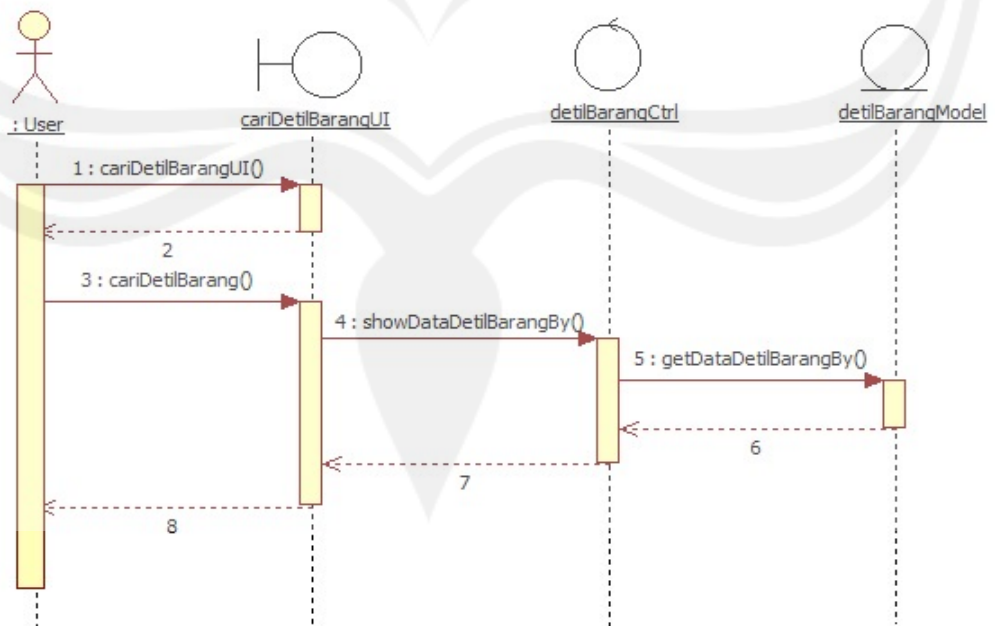
Gambar 2. 29 Sequence Diagram Ubah Data Detil Barang

2.2.8.3 Hapus Data Detil Barang



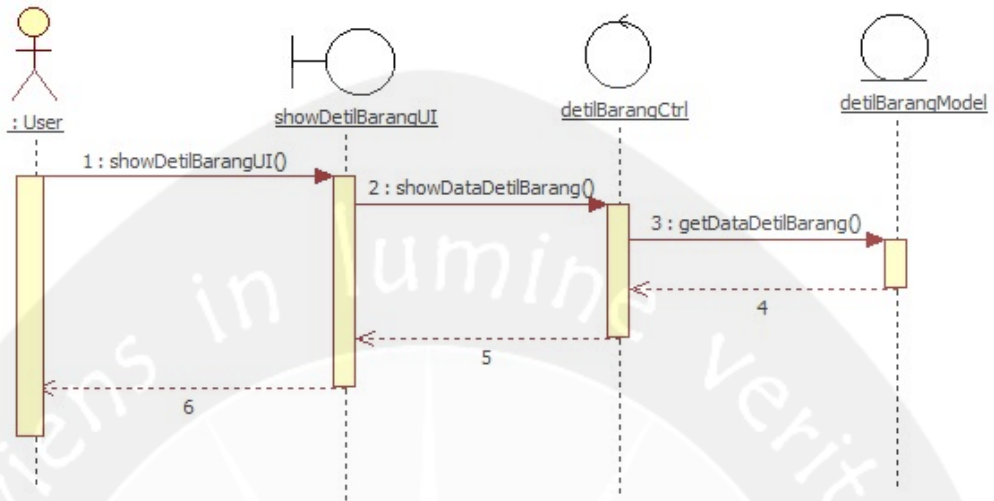
Gambar 2. 30 Sequence Diagram Hapus Data Detil Barang

2.2.8.4 Cari Data Detil Barang



Gambar 2. 31 Sequence Diagram Cari Data Detil Barang

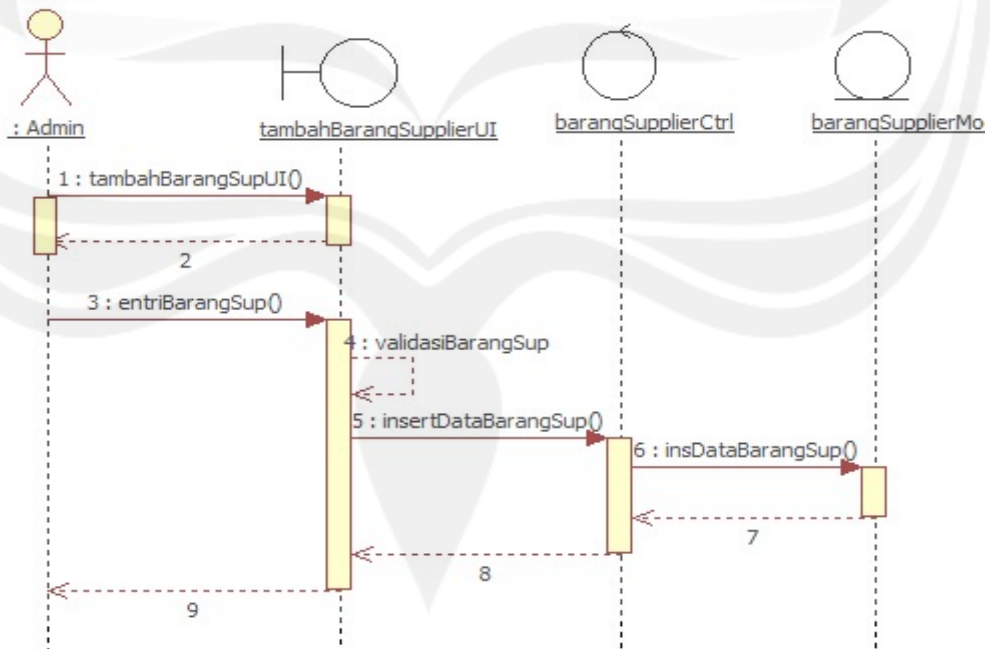
2.2.8.5 Tampil Data Detil Barang



Gambar 2. 32 Sequence Diagram Tampil Data Detil Barang

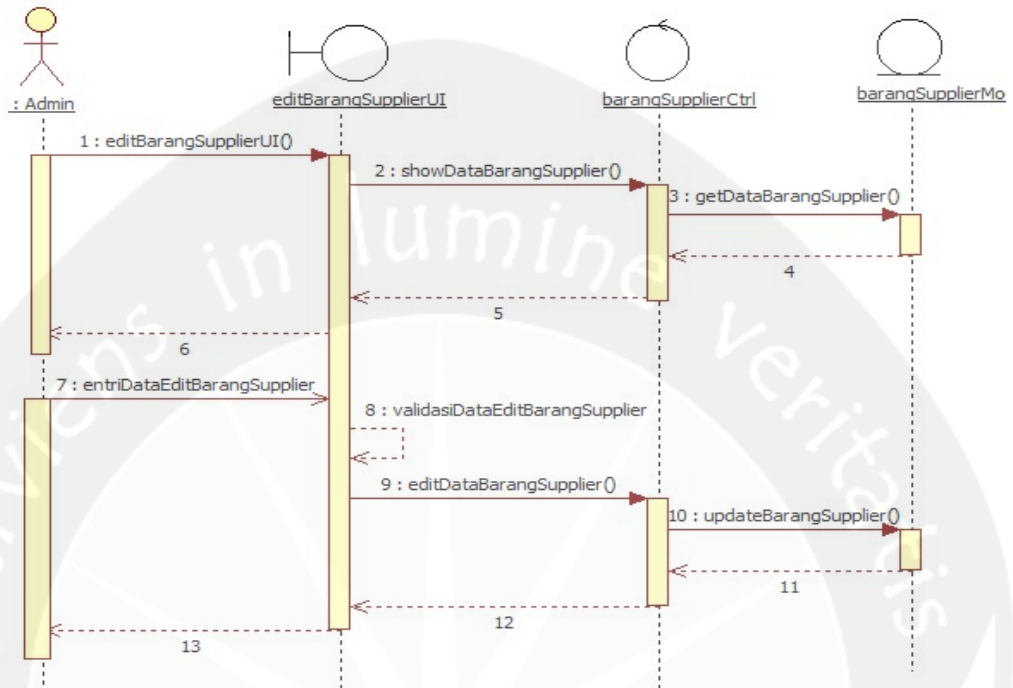
2.2.9 Mengelola Barang Supplier

2.2.9.1 Tambah Data Barang Supplier



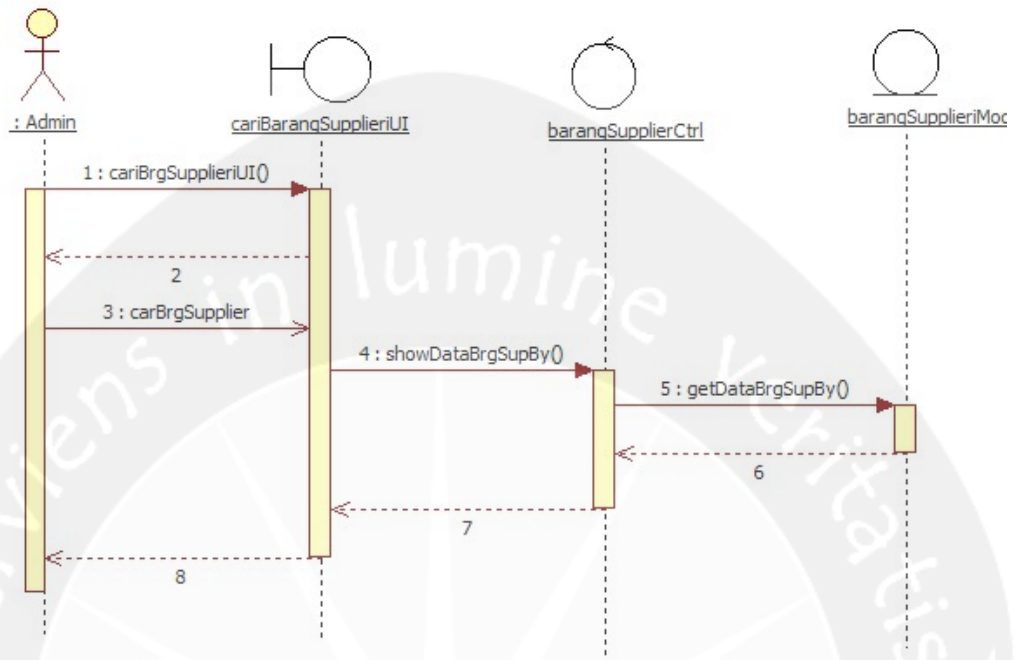
Gambar 2. 33 Sequence Diagram Tambah Data Barang Supplier

2.2.9.2 Ubah Data Barang Supplier



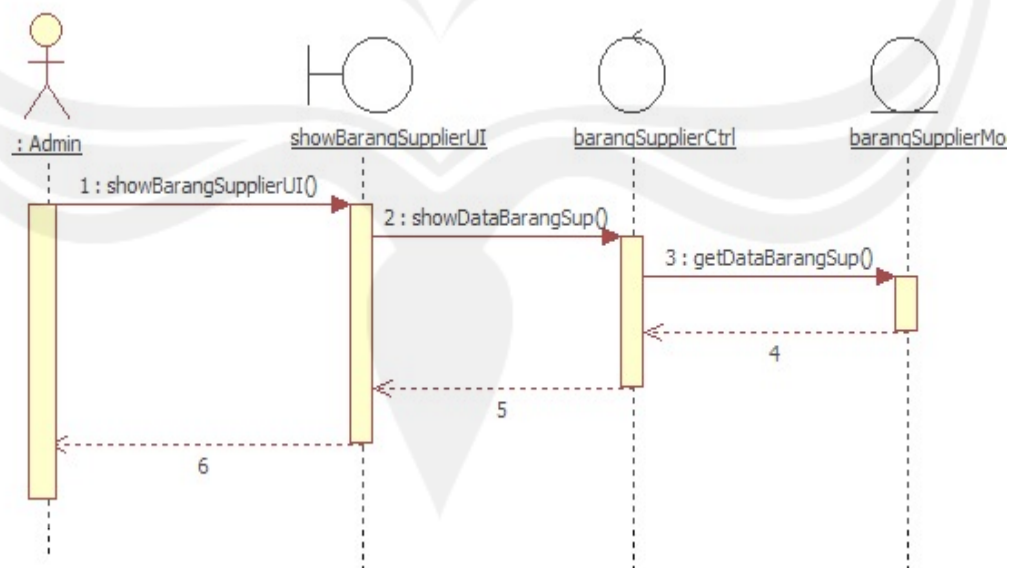
Gambar 2. 34 Ubah Data Barang Supplier

2.2.9.3 Cari Data Barang Supplier



Gambar 2. 35 Cari Data Barang Supplier

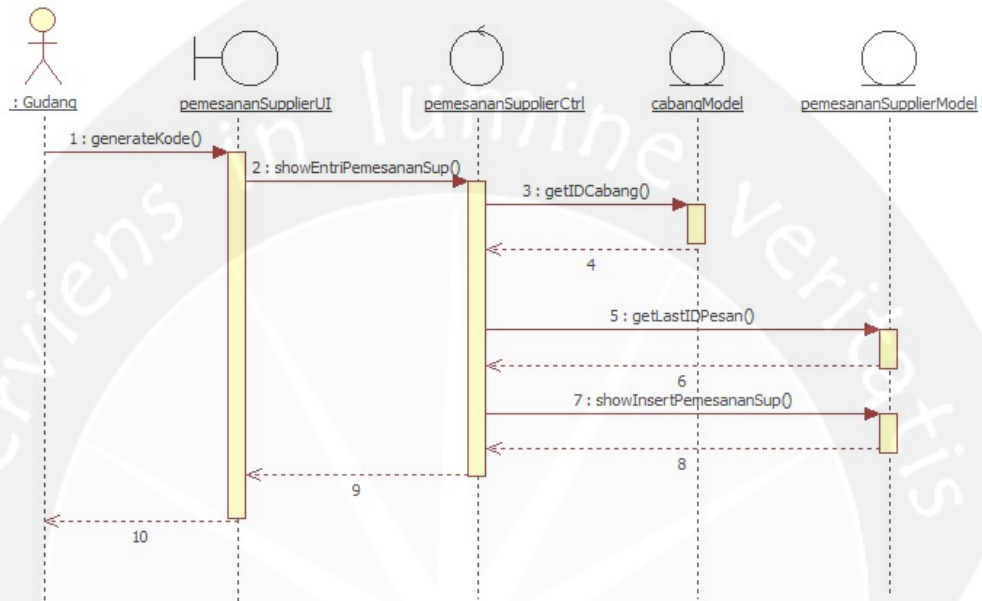
2.2.9.4 Tampil Data Barang Supplier



Gambar 2. 36 Tampil Data Barang Supplier

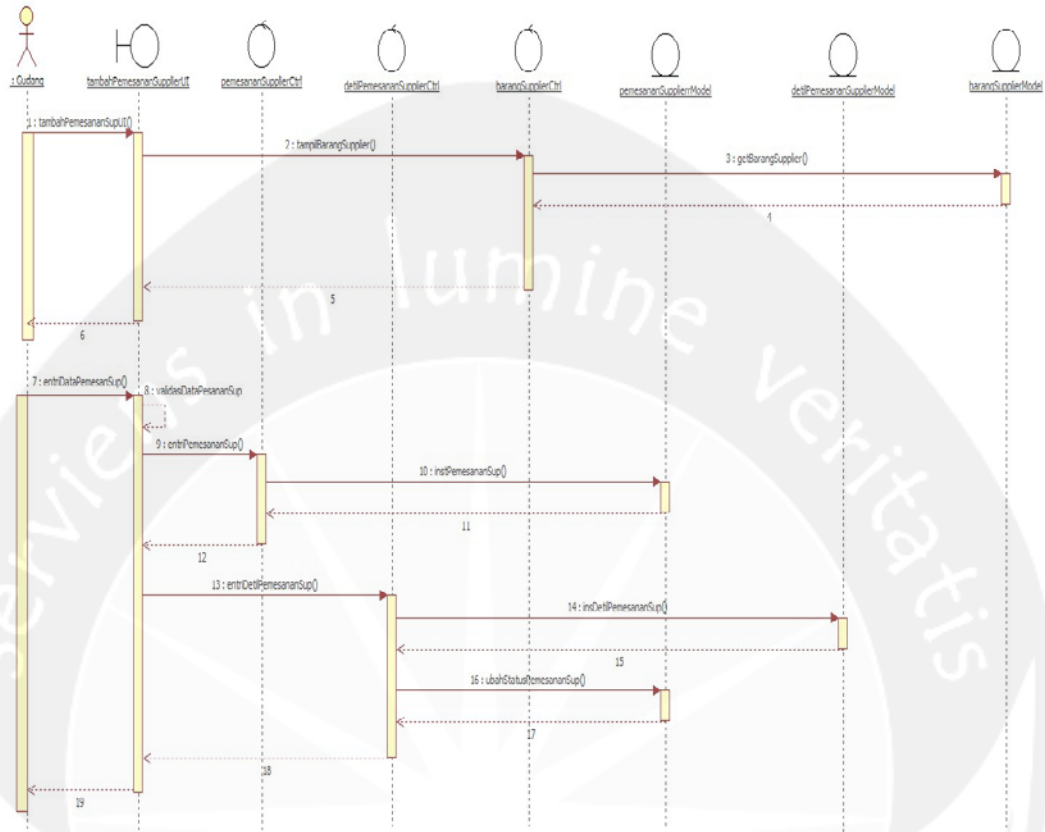
2.2.10 Mengelola Data Pemesanan Barang ke Supplier

2.2.10.1 Generate Kode Pemesanan Barang ke Supplier



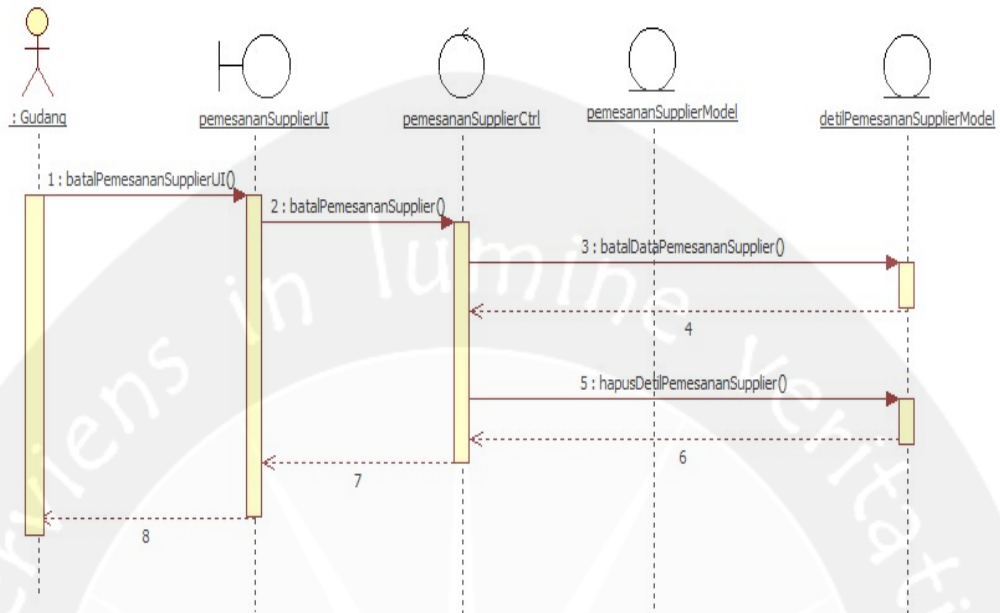
Gambar 2. 37 Sequence Diagram Generate Kode Pemesanan Barang Supplier

2.2.10.2 Tambah Data Pemesanan Barang ke Supplier



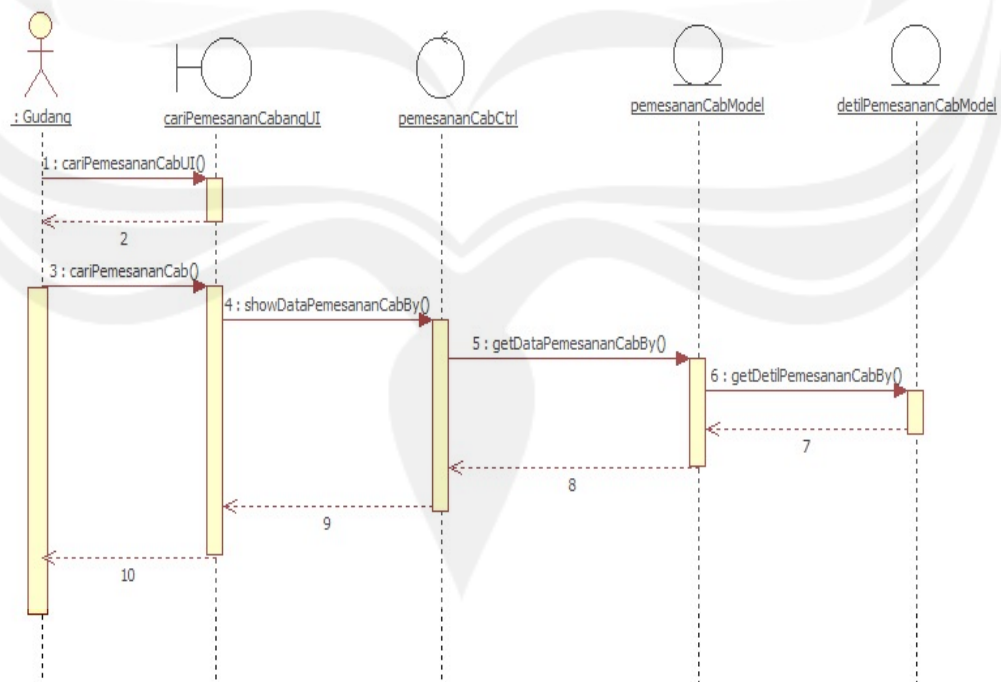
Gambar 2. 38 Sequence Diagram Tambah Data Pemesanan Barang Supplier

2.2.10.3 Batal Pemesanan Barang ke Supplier



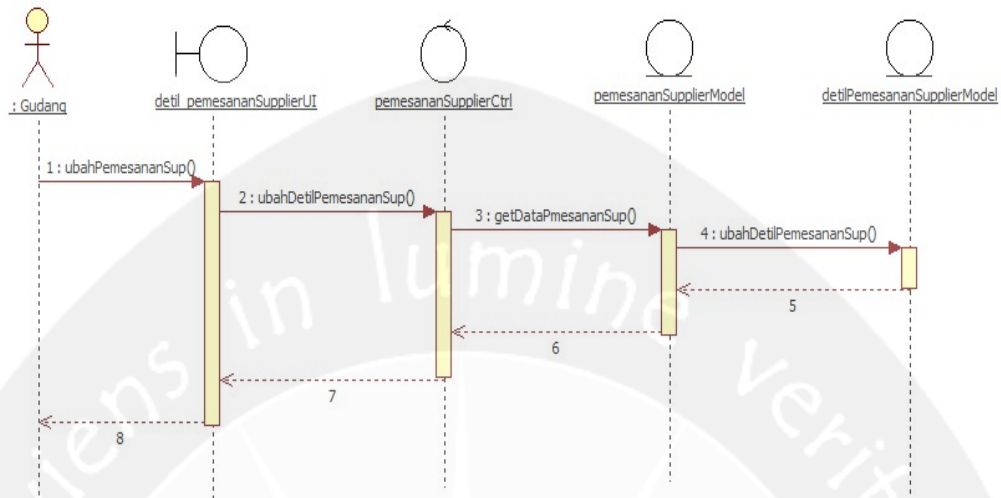
Gambar 2. 39 Sequence Diagram Batal Pemesanan Barang ke Supplier

2.2.10.4 Cari Pemesanan Barang ke Supplier



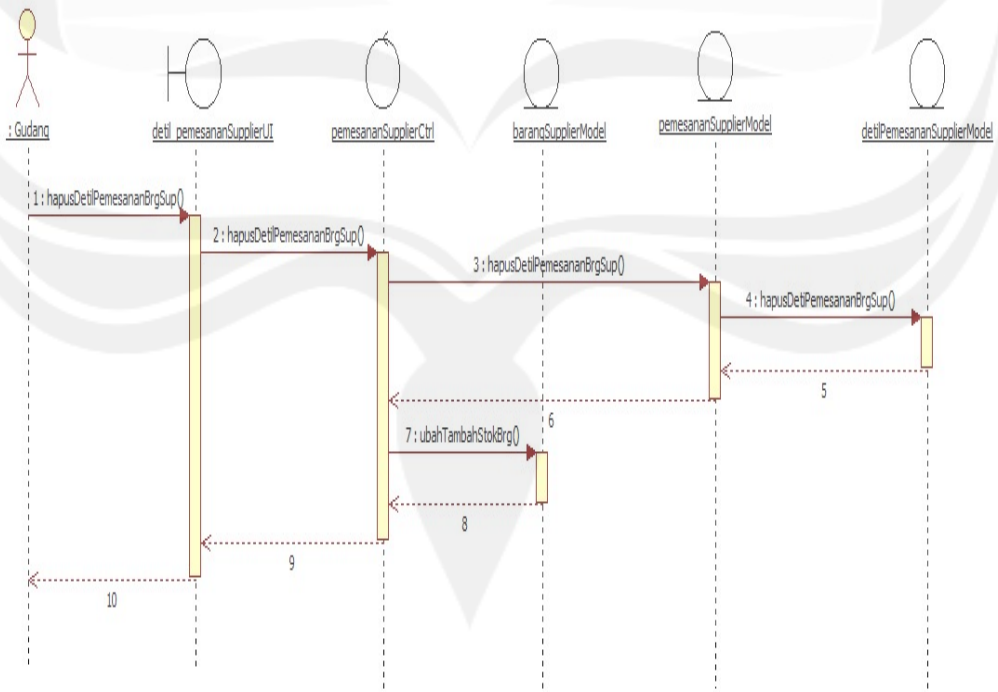
Gambar 2. 40 Sequence Diagram Cari Data Pemesanan Barang ke Supplier

2.2.10.5 Ubah Detil Pemesanan Barang ke Supplier



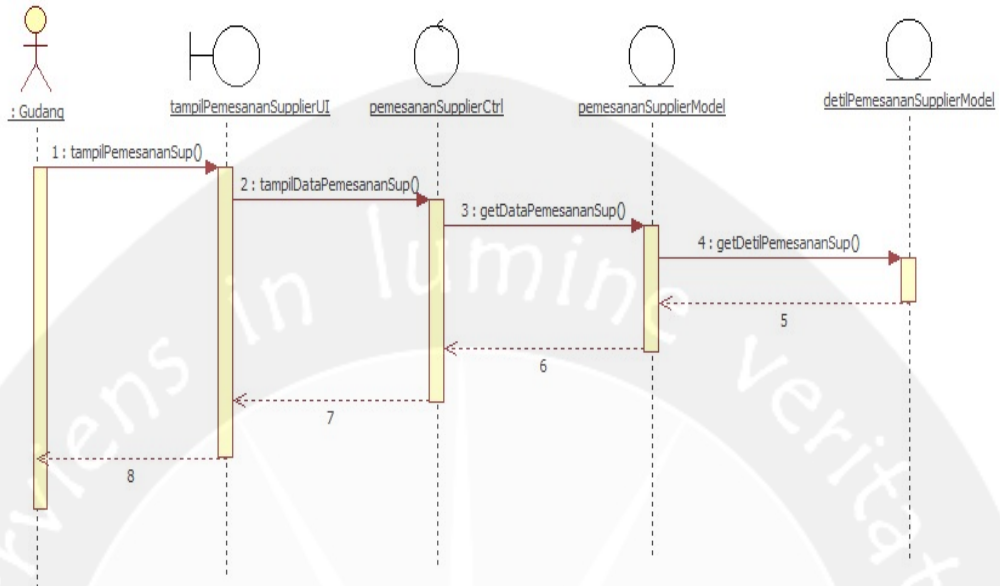
Gambar 2. 41 Sequence Diagram Ubah Detil Pemesanan Barang

2.2.10.6 Hapus Detil Pemesanan Barang ke Supplier



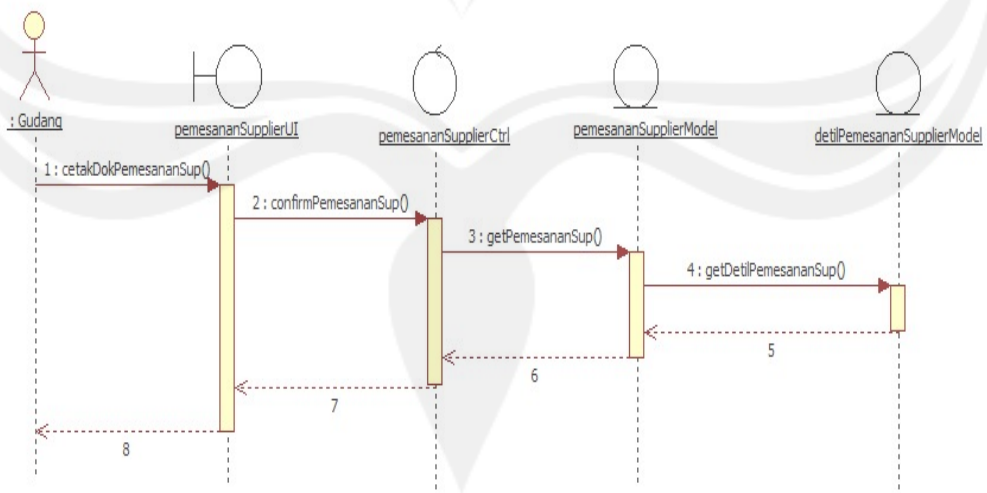
Gambar 2. 42 Sequence Diagram Hapus Detil Pemesanan Barang

2.2.10.7 Tampil Pemesanan Barang ke Supplier



Gambar 2. 43 Sequence Diagram Tampil Data Pemesanan Barang ke Supplier

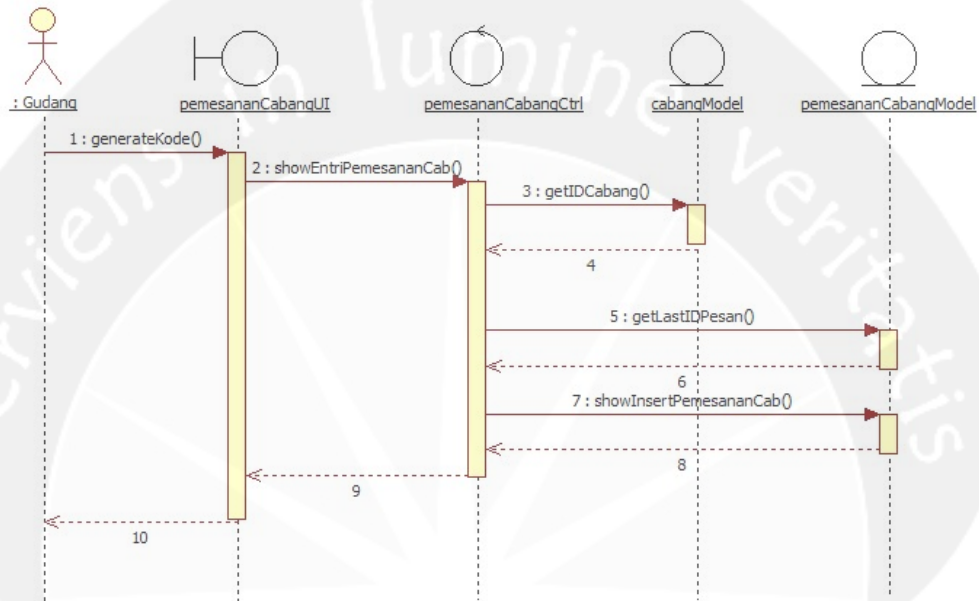
2.2.10.8 Cetak Dokumen Pemesanan Barang ke Supplier



Gambar 2. 44 Sequence Diagram Cetak Dokumen Pemesanan Barang ke Supplier

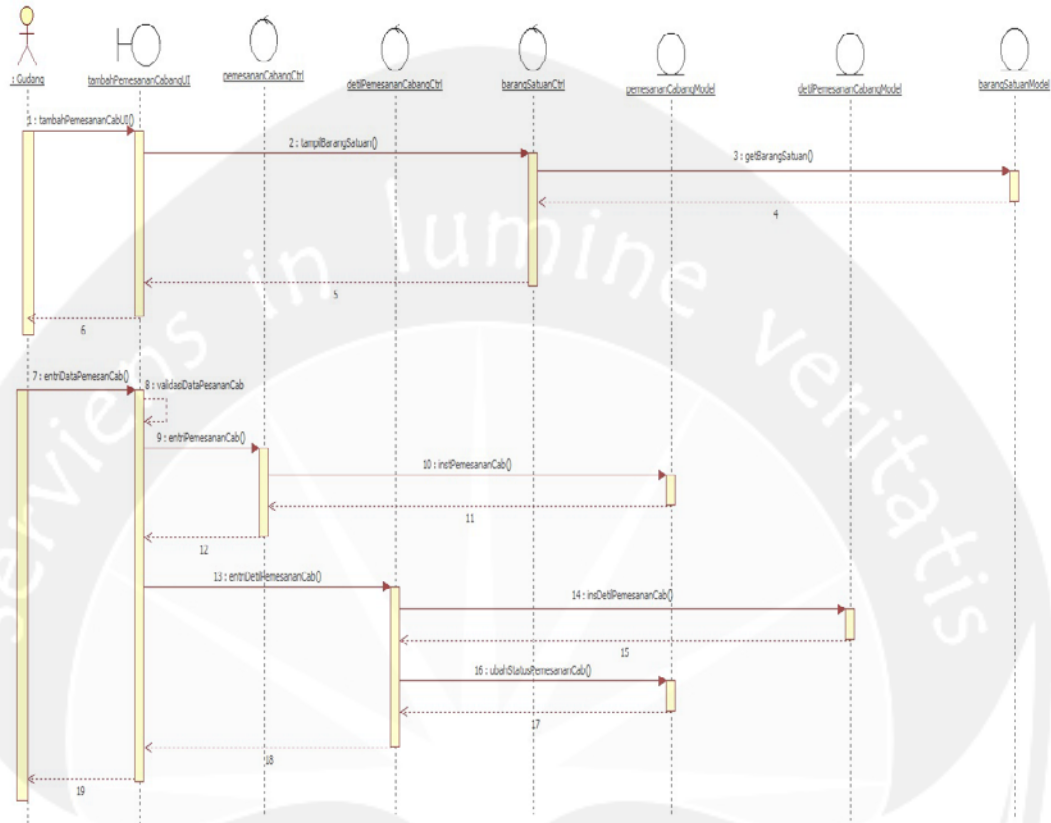
2.2.11 Mengelola Data Pemesanan Barang ke Pusat/Antar Cabang

2.2.11.1 Generate Kode Pemesanan Barang ke Pusat/Antar Cabang



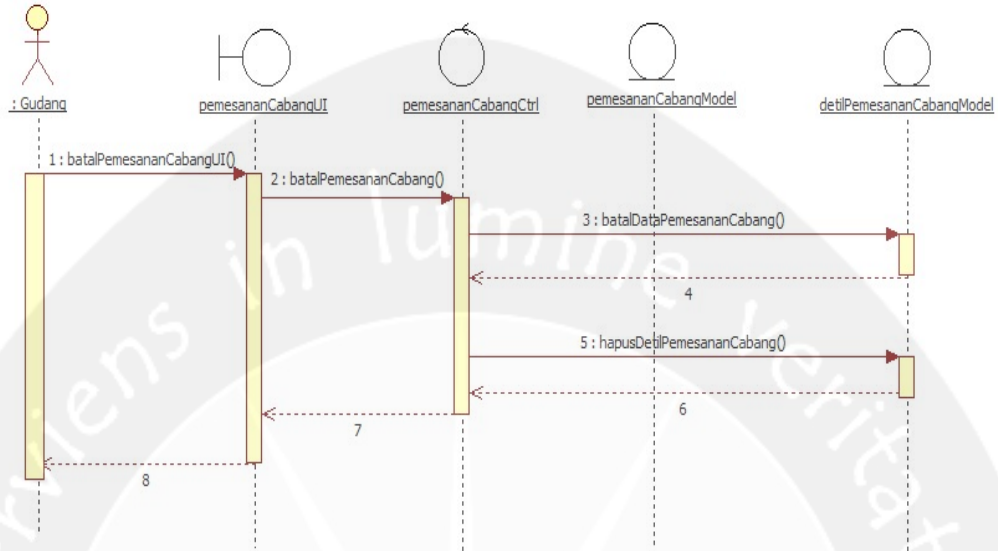
Gambar 2. 45 Sequence Diagram Generate Kode Pemesanan Barang ke Pusat/Antar Cabang

2.2.11.2 Tambah Data Pemesanan Barang ke Pusat/Antar Cabang



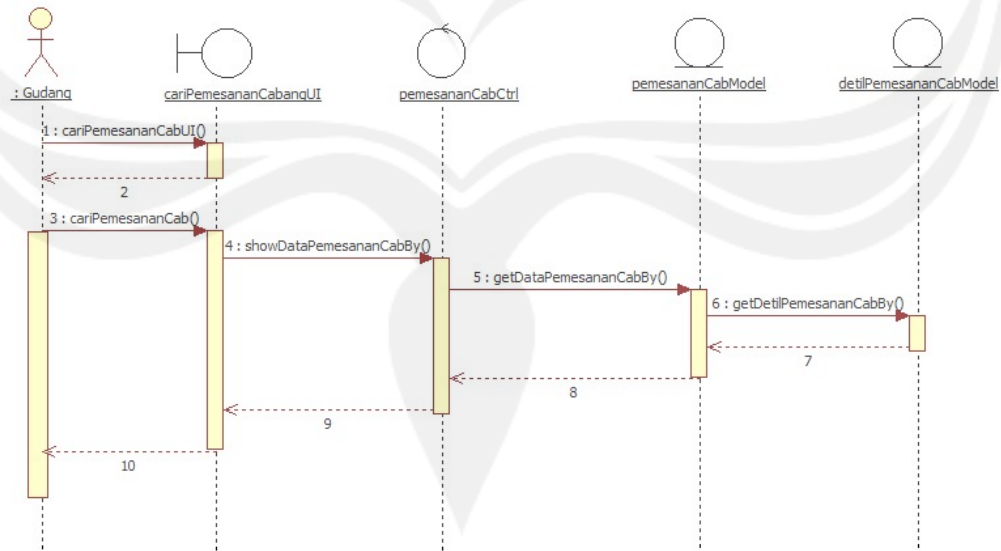
Gambar 2. 46 Sequence Diagram Tambah Data Pemesanan Barang ke Pusat/Antar Cabang

2.2.11.3 Batal Pemesanan Barang ke Pusat/Antar Cabang



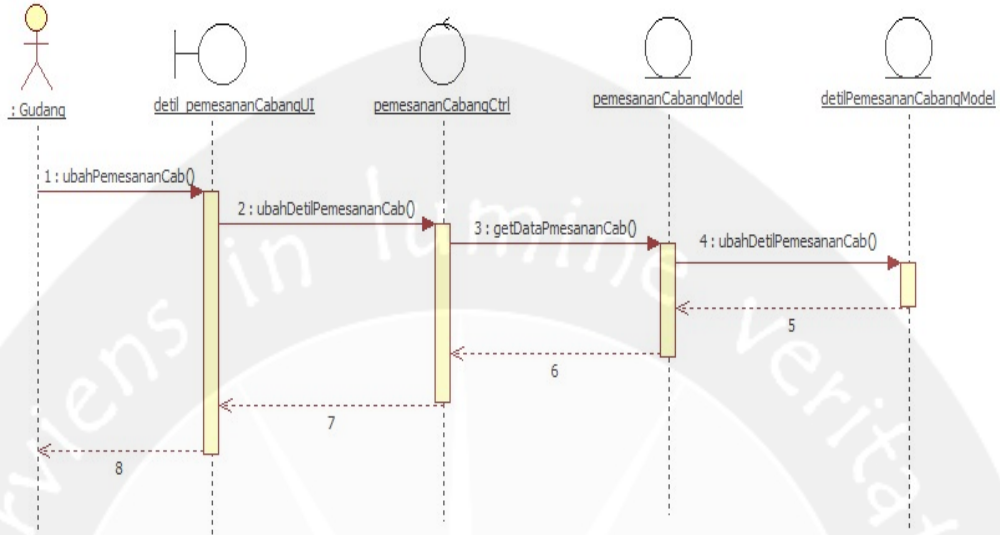
Gambar 2. 47 Sequence Diagram Batal Pemesanan Barang ke Pusat/Antar Cabang

2.2.11.4 Cari Pemesanan Barang ke Pusat/Antar Cabang



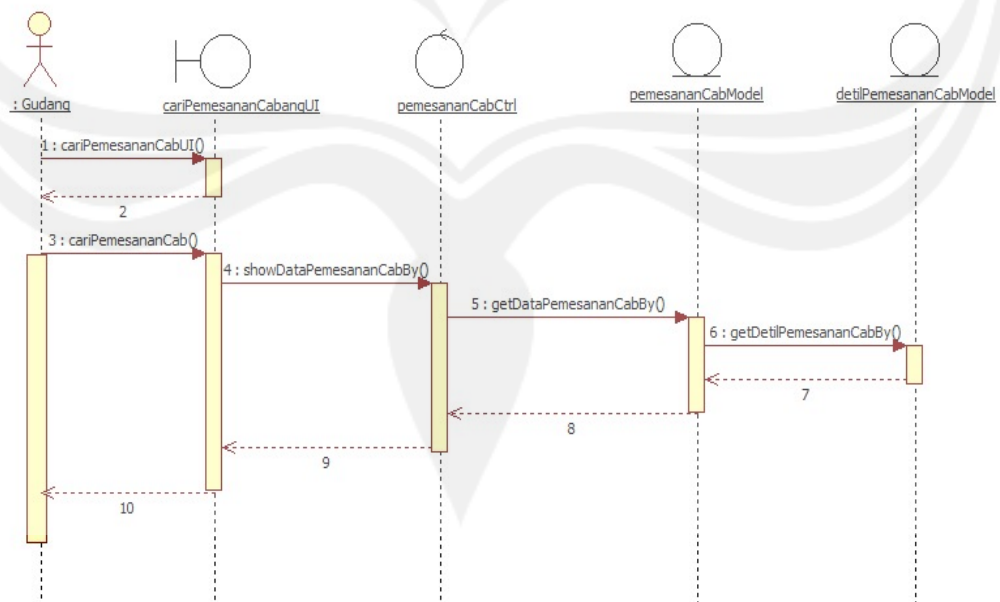
Gambar 2. 48 Sequence Diagram Cari Data Pemesanan Barang ke Pusat/Antar Cabang

2.2.11.5 Ubah Detil Pemesanan Barang ke Pusat/Antar Cabang



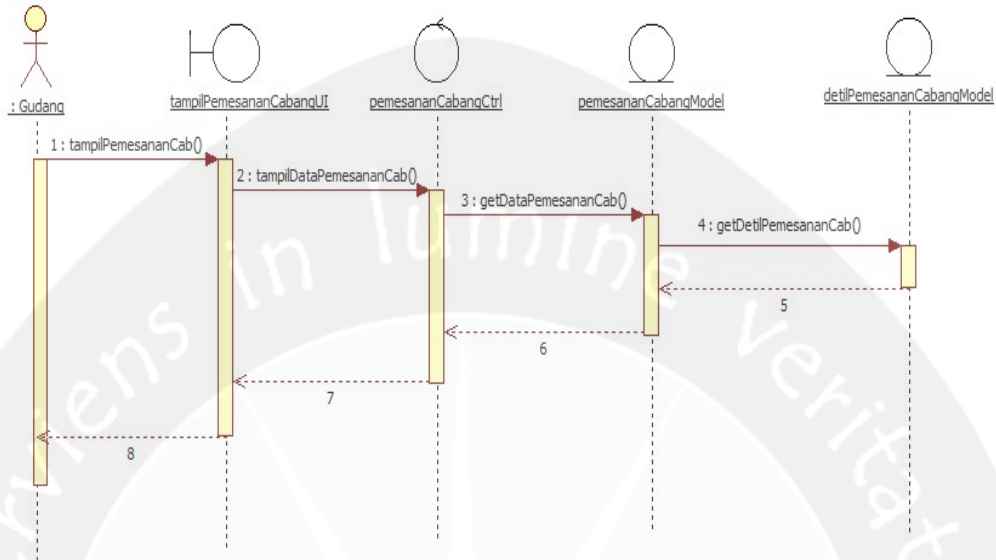
Gambar 2. 49 Sequence Diagram Ubah Detil Pemesanan Barang

2.2.11.6 Hapus Detil Pemesanan Barang ke Pusat/Antar Cabang



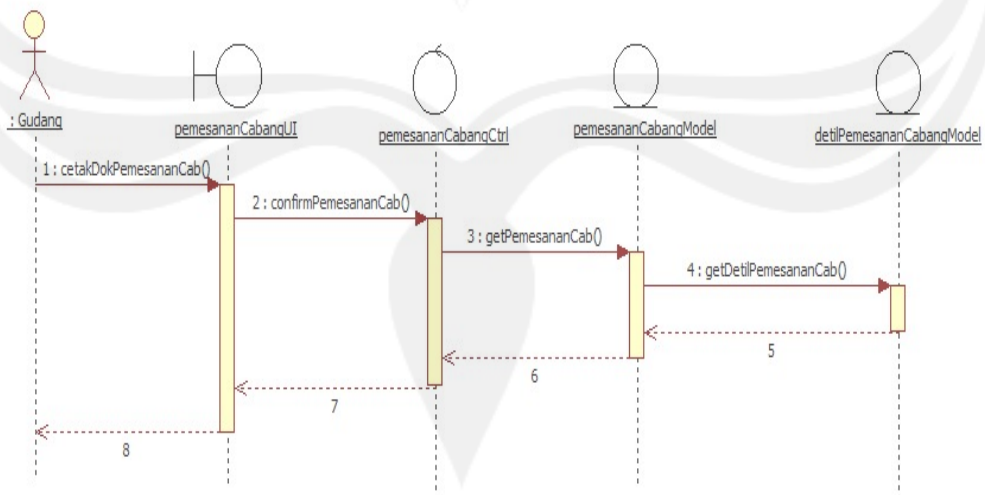
Gambar 2. 50 Sequence Diagram Hapus Detil Pemesanan Barang

2.2.11.7 Tampil Pemesanan Barang ke Pusat/Supplier



Gambar 2. 51 Sequence Diagram Tampil Data Pemesanan Barang ke Pusat/Antar Cabang

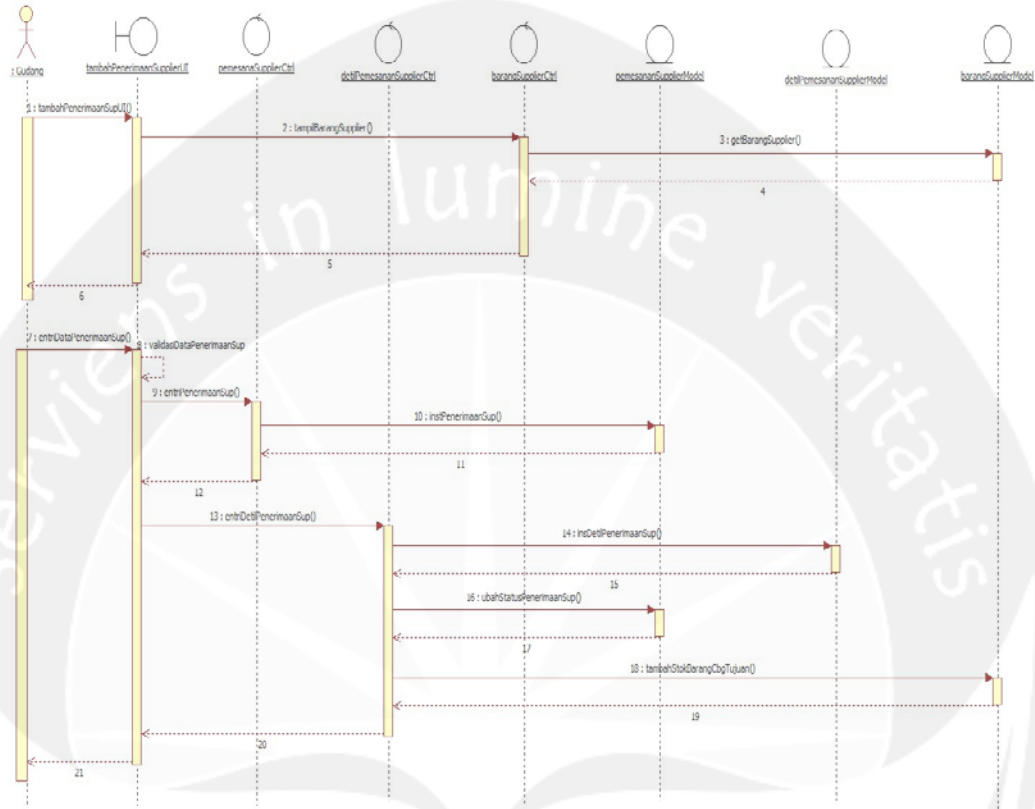
2.2.11.8 Cetak Dokumen Pemesanan Barang ke Pusat/Antar Cabang



Gambar 2. 52 Sequence Diagram Tampil Data Pemesanan Barang ke Pusat/Antar Cabang

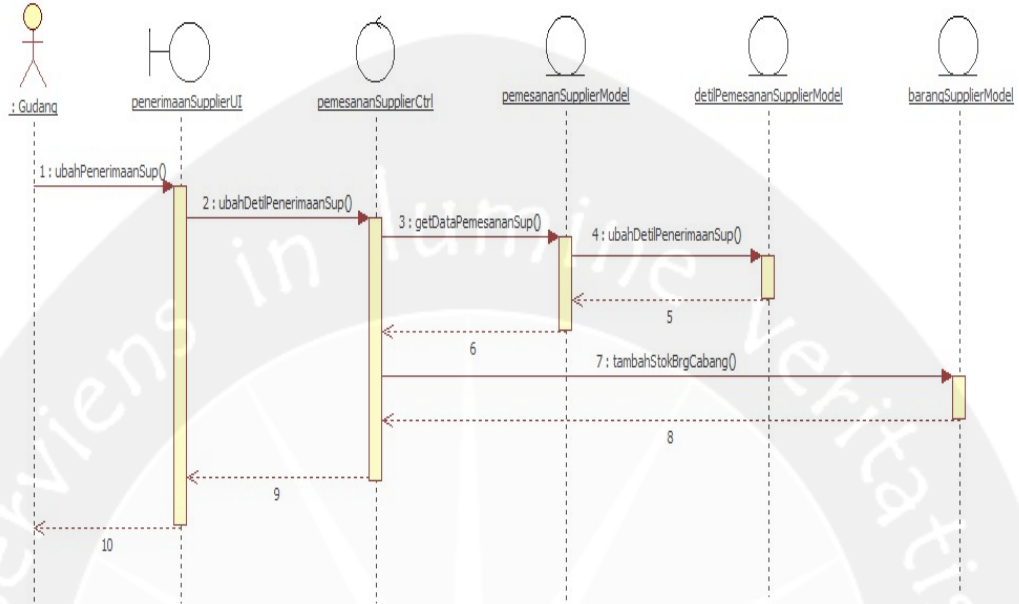
2.2.12 Mengelola Penerimaan Barang dari Supplier

2.2.12.1 Tambah Penerimaan Barang dari Supplier



Gambar 2. 53 Sequence Diagram Tambah Data Penerimaan Barang dari Supplier

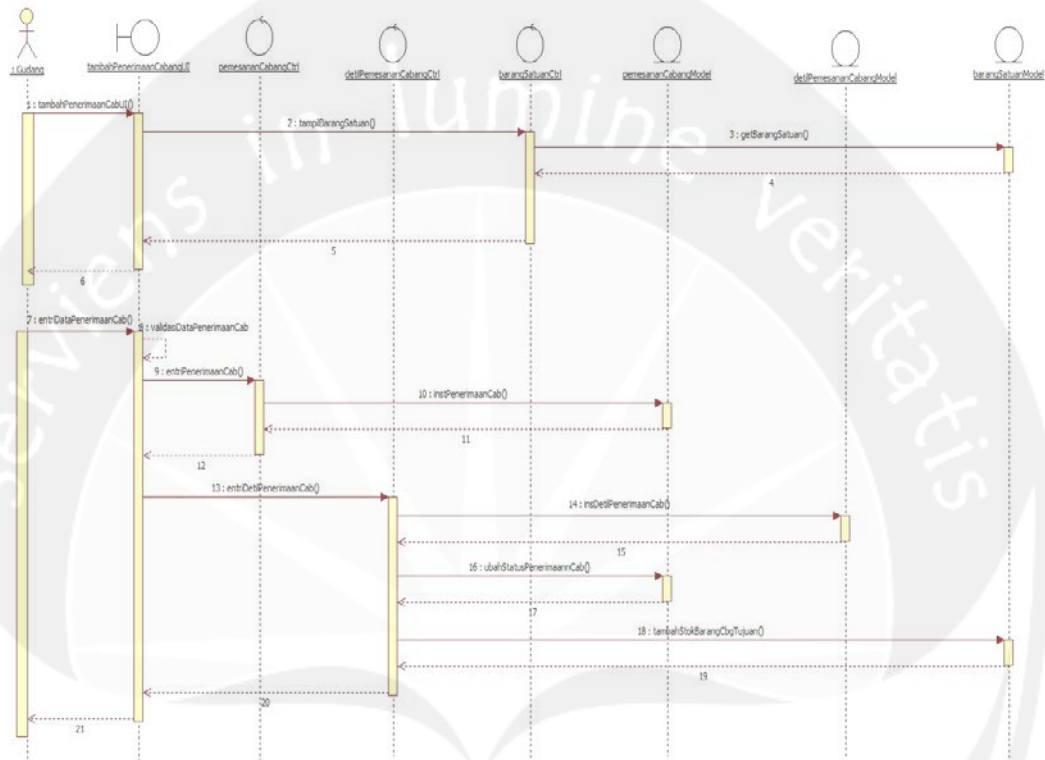
2.2.12.1 Ubah Detil Penerimaan Barang dari Supplier



Gambar 2. 54 Sequence Diagram Ubah Data Penerimaan Barang dari Supplier

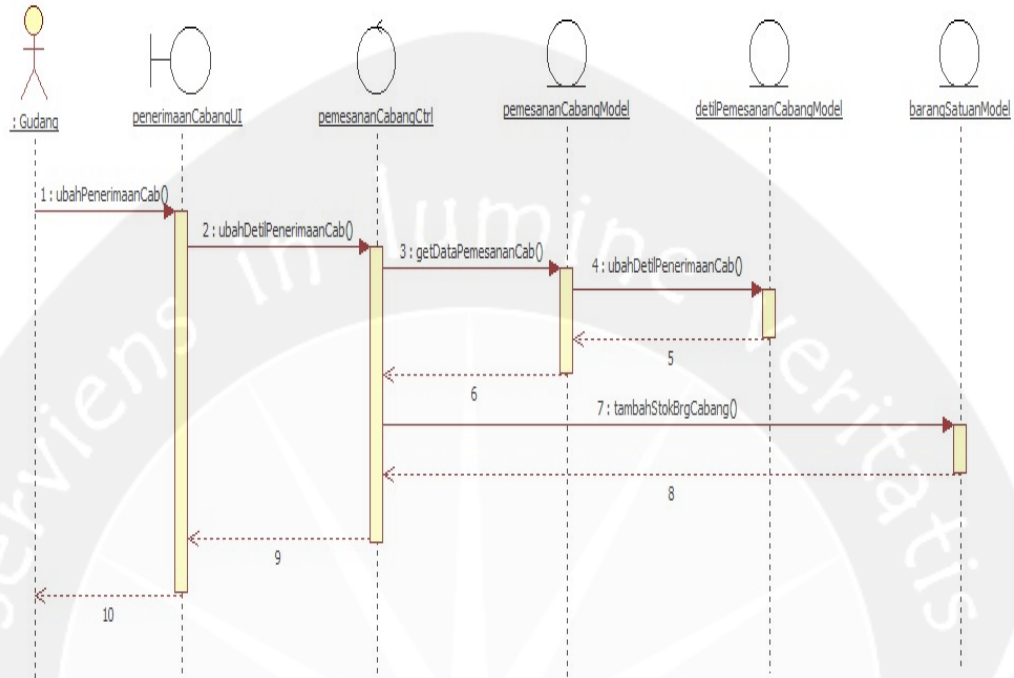
2.2.13 Mengelola Penerimaan Barang dari Pusat/Antar Cabang

2.2.13.1 Tambah Penerimaan Barang dari Pusat/Antar Cabang



Gambar 2. 55 Sequence Diagram Tambah Data Penerimaan Barang dari Pusat/Antar Cabang

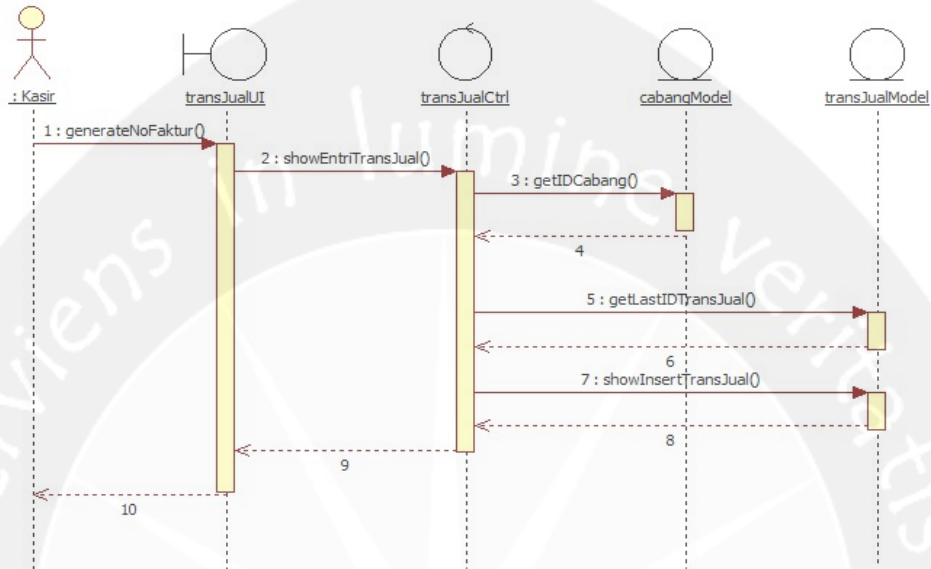
2.2.13.2 Ubah Detil Penerimaan Barang dari Pusat/Antar Cabang



Gambar 2. 56 Sequence Diagram Ubah Data Penerimaan Barang dari Pusat/Antar Cabang

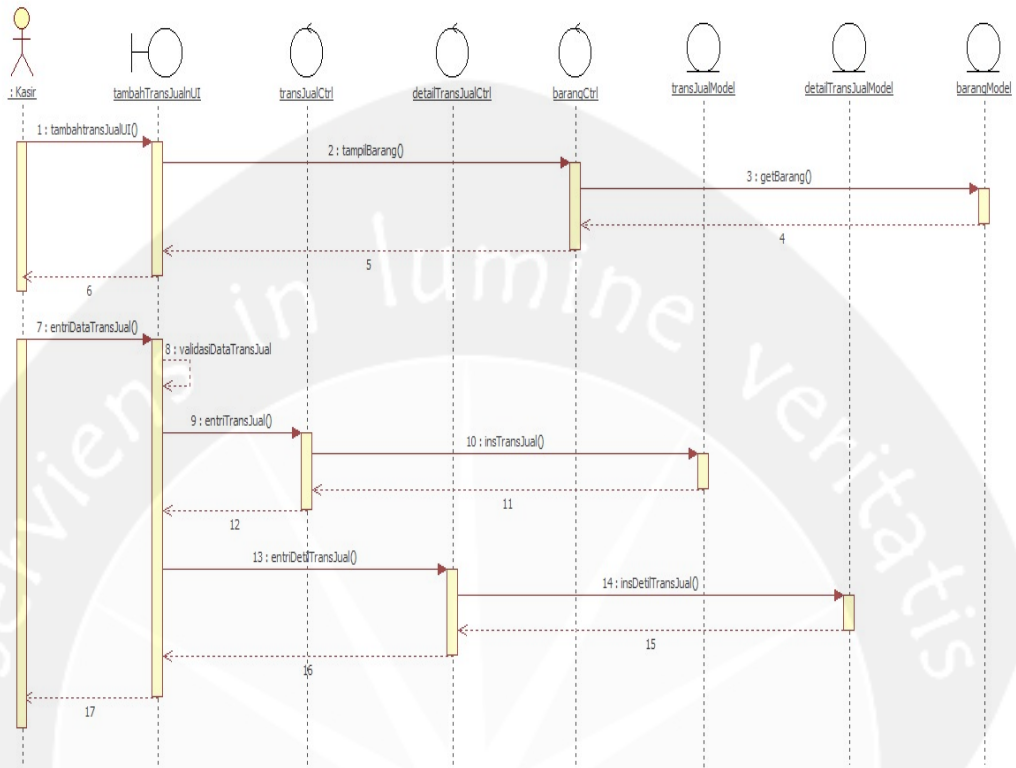
2.2.14 Mengelola Transaksi Penjualan

2.2.14.1 Generate No Faktur Transaksi Penjualan



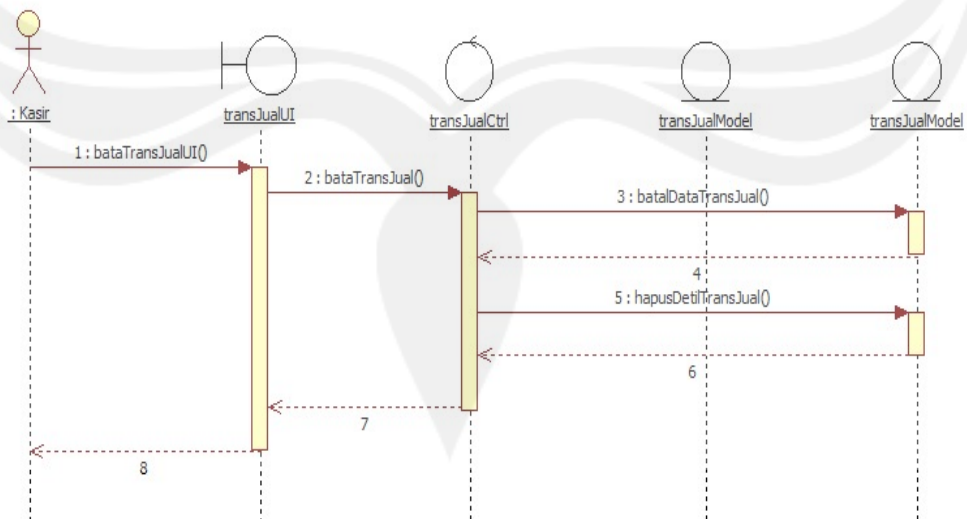
Gambar 2. 57 Sequence Diagram Generate No. Faktur Transaksi Penjualan

2.2.14.2 Tambah Transaksi Penjualan



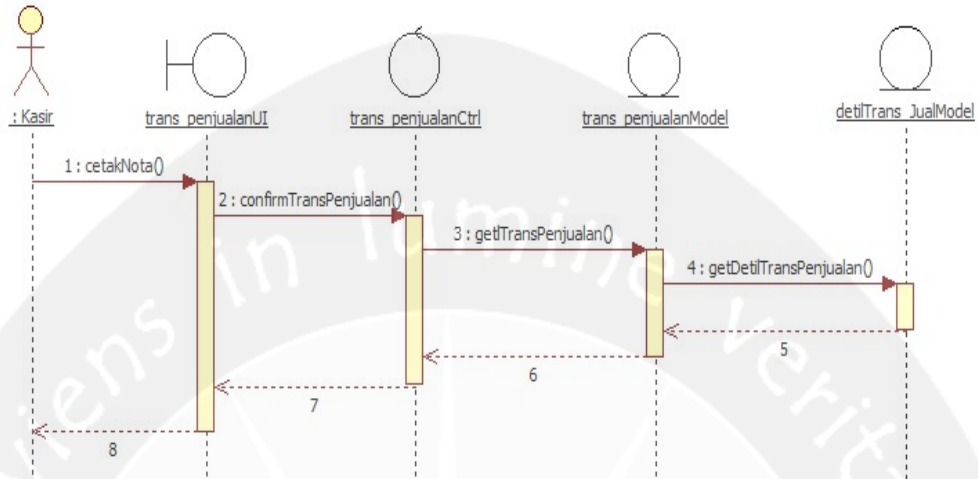
Gambar 2. 58 Sequence Diagram Tambah Transaksi Penjualan

2.2.14.3 Batal Transaksi Penjualan



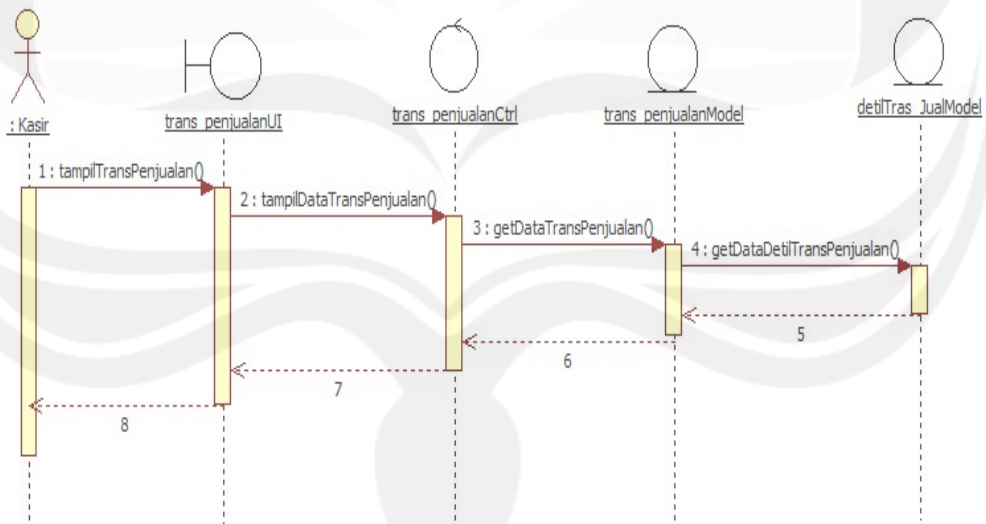
Gambar 2. 59 Sequence Diagram Batal Transaksi Penjualan

2.2.14.4 Cetak Nota Transaksi Penjualan



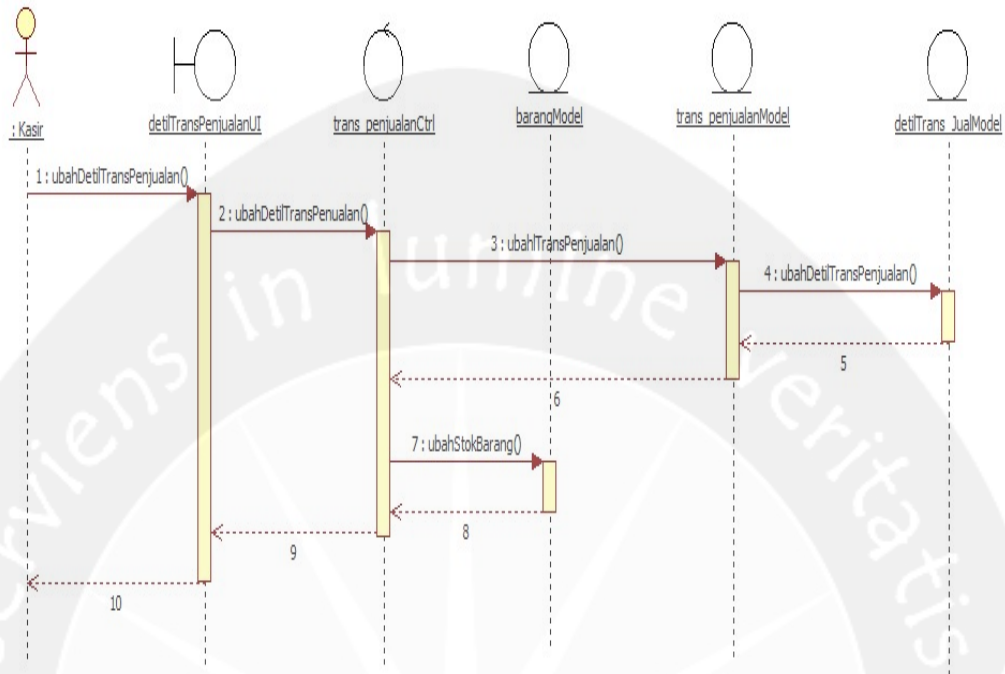
Gambar 2. 60 Sequence Diagram Cetak Nota Transaksi Penjualan

2.2.14.5 Tampil Transaksi Penjualan



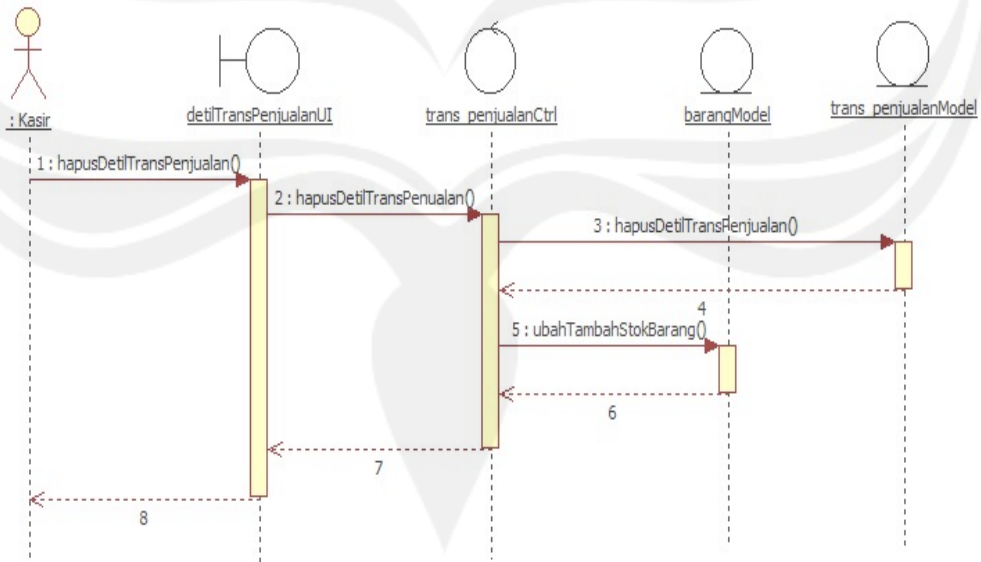
Gambar 2. 61 Sequence Diagram Tampil Transaksi Penjualan

2.2.14.6 Ubah Detil Transaksi Penjualan



Gambar 2. 62 Sequence Diagram Ubah Detil Transaksi Penjualan

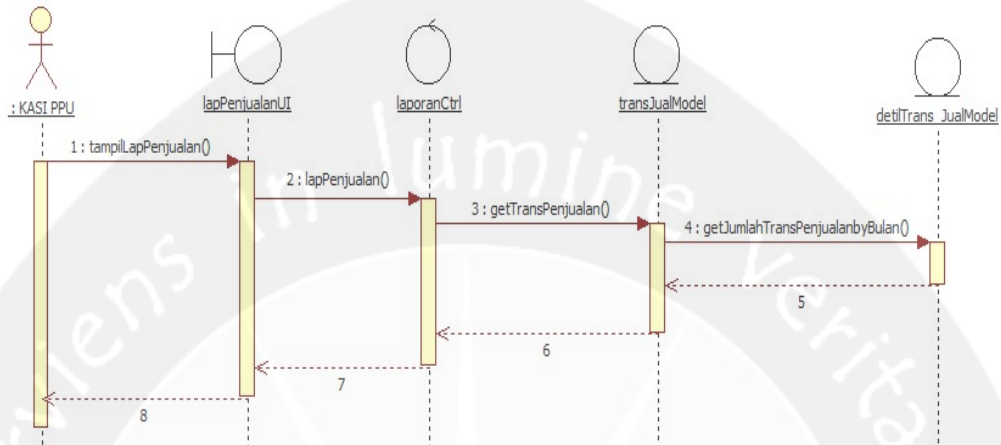
2.2.14.7 Hapus Detil Transaksi Penjualan



Gambar 2. 63 Sequence Diagram Hapus Detil Transaksi Penjualan

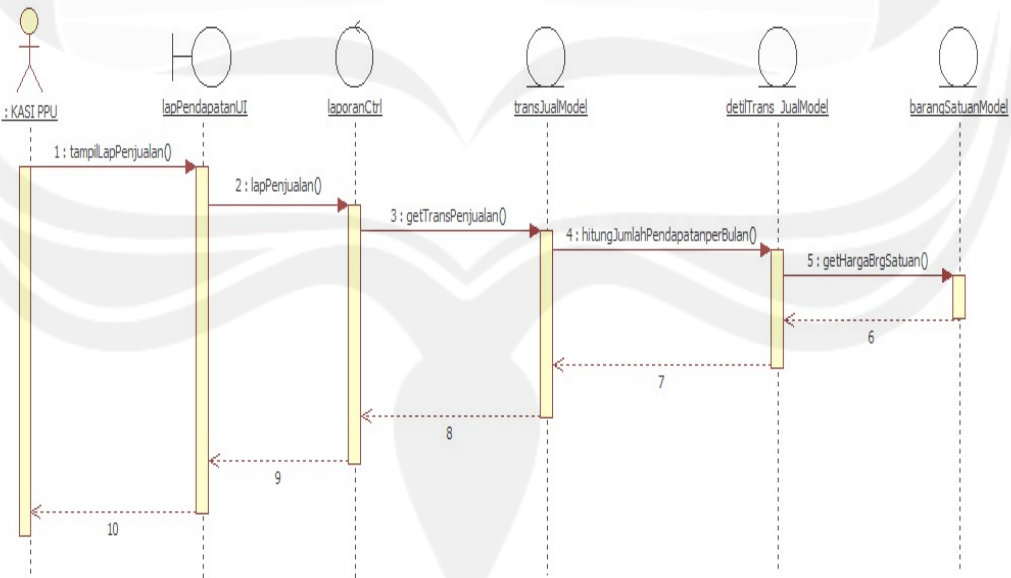
2.2.15 Reporting (Pelaporan)

2.2.15.1 Laporan Penjualan



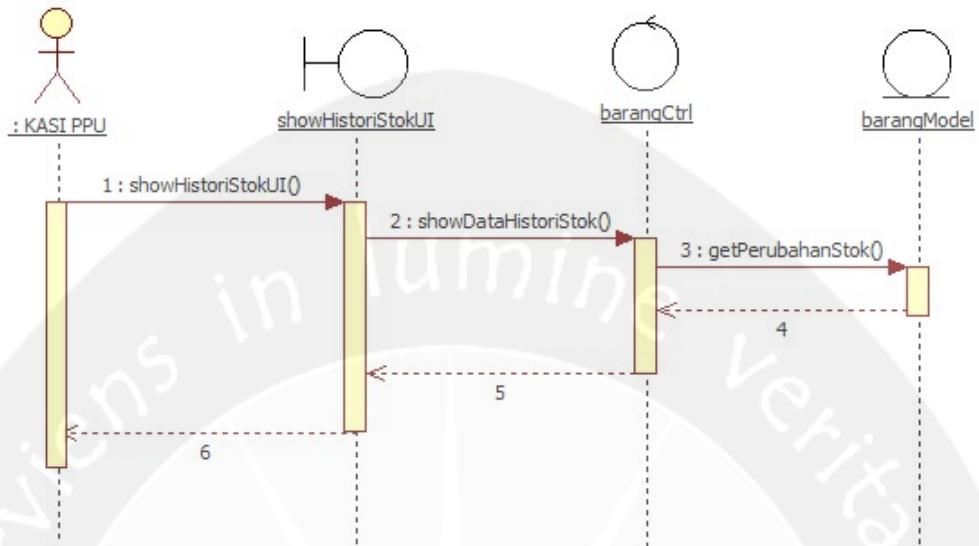
Gambar 2. 64 Sequence Diagram Laporan Penjualan

2.2.15.2 Laporan Pendapatan



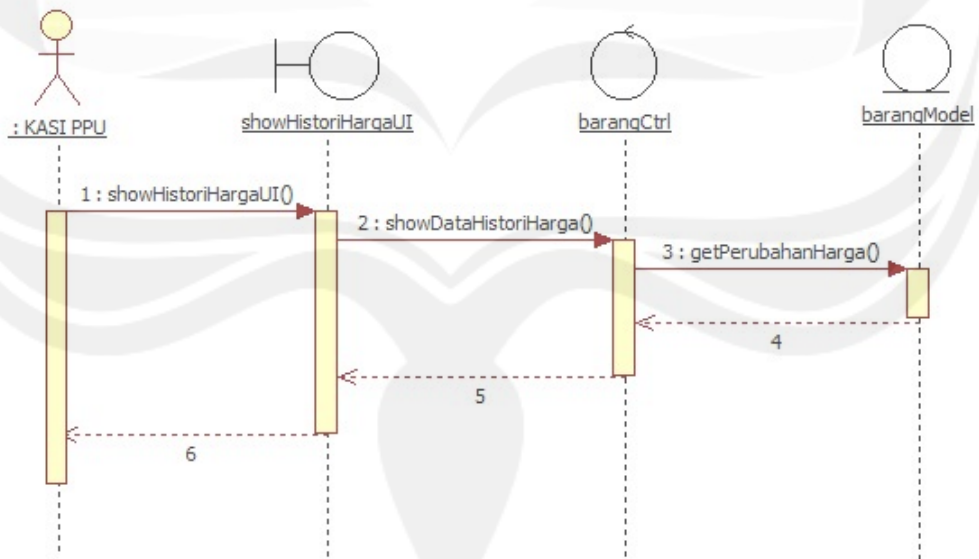
Gambar 2. 65 Sequence Diagram Laporan Pendapatan

2.2.15.3 Laporan Histori Stok Barang



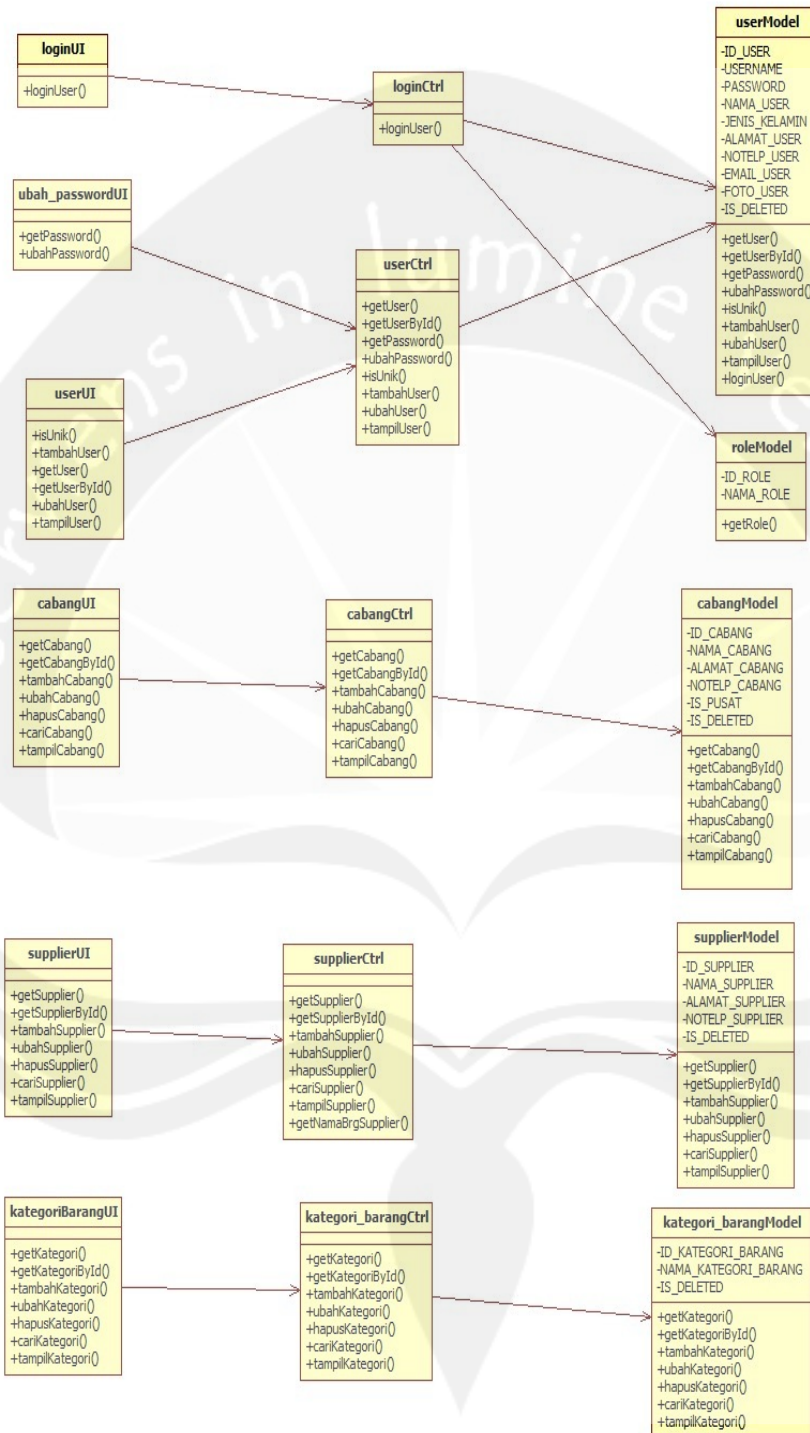
Gambar 2. 66 Sequence Diagram Laporan Histori Stok Barang

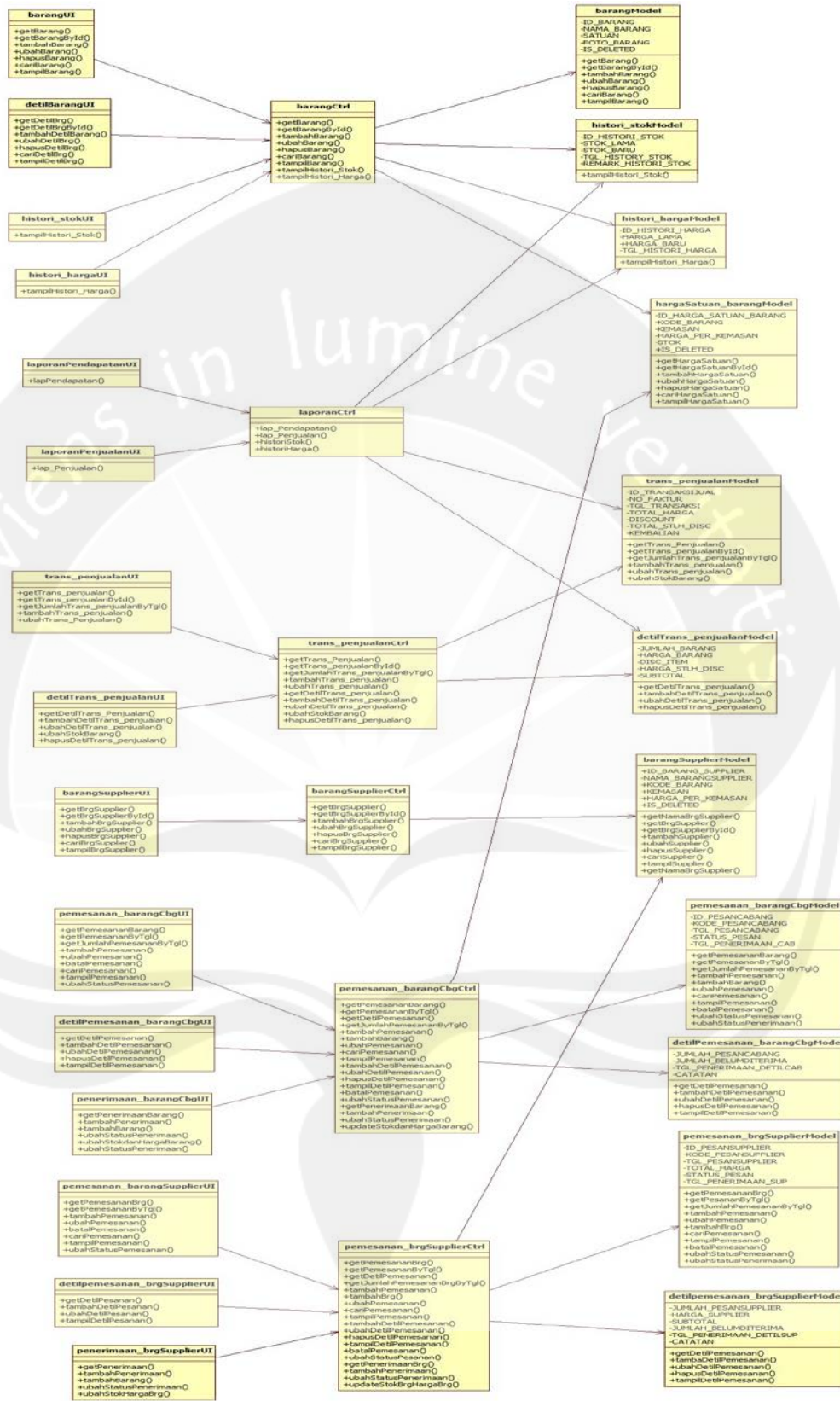
2.2.15.4 Histori Harga Barang



Gambar 2. 67 Sequence Diagram Laporan Histori Harga Barang

2.2.16 Class Diagram





Gambar 2. 68 Class Diagram SISIP

2.2.17 Class Diagram Specific Descriptions

2.2.17.1 Specific Design Class LoginDekstopUI

loginUI	<<boundary>>
+loginUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+loginUser(username, password) : bool Operasi ini digunakan untuk melakukan pengecekan apakah username dan password yang dimasukkan valid atau tidak sebelum memasuki sistem sesuai dengan role pengguna masing-masing.	

2.2.17.2 Specific Design Class ChangePassUI

Ubah_passwordUI	<<boundary>>
+ubah_passwordUI() Digunakan untuk memanggil UI untuk merubah password	
+getPass() : void Operasi ini digunakan untuk mengambil password dari pengguna	
+ubahPass(pass_lama,pass_baru,confirm_passBaru):void Operasi ini digunakan untuk merubah password pada Id yang bersangkutan.	

2.2.17.3 Specific Design Class insert_userUI

Insert_userUI	<<boundary>>
+insert_userUI() Default konstruktor dimana semua atribut pada class	

ini di inisialisasi.

+isUnik() : void

Operasi ini digunakan untuk melakukan pengecekan apakah username sudah terdaftar atau belum

+insertUser(user) : void

Operasi ini digunakan untuk menambah data user baru.

2.2.17.4 Specific Design Class UserUI

pengUserUI	<<boundary>>
<p>+userUI() Default konstruktor dimana semua atribut pada class ini di inisialisasi.</p> <p>+getUser() : Tabel Operasi ini digunakan untuk menampilkan data user</p> <p>+editDataUser(user) : void Operasi ini digunakan untuk mengedit data user yang ada di database.</p> <p>+tampilDataUser(user) : void Operasi ini digunakan untuk menampilkan data user yang ada di database.</p> <p>+cariDataUser() : void Operasi ini digunakan untuk mencari data user berdasarkan nama.</p> <p>+validasiDataUser() Operasi ini digunakan untuk memvalidasi apakah ada data yang belum diisikan oleh user.</p> <p>+resPass() : void Operasi ini digunakan untuk mengeset password pada user tertentu dengan default password(userpass).</p>	

2.2.17.5 Specific Design Class CabangUI

cabangUI	<<boundary>>
<pre> +cabangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDisplayCabang() : Tabel Operasi ini digunakan untuk menampilkan data cabang +getCabangById(id_cabang):void Operasi ini digunakan untuk mengambil data cabang yang dipergunakan untuk edit data cabang +entriDataCabang(cabang) : void Operasi ini digunakan untuk menyimpan data cabang dari form yang diinputkan oleh Admin. +editDataCabang(cabang) : void Operasi ini digunakan untuk mengedit data cabang yang ada di database. +hapusDataCabang(cabang) : void Operasi ini digunakan untuk menghapus semu data cabang yang ada di database. +tampilDataCabang() : void Operasi ini digunakan untuk menampilkan data cabang yang ada di database. +cariDataCabang() : void Operasi ini digunakan untuk mencari data cabang berdasarkan lokasi. +validasiDataCabang() Operasi ini digunakan untuk mengecek data cabang yang diinputkan apakah sudah benar atau tidak. </pre>	

2.2.17.6 Specific Design Class KategoriBarangUI

kategori_barangUI	<<boundary>>
<pre> +kategori_barangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDisplayKategori() : Tabel Operasi ini digunakan untuk menampilkan data kategori +getKategoriById(id_kategori):void Operasi ini digunakan untuk mengambil data kategori yang dipergunakan untuk edit data kategori +entriDataKategori(kategori) : void Operasi ini digunakan untuk menyimpan data kategori dari form yang diinputkan oleh Admin. +editDataKategori(kategori) : void Operasi ini digunakan untuk mengedit data kategori yang ada di database. +hapusDataKategori(kategori) : void Operasi ini digunakan untuk menghapus semu data kategori yang ada di database. +tampilDataKategori(kategori) : void Operasi ini digunakan untuk menampilkan data kategori yang ada di database. +cariDataKategori() : void Operasi ini digunakan untuk mencari data kategori barang. +validasiDataKategori() Operasi ini digunakan untuk mengecek data kategori yang diinputkan apakah sudah benar atau tidak. </pre>	

2.2.17.7 Specific Design Class BarangUI

barangUI	<<boundary>>
<pre> + barangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDisplayBarang() : Tabel Operasi ini digunakan untuk menampilkan data barang +getBarangById(id_barang):void Operasi ini digunakan untuk mengambil data barang yang dipergunakan untuk edit data barang +entriDataBarang(barang) : void Operasi ini digunakan untuk menyimpan data barang dari form yang diinputkan oleh Admin. +editDataBarang(barang) : void Operasi ini digunakan untuk mengedit data barang yang ada di database. +hapusDataBarang(barang) : void Operasi ini digunakan untuk menghapus semu data barang yang ada di database. +tampilDataBarang(barang) : void Operasi ini digunakan untuk menampilkan data barang yang ada di database. +cariDataBarang() : void Operasi ini digunakan untuk mencari data barang. +validasiDataBarang() Operasi ini digunakan untuk mengecek data barang yang diinputkan apakah sudah benar atau tidak.</pre>	

2.2.17.8 Specific Design Class detilBarangUI

detilbarangUI	<<boundary>>
----------------------	---------------------------------

```
+ detailbarangUI()
```

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

```
+getDisplayDetailBarang() : Tabel
```

Operasi ini digunakan untuk menampilkan data detail barang

```
+getDetailBarangById(id_barang):void
```

Operasi ini digunakan untuk mengambil data detail barang yang dipergunakan untuk edit data barang

```
+entriDataDetailBarang(barang) : void
```

Operasi ini digunakan untuk menyimpan data detail barang dari form yang diinputkan oleh Admin.

```
+editDataDetailBarang(barang) : void
```

Operasi ini digunakan untuk mengedit data detail barang yang ada di database.

```
+hapusDataDetailBarang(barang) : void
```

Operasi ini digunakan untuk menghapus semua data detail barang yang ada di database.

```
+tampilDataDetailBarang(barang) : void
```

Operasi ini digunakan untuk menampilkan data detail barang yang ada di database.

```
+cariDataDetailBarang() : void
```

Operasi ini digunakan untuk mencari data detail barang.

```
+validasiDataDetailBarang()
```

Operasi ini digunakan untuk mengecek data detail barang yang diinputkan apakah sudah benar atau tidak.

**2.2.17.9 Specific Design Class
barangSupplierUI**

barangSupplierUI	<<boundary>>
<pre> + barangSupplierUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDisplayDetilBarangSupplier() : Tabel Operasi ini digunakan untuk menampilkan data barang supplier +getDetilBarangSupplierById(id_barang):void Operasi ini digunakan untuk mengambil data barang supplier yang dipergunakan untuk edit data barang supplier +entriDataBarangSupplier (barang) : void Operasi ini digunakan untuk menyimpan data barang supplier dari form yang diinputkan oleh Admin. +editDataBarangSupplier (barang) : void Operasi ini digunakan untuk mengedit data barang supplier yang ada di database. +hapusDataBarangSupplier (barang) : void Operasi ini digunakan untuk menghapus semu data barang supplier yang ada di database. +tampilDataBarangSupplier (barang) : void Operasi ini digunakan untuk menampilkan data barang supplier yang ada di database. +cariDataBarangSupplier () : void Operasi ini digunakan untuk mencari data barang supplier. +validasiDataBarangSupplier () Operasi ini digunakan untuk mengecek data barang </pre>	

supplier yang diinputkan apakah sudah benar atau tidak.

2.2.17.10 Specific Design Class pemesananSupplierUI

pemesananSupplierUI	<<boundary>>
<pre>+ pemesananSupplierUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDisplayPemesananSup() : Tabel Operasi ini digunakan untuk menampilkan data pemesanan barang supplier +getPemesananSupById(id_pemesanan):void Operasi ini digunakan untuk mendapatkan jumlah pemesanan barang berdasarkan tanggal yang dipergunakan untuk kode pemesanan barang supplier +entriDataPemesananSup(pemesanan) : void Operasi ini digunakan untuk menyimpan data pemesanan barang supplier dari form yang diinputkan oleh Gudang. +editDataPemesananSupplier(pemesanan) : void Operasi ini digunakan untuk mengedit data pemesanan barang supplier yang ada di database. +tampilDataPemesananSupplier() : void Operasi ini digunakan untuk menampilkan data pemesanan barang supplier yang ada di database. +cariDataPemesananSupplier() : void Operasi ini digunakan untuk mencari data pemesanan barang supplier. +batalPemesananSupplier(id_pemesanan) : void Operasi ini digunakan untuk melakukan pembatalan</pre>	

```

data pemesanan barang supplier.
+ubahStatusPemesananSupplier(id_pemesanan_ : void
Operasi ini digunakan untuk mengubah status
pemesanan menjadi "Ter kirim ke supplier"

```

**2.2.17.11 Specific Design Class
detail_pemesananSupplierUI**

detail_pemesananUI	<<boundary>>
<pre> +detil_pemesananSupplierUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDetilPemesananSupplier() : Tabel Operasi ini digunakan untuk menampilkan data detil pemesanan barang supplier. +insertDetailPemesananSupplier(pemesanan): void Operasi ini digunakan untuk menambah data detil pemesanan barang supplier. +ubahDetailPemesananSupplier(pemesanan) : void Operasi ini digunakan untuk mengubah data detil pemesanan barang supplier. +hapusDetailPemesananSupplier(pemesanan) : void Operasi ini digunakan untuk menghapus data detil pemesanan barang supplier. </pre>	

**2.2.17.10 Specific Design Class
pemesananCabangUI**

pemesananCabangUI	<<boundary>>
<pre> + pemesananCabangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDisplayPemesananCab() : Tabel Operasi ini digunakan untuk menampilkan data pemesanan barang cabang +getPemesananCabById(id_pemesanan):void Operasi ini digunakan untuk mendapatkan jumlah pemesanan barang berdasarkan tanggal yang dipergunakan untuk kode pemesanan barang cabang +entriDataPemesananCab(pemesanan) : void Operasi ini digunakan untuk menyimpan data pemesanan barang cabang dari form yang diinputkan oleh Gudang. +editDataPemesananCabang(pemesanan) : void Operasi ini digunakan untuk mengedit data pemesanan barang cabang yang ada di database. +tampilDataPemesananCabang() : void Operasi ini digunakan untuk menampilkan data pemesanan barang cabang yang ada di database. +cariDataPemesananCabang() : void Operasi ini digunakan untuk mencari data pemesanan barang cabang. +batalPemesananCabang(id_pemesanan) : void Operasi ini digunakan untuk melakukan pembatalan data pemesanan barang cabang. +ubahStatusPemesananCabang(id_pemesanan_ : void </pre>	

Operasi ini digunakan untuk mengubah status pemesanan menjadi "Ter kirim ke cabang"

2.2.17.11 Specific Design Class detail_pemesananCabangUI

detail_pemesananUI	<<boundary>>
<pre> +detil_pemesananCabangUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDetilPemesananCabang() : Tabel Operasi ini digunakan untuk menampilkan data detil pemesanan barang cabang. +insertDetailPemesananCabang(pemesanan): void Operasi ini digunakan untuk menambah data detil pemesanan barang cabang. +ubahDetailPemesananCabang(pemesanan) : void Operasi ini digunakan untuk mengubah data detil pemesanan barang cabang. +hapusDetailPemesananCabang(pemesanan) : void Operasi ini digunakan untuk menghapus data detil pemesanan barang cabang. </pre>	

2.2.17.12 Specific Design Class PenerimaanSupplierUI

penerimaanSupplierUI	<<boundary>>
<pre> +penerimaanSupplierUI() Default konstruktor dimana semua atribut pada class ini di inisialisasi. +getDisplayPenerimaanSupplier():Tabel </pre>	

Operasi ini digunakan untuk menampilkan data penerimaan barang supplier.

```
+entriPenerimaanSupplier(penerimaan) : void
```

Operasi ini digunakan untuk menyimpan data penerimaan barang supplier.

```
+ubahStatusPenerimaanSupplier(id_pemesanan_ : void
```

Operasi ini digunakan untuk mengubah status dari yang "Barang Diterima Belum Lengkap" menjadi "Barang Diterima Lengkap".

```
+updateStokdanHargaBarang(id_barang) : void
```

Operasi ini digunakan untuk memperbaharui stok dan harga dari barang supplier.

2.2.17.13 Specific Design Class PenerimaanCabangUI

penerimaanCabangUI	<<boundary>>
<pre>+penerimaanCabangUI()</pre> <p>Default konstruktor dimana semua atribut pada class ini di inisialisasi.</p> <pre>+getDisplayPenerimaanCabang():Tabel</pre> <p>Operasi ini digunakan untuk menampilkan data penerimaan barang cabang.</p> <pre>+entriPenerimaanCabang(penerimaan) : void</pre> <p>Operasi ini digunakan untuk menyimpan data penerimaan barang cabang.</p> <pre>+ubahStatusPenerimaanCabang(id_pemesanan_ : void</pre> <p>Operasi ini digunakan untuk mengubah status dari yang "Barang Diterima Belum Lengkap" menjadi "Barang Diterima Lengkap".</p> <pre>+updateStokdanHargaBarang(id_barang) : void</pre>	

Operasi ini digunakan untuk memperbaharui stok dan harga dari barang cabang.

2.2.17.14 Specific Design Class TransPenjualanUI

trans_penjualanUI	<<boundary>>
<pre>+trans_penjualanUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getTansPenjualan() : Tabel Operasi ini digunakan untuk menampilkan transaksi penjualan. +getTransByTgl(): void Operasi ini digunakan untuk mencari adta transaksi berdasarkan tanggal. +getJumlahTransByTgl : void Operasi ini digunakan untuk mendapatkan jumlah transaksi penjualan berdasarkan tanggal digunakan untuk kode transaksi penjualan. +insertTrans_penjualan (trans_penjualan) : void Operasi ini digunakan untuk menambah data transaksi penjualan yang baru. +batalTrans_Penjualan(trans_penjualan) : void Operasi ini digunakan untuk melakukan pembatalan terhadap transaksi penjualan. +cetakNota(id_transPenjualan) : void Operasi ini digunakan untuk mencetak nota pembayaran setelah proses transaksi selesai.</pre>	

**2.2.17.15 Specific Design Class
detailtrans_penjualanUI**

DetailTrans_PenjualanUI	<<boundary>>
<pre> +detailTrans_penjualanUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDetilTransPenjualan() : Tabel Operasi ini digunakan untuk menampilkan detil transaksi penjualan. +insertDetilTransPenjualan(detilTrans_penjualan):voi d Operasi ini digunakan untuk menambah detil transaksi penjualan yang baru. +ubahDetilTrans_Penjualan(detilTrans_penjualan):void Operas digunakan untuk mengubah detil transaksi penjualan. +updateStokBarang(barang): void Operasi ini digunakan untuk memperbaharui stok barang dengan melakukan pengurangan stok setelah transaksi selesai. +hapusDetilTrans_penjualann(detilTrans_penjualan):vo id Operasi ini digunakan untuk menghapus detil transaksi penjualan. </pre>	

**2.2.17.16 Specific Design Class
lapPenjualanUI()**

lapPenjualanUI	<<control>>
<pre> +lapPenjualanUI() </pre>	

Default konstruktor yang menginisialisasi semua atribut pada class ini

+lap_Penjualan () : void

Operasi ini digunakan untuk menampilkan laporan banyaknya transaksi penjualan berdasarkan periode bulan.

2.2.17.17 Specific Design Class lapPendapatanUI()

lapPendapatanUI	<<control>>
<p>+lapPendapatanUI()</p> <p>Default konstruktor yang menginisialisasi semua atribut pada class ini</p> <p>+lap_Penjualan () : void</p> <p>Operasi ini digunakan untuk menampilkan laporan jumlah pendapatan berdasarkan periode bulan.</p>	

2.2.17.18 Specific Design Class historiStokUI()

Histori_StokUI	<<control>>
<p>+histori_stokUI()</p> <p>Default konstruktor yang menginisialisasi semua atribut pada class ini</p> <p>+tampilHistoriStok() : void</p> <p>Operasi ini digunakan untuk menampilkan histori stok barang yang pernah dipesan.</p>	

2.2.17.19 Specific Design Class historiHargaUI()

Histori_hargaUI	<<control>>

+histori_hargaUI()

Default konstruktor yang menginisialisasi semua atribut pada class ini.

+tampilHistori_harga() : void

Operasi yang digunakan untuk menampilkan data histori harga barang.

2.2.17.20 Specific Design Class loginCtrl()

loginCtrl	<<control>>
+loginCtrl() Default konstruktor yang menginisialisasi semua atribut pada class ini. +loginUser() Operasi ini digunakan untuk melakukan pengecekan apakah username dan password yang dimasukkan valid atau tidak sesuai dengan role masing-masing.	

2.2.17.21 Specific Design Class userCtrl()

userCtrl	<<control>>
+userCtrl () Default konstruktor yang menginisialisasi semua atribut pada class ini +getUser() : Tabel Operasi ini digunakan untuk menampilkan data user. +getUserById() : void Operasi ini digunakan untuk mengambil mengambil data user yang digunakan untuk mengambil data user yang digunakan untuk mengubah data user. +getPassword() : void	

operasi ini digunakan untuk mengambil password dari pengguna.

```
+ubahPassword(pass_lama,pass_baru,confirm_passBaru):
void
```

Operasi ini digunakan untuk mengubah password dari pengguna.

```
+isUnik()
```

Operasi ini digunakan untuk melakukan pengecekan apakah username sudah terdaftar atau belum.

```
+insertUser(user):void
```

Operasi ini digunakan untuk menambah data user baru.

```
+editUser(user):void
```

Operasi ini digunakan untuk mengubah data user

```
+tampilUser(user) : void
```

Operasi ini digunakan untuk menampilkan semua data user yang atribut is_deleted = False dari database.

2.2.17.22 Specific Design Class cabangCtrl()

cabangCtrl	<<control>>
<pre>+cabangCtrl()</pre> <p>Default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini</p> <pre>+getCabangById(id_cabang) : void</pre> <p>Operasi ini digunakan untuk mengambil data cabang berdasarkan id yang digunakan untuk ubah data cabang</p> <pre>+insertCabang(cabang) : void</pre> <p>Operasi ini digunakan untuk menambah data cabang yang baru.</p> <pre>+ubahCabang(cabang) : void</pre>	

Operasi ini digunakan untuk mengubah data cabang yang ada di database.

+hapusCabang(cabang) : void

Operasi ini digunakan untuk menghapus semua data cabang.

+cariCabang(cabang) : void

Operasi ini digunakan untuk mencari data cabang yang ada di database.

+tampilCabang(cabang) : void

Operasi ini digunakan untuk menampilkan semua data cabang yang atribut is_deleted = False dari database.

2.2.17.23 Specific Design Class

kategori_barangCtrl()

kategori_barangCtrl	<<control>>
<p>+kategori_barangCtrl() Default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini</p> <p>+getkategori_barangById(id_kategori_barang) : void Operasi ini digunakan untuk mengambil data kategori_barang berdasarkan id yang digunakan untuk ubah data kategori_barang</p> <p>+insertkategori_barang (kategori_barang) : void Operasi ini digunakan untuk menambah data kategori_barang yang baru.</p> <p>+ubahkategori_barang (kategori_barang) : void Operasi ini digunakan untuk mengubah data kategori_barang yang ada di database.</p>	

```

+hhapuskategori_barang (kategori_barang) : void
Operasi ini digunakan untuk menghapus semua data
kategori_barang.

+carikategori_barang (kategori_barang) : void
Operasi ini digunakan untuk mencari data
kategori_barang yang ada di database.

+tampilkategori_barang (kategori_barang) : void
Operasi ini digunakan untuk menampilkan semua data
kategori_barang yang atribut is_deleted = False dari
database.

```

2.2.17.24 Specific Design Class barangCtrl()

barangCtrl	<<control>>
<pre> +barangCtrl() Default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini +getbarangById(id_barang) : void Operasi ini digunakan untuk mengambil data barang berdasarkan id yang digunakan untuk ubah data barang +insertbarang (barang) : void Operasi ini digunakan untuk menambah data barang yang baru. +ubahbarang (barang) : void Operasi ini digunakan untuk mengubah data barang yang ada di database. +hapusbarang (barang) : void Operasi ini digunakan untuk menghapus semua data barang. +caribarang (barang) : void </pre>	

Operasi ini digunakan untuk mencari data barang yang ada di database.

```
+tampilbarang (barang) : void
```

Operasi ini digunakan untuk menampilkan semua data barang yang atribut is_deleted = False dari database.

2.2.17.25 Specific Design Class detailbarangCtrl()

detailbarangCtrl	<<control>>
<pre>+barangDetailCtrl()</pre> <p>Default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini</p> <pre>+getDetailbarangById(id_barang) : void</pre> <p>Operasi ini digunakan untuk mengambil data detail barang berdasarkan id yang digunakan untuk ubah data detail barang</p> <pre>+insertDetailbarang (barang) : void</pre> <p>Operasi ini digunakan untuk menambah data detail barang yang baru.</p> <pre>+ubahDetailbarang (barang) : void</pre> <p>Operasi ini digunakan untuk mengubah data detail barang yang ada di database.</p> <pre>+hapusDetailbarang (barang) : void</pre> <p>Operasi ini digunakan untuk menghapus semua data detail barang.</p> <pre>+cariDetailbarang (barang) : void</pre> <p>Operasi ini digunakan untuk mencari data detail barang yang ada di database.</p> <pre>+tampilDetailbarang (barang) : void</pre>	

Operasi ini digunakan untuk menampilkan semua data detail barang yang atribut `is_deleted = False` dari database.

2.2.17.26 Specific Design Class `barangSupplierCtrl()`

<code>barangSupplierCtrl</code>	<code><<control>></code>
<pre>+barangSupplierCtrl() Default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini +getbarangSupplierById(id_barang) : void Operasi ini digunakan untuk mengambil data barang supplier berdasarkan id yang digunakan untuk ubah data barang supplier +insertbarangSupplier (barang) : void Operasi ini digunakan untuk menambah data barang supplier yang baru. +ubahbarangSupplier (barang) : void Operasi ini digunakan untuk mengubah data barang supplier yang ada di database. +hapusbarangSupplier (barang) : void Operasi ini digunakan untuk menghapus semua data barang supplier. +caribarangSupplier (barang) : void Operasi ini digunakan untuk mencari data barang supplier yang ada di database. +tampilbarangSupplier (barang) : void Operasi ini digunakan untuk menampilkan semua data barang supplier yang atribut is_deleted = False dari database.</pre>	

**2.2.17.27 Specific Design Class
pemesananSupplierCtrl**

pemesananSupplierCtrl	<<control>>
<pre> + pemesananSupplierCtrl() Default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini + getPesananSupplier() : Tabel Operasi ini digunakan untuk menampilkan data pemesanan barang supplier. + getPesananSupplierByTgl() : void Operasi ini digunakan untuk mencari data pemesanan barang supplier berdasarkan tanggal. + getDetilPemesananSupplier() : Tabel Operasi ini digunakan untuk menampilkan data detil pemesanan barang supplier. + getJumlahPemesananSupByTgl() : void Operasi ini digunakan untuk mendapatkan jumlah pemesanan berdasarkan tanggal yang akan digunakan untuk kode pemesanan barang supplier. + insertPemesananSupplier (pemesanan_barang) : void Operasi ini digunakan untuk menambah data pemesanan barang supplier yang baru. + ubahPemesanaSupplier (pemesanan_barang) : void Operasi ini digunakan untuk mengubah data pemesanan barang supplier yang ada di database. + insertDetilPemesananSupplier (pemesanan_barang) : void Operasi ini digunakan untuk menambah data detil pemesanan barang supplier yang baru. + ubahDetilPemesananSupplier (pemesanan_barang) :</pre>	

```

void
Operasi ini digunakan untuk mengubah data detil
pemesanan barang supplier yang ada di database.
+hapusDetilPemesananSupplier
(id_detailpemesanan):void
Operasi ini digunakan untuk mengubah data detil
pemesanan barang supplier yang ada di database.
+batalPemesananSupplier (id_pemesanan) : void
Operasi ini digunakan untuk melakukan pembatalan
pemesanan barang supplier.
+updateStatusPemesananSupplier (id_pemesanan) : void
Operasi ini digunakan untuk mengubah status
pemesanan barang supplier.
+gettampilPenerimaanSupplier (barang) : Tabel
Operasi ini digunakan untuk menampilkan semua data
barang supplier yang atribut is_deleted = False dari
database.
+updateSokdanHargaBarang (id_barang) : void
Operasi ini digunakan untuk mengubah stok dan harga
barang.

```

2.2.17.28 Specific Design Class pemesananCabangCtrl

pemesananCabangCtrl	<<control>>
<pre> + pemesananCabangCtrl() Default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini + getPemesananCabang() : Tabel Operasi ini digunakan untuk menampilkan data pemesanan barang cabang. </pre>	

```

+getPemesananCabangByTgl() : void
Operasi ini digunakan untuk mencari data pemesanan
barang cabang berdasarkan tanggal.
+getDetilPemesananCabang () : Tabel
Operasi ini digunakan untuk menampilkan data detil
pemesanan barang cabang.
+getJumlahPemesananCabByTgl() : void
Operasi ini digunakan untuk mendapatkan jumlah
pemesanan berdasarkan tanggal yang akan digunakan
untuk kode pemesanan barang cabang.
+insertPemesananCabang (pemesanan_barang) : void
Operasi ini digunakan untuk menambah data pemesanan
barang cabang yang baru.
+ubahPemesanaCabang (pemesanan_barang) : void
Operasi ini digunakan untuk mengubah data pemesanan
barang cabang yang ada di database.
+insertDetilPemesananCabang (pemesanan_barang): void
Operasi ini digunakan untuk menambah data detil
pemesanan barang cabang yang baru.
+ubahDetilPemesananCabang (pemesanan_barang) : void
Operasi ini digunakan untuk mengubah data detil
pemesanan barang cabang yang ada di database.
+hapusDetilPemesananCabang (id_detailpemesanan):void
Operasi ini digunakan untuk mengubah data detil
pemesanan barang cabang yang ada di database.
+batalPemesananCabang (id_pemesanan) : void
Operasi ini digunakan untuk melakukan pembatalan
pemesanan barang cabang.
+updateStatusPemesananCabang (id_pemesanan) : void
Operasi ini digunakan untuk mengubah status

```

Program Studi Teknik Informatika	DPPL-SISIP	80/151
Dokumen ini adalah 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		

pemesanan barang cabang.

+gettampilPenerimaanCabang (barang) : Tabel

Operasi ini digunakan untuk menampilkan semua data barang cabang yang atribut is_deleted = False dari database.

+updateSokdanHargaBarang (id_barang) : void

Operasi ini digunakan untuk mengubah stok dan harga barang.

2.2.17.29 Specific Design Class trans_penjualanCtrl

Trans_penjualanCtrl	<<control>>
<p>+trans_penjualanCtrl() Default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini</p> <p>+getTrans_penjualan() : Tabel Operasi ini digunakan untuk menampilkan data transaksi penjualan.</p> <p>+getTrans_PenjualanByTgl() : void Operasi ini digunakan untuk mencari data transaksi penjualan berdasarkan tanggal.</p> <p>+getDetilTrans_penjualan() : Tabel Operasi ini digunakan untuk menampilkan data detil transaksi penjualan.</p> <p>+getJumlahTrans_PenjualanByTgl() : void Operasi ini digunakan untuk mendapatkan jumlah transaksi penjualan berdasarkan tanggal yang akan digunakan untuk kode transaksi penjualan.</p> <p>+insertTrans_penjualan (trans_penjualan) : void Operasi ini digunakan untuk menambah data transaksi</p>	

penjualan yang baru.

+ubahTrans_penjualan (trans_penjualan) : void
 Operasi ini digunakan untuk mengubah data transaksi penjualan.

+insertDetilTrans_Penjualan (trans_penjualan): void
 Operasi ini digunakan untuk menambah data detil transaksi penjualan yang baru.

+ubahDetilTrans_penjualan (trans_penjualan) : void
 Operasi ini digunakan untuk mengubah data detil transaksi penjualan yang ada di database.

+hapusDetilTrans_Penjualan(id_detailTrans_penjualan):void
 Operasi ini digunakan untuk mengubah data detil transaksi penjualan yang ada di database.

+batalTrans_penjualan (id_trans_penjualan) : void
 Operasi ini digunakan untuk melakukan pembatalan transaksi penjualan.

+updateStokBarang(barang) : void
 Operasi ini digunakan untuk melakukan pengurangan terhadap stok barang setelah transaksi penjualan.

+updateStatusPemesanan (id_pemesanan) : void
 Operasi ini digunakan untuk mengubah status pemesanan barang.

+batalDetilTrans_Penjualan (id_detilTrans_Penjualan):
 Tabel
 Operasi ini digunakan untuk melakukan pembatalan terhadap data detil transaksi penjualan.

2.2.17.30 Spesific Design Class laporanCtrl

laporanCtrl	<<control>>

```

+lap_Penjualan () : void
Operasi ini digunakan untuk menampilkan laporan
banyaknya transaksi penjualan yang terjadi
berdasarkan periode bulan dalam setahun.
+lap_Pendapatan () : void
Operasi ini digunakan untuk menampilkan laporan
jumlah pendapatan tiap cabang berdasarkan periode
bulan dalam setahun.
+historiStokBarang() : void
Operasi ini digunakan untuk menampilkan laporan
histori stok barang jika ada perubahan stok tiap
cabang.
+ historiHargaBarang () : void
Operasi ini digunakan untuk menampilkan laporan
histori harga barang di tiap cabang jika terjadi
perubahan harga.

```

2.2.17.31 Specific Design Class userModel

userModel	<<entity>>
<p>-ID_USER: integer, digunakan untuk menyimpan data numerik identitas user.</p> <p>-USERNAME: string, digunakan untuk menyimpan data username user.</p> <p>-PASSWORD: string, digunakan untuk menyimpan data password user.</p> <p>-NAMA_USER: string, digunakan untuk menyimpan data nama lengkap user.</p> <p>-JENIS_KELAMIN: string, digunakan untuk menyimpan data gender user.</p> <p>-ALAMAT_USER: string, digunakan untuk menyimpan data alamat user.</p>	

-NOTELEP_USER: string, digunakan untuk menyimpan data telepon user.

-EMAIL_USER: string, digunakan untuk menyimpan data email user.

-FOTO_USER: string, digunakan untuk menyimpan data foto user.

-IS_DELETED: bit, digunakan untuk menyimpan kode status hapus user.

+user()
Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini

+getDataUser() : Tabel
Operasi ini digunakan untuk mengambil semua data user dari database

+getUserById(id_user) : void
Operasi ini digunakan untuk mengambil data user berdasarkan id yang dipergunakan untuk ubah data user.

+getPassword() : void
Operasi ini digunakan untuk mengambil password dari user.

+updatePassword(pass_lama,pass_baru,confirm_pass_baru):void
Operasi ini digunakan untuk mengubah password dari user.

+isUnik()
Operasi ini digunakan untuk melakukan pengecekan apakah username sudah terdaftar di database atau belum.

+insDataUser(user): void
Operasi ini digunakan untuk mengentrikan data user ke dalam database.

+ubahDataUser(user) : void
Operasi ini digunakan untuk mengedit data user yang ada di database


```
+loginUser(username,password) : bool
```

Operasi ini digunakan untuk melakukan pengecekan apakah username dan password sudah valid atau belum.

2.2.17.32 Specific Design Class roleModel

roleModel	<<entity>>
<pre>-ID_ROLE: integer, digunakan untuk menyimpan data numerik identitas role.</pre>	
<pre>-NAMA_ROLE: string, digunakan untuk menyimpan data nama role.</pre>	
<pre>+role() Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini</pre>	
<pre>+getRole() : Tabel Operasi ini digunakan untuk mengambil semua data user dari database</pre>	

2.2.17.33 Specific Design Class supplierModel

supplierModel	<<entity>>
<pre>-ID_SUPPLIER: integer, digunakan untuk menyimpan data numerik identitas supplier.</pre>	
<pre>-NAMA_SUPPLIER: string, digunakan untuk menyimpan data nama supplier.</pre>	
<pre>-ALAMAT_SUPPLIER: string, digunakan untuk menyimpan data alamat supplier.</pre>	
<pre>-NOTELP: string, digunakan untuk menyimpan data no telp supplier.</pre>	
<pre>-IS_DELETED: bit, digunakan untuk menyimpan kode status hapus supplier.</pre>	
<pre>+ supplier () Default konstruktor untuk menginisialisasikan</pre>	

seluruh variabel di kelas ini

+getDataSupplier() : Tabel
Operasi ini digunakan untuk mengambil semua data supplier dari database

+getSupplierById(id_supplier) : void
Operasi ini digunakan untuk mengambil data supplier berdasarkan id yang dipergunakan untuk ubah data supplier.

+insDataSupplier(supplier): void
Operasi ini digunakan untuk mengentrikan data supplier ke dalam database.

+ubahDataSupplier (supplier) : void
Operasi ini digunakan untuk mengedit data supplier yang ada di database.

+hapusSupplier (supplier) : void
Operasi ini digunakan untuk menghapus data supplier.

2.2.17.34 Specific Design Class cabangModel

cabangModel	<<entity>>
<p>-ID_CABANG: integer, digunakan untuk menyimpan data numerik identitas pegawai.</p> <p>-NAMA_CABANG: string, digunakan untuk menyimpan data nama cabang.</p> <p>-ALAMAT_CABANG: string, digunakan untuk menyimpan data alamat cabang.</p> <p>-NOTELP: string, digunakan untuk menyimpan data no telp cabang.</p> <p>-IS_PUSAT: bit, digunakan untuk menyimpan kode status bahwa cabang tersebut merupakan pusat atau bukan.</p> <p>-IS_DELETED: bit, digunakan untuk menyimpan kode</p>	

status hapus cabang.

+cabang()

Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini

+getDataCabang() : Tabel

Operasi ini digunakan untuk mengambil semua data cabang dari database

+getCabangById(id_cabang) : void

Operasi ini digunakan untuk mengambil data cabang berdasarkan id yang dipergunakan untuk ubah data cabang.

+insDataCabang (cabang): void

Operasi ini digunakan untuk mengentrikan data cabang ke dalam database.

+ubahDataCabang (cabang) : void

Operasi ini digunakan untuk mengedit data cabang yang ada di database.

+hapusCabang (cabang) : void

Operasi ini digunakan untuk menghapus data cabang.

2.2.17.35 Specific Design Class

kategori_barangModel

Kategori_barangModel	<<Entity>>
-ID_KATEGORI_BARANG:integer, digunakan untuk menyimpan data karakter identitas dari layanan.	
-NAMA_KATEGORI_BARANG:string, digunakan untuk menyimpan data nama layanan.	
-IS_DELETED: bit, digunakan untuk menyimpan kode status hapus kategori barang.	

+kategoriBarang()

Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini

+getDataKategoriBarang() : Tabel

Operasi ini digunakan untuk mengambil semua data kategori barang dari database

+getKategoriById(id_kategori) : void

Operasi ini digunakan untuk mengambil data kategori barang berdasarkan id yang dipergunakan untuk ubah data kategori barang.

+insDataKategori (kategori): void

Operasi ini digunakan untuk mendaftarkan data kategori barang ke dalam database.

+ubahDataKategori (kategori) : void

Operasi ini digunakan untuk mengedit data kategori barang yang ada di database.

+hapusKategori (kategori) : void

Operasi ini digunakan untuk menghapus data kategori.

2.2.17.36 Spesific Design Class barangModel

barangModel	<<Entity>>
-ID_BARANG: integer, digunakan untuk menyimpan data numerik identitas barang.	
-NAMA_BARANG: string, digunakan untuk menyimpan data nama barang.	
-SATUAN: string, digunakan untuk menyimpan data satuan barang (kilogram, liter, mililiter, dsb.).	
-FOTO_BARANG: string, digunakan untuk menyimpan data foto barang.	
-IS_DELETED: bit, digunakan untuk menyimpan kode status hapus semu barang.	

```

+Barang()
Default konstruktor untuk menginisialisasikan
seluruh variabel di kelas ini
+getDataBarang() : Tabel
Operasi ini digunakan untuk mengambil semua data
barang dari database
+getBarangById(id_barang) : void
Operasi ini digunakan untuk mengambil data barang
berdasarkan id yang dipergunakan untuk ubah data
barang.
+insDataBarang (barang): void
Operasi ini digunakan untuk mengentrikan data barang
ke dalam database.
+ubahDataBarang (barang) : void
Operasi ini digunakan untuk mengedit data barang
yang ada di database.
+hapusBarang (barang) : void
Operasi ini digunakan untuk menghapus data barang.

```

2.2.17.1 Spesific Design Class detilBarangModel

detilbarangModel	<<Entity>>
<p>-ID_HARGA_SATUAN_BARANG: integer, digunakan untuk menyimpan data numerik identitas barang.</p> <p>-KODE_BARANG: string, digunakan untuk menyimpan data kode unik identitas barang untuk barcode.</p> <p>-KEMASAN: string, digunakan untuk menyimpan data kemasan barang.</p> <p>-HARGA_PER_KEMASAN: decimal, digunakan untuk menyimpan data harga barang.</p> <p>-STOK: integer, digunakan untuk menyimpan data stok</p>	

<p>barang.</p> <p>-IS_DELETED: bit, digunakan untuk menyimpan kode status hapus semu barang.</p>
<p>+detailBarang() Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini</p> <p>+getDatadetailBarang() : Tabel Operasi ini digunakan untuk mengambil semua data detail barang dari database</p> <p>+getdetailBarangById(id_barang) : void Operasi ini digunakan untuk mengambil data detailbarang berdasarkan id yang dipergunakan untuk ubah data detail barang.</p> <p>+insDatadetailBarang (barang): void Operasi ini digunakan untuk mengentrikan data detail barang ke dalam database.</p> <p>+ubahDatadetailBarang (barang) : void Operasi ini digunakan untuk mengedit data detail barang yang ada di database.</p> <p>+hapusdetailBarang (barang) : void Operasi ini digunakan untuk menghapus data detailbarang.</p>

2.2.17.1 Spesific Design Class barangSupplierModel

barangSupplierModel	<<Entity>>
<p>-ID_BARANG: integer, digunakan untuk menyimpan data numerik identitas barang.</p> <p>-KODE_BARANG: string, digunakan untuk menyimpan data kode unik identitas barang.</p> <p>-KEMASAN: integer, digunakan untuk menyimpan data</p>	

kemasan barang.

-HARGA_PER_KEMASAN: decimal, digunakan untuk menyimpan data harga barang.

-IS_DELETED: bit, digunakan untuk menyimpan kode status hapus semu barang supplier.

+BarangSupplier()

Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini

+getDataBarangSupplier () : Tabel

Operasi ini digunakan untuk mengambil semua data barang supplier dari database

+getBarangSupplierById(id_barang) : void

Operasi ini digunakan untuk mengambil data barang berdasarkan id yang dipergunakan untuk ubah data barang supplier.

+insDataBarangSupplier (barang): void

Operasi ini digunakan untuk mengentrikan data barang supplier ke dalam database.

+ubahDataBarangSupplier (barang) : void

Operasi ini digunakan untuk mengedit data barang supplier yang ada di database.

+hapusBarangSupplier (barang) : void

Operasi ini digunakan untuk menghapus data barang supplier.

2.2.17.2 Spesific Design Class histori_hargaModel

Histori_hargaModel

<<Entity>>

-ID_HISTORI_HARGA: integer, digunakan untuk menyimpan data numerik identitas histori harga barang.

<p>-HARGA_LAMA: Decimal, digunakan untuk menyimpan data harga lama.</p> <p>-HARGA_BARU: Decimal, digunakan untuk menyimpan data harga baru.</p> <p>-TGL_HISTORI_HARGA: datetime, digunakan untuk menyimpan data tanggal histori harga.</p>
<p>+histori_harga()</p> <p>Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini.</p> <p>+tampilHistori_harga() : void</p> <p>Operasi ini digunakan untuk menampilkan data histori harga.</p>

2.2.17.3 Specific Design Class histori_stokModel

Histori_stokModel	<<Entity>>
<p>-ID_HISTORI_STOK: integer, digunakan untuk menyimpan data numerik identitas histori stok barang.</p> <p>-STOK_LAMA: Decimal, digunakan untuk menyimpan data stok lama.</p> <p>-STOK_BARU: Decimal, digunakan untuk menyimpan data stok baru.</p> <p>-TGL_HISTORI_STOK: datetime, digunakan untuk menyimpan data tanggal histori stok.</p>	
<p>+histori_stok()</p> <p>Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini.</p> <p>+tampilHistori_stok() : void</p> <p>Operasi ini digunakan untuk menampilkan data histori stok.</p>	

**2.2.17.4 Specific Design Class
hargaSatuan barangModel**

hargaSatuan_barangModel	<<Entity>>
<p>-ID_HARGA_SATUAN_BARANG: integer, digunakan untuk menyimpan data numerik identitas harga satuan barang.</p> <p>-KODE_BARANG: string, digunakan untuk menyimpan data kode unik harga satuan barang.</p> <p>-KEMASAN: integer, digunakan untuk menyimpan data besar kemasan satuan barang.</p> <p>-HARGA_PER_KEMASAN: decimal, digunakan untuk menyimpan data harga satuan barang.</p> <p>-STOK: integer, digunakan untuk menyimpan data stok satuan barang.</p> <p>-IS_DELETED: bit, digunakan untuk menyimpan kode status hapus semu satuan barang.</p>	
<p>+harga_satuan_barang() Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini.</p> <p>+getHargaSatuan() : Tabel Operasi ini digunakan untuk menampilkan data harga satuan barang</p>	

**2.2.17.5 Specific Design Class
barangSupplierModel**

barangSupplierModel	<<Entity>>
<p>-ID_BARANG_SUPPLIER: integer, digunakan untuk menyimpan data numerik identitas harga satuan barang supplier.</p> <p>-KODE_BARANG: string, digunakan untuk menyimpan data kode unik harga satuan barang supplier.</p>	

<p>-KEMASAN: integer, digunakan untuk menyimpan data besar kemasan satuan barang supplier.</p> <p>-HARGA_PER_KEMASAN: decimal, digunakan untuk menyimpan data harga satuan barang supplier.</p> <p>-IS_DELETED: bit, digunakan untuk menyimpan kode status hapus semu satuan barang supplier.</p>
<p>+harga_satuan_barang()</p> <p>Default konstruktor untuk menginisialisasikan seluruh variabel di kelas ini.</p> <p>+getHargaSatuan() : Tabel</p> <p>Operasi ini digunakan untuk menampilkan data harga satuan barang</p>

2.2.17.6 Spesific Design Class pemesananSupplierModel

Pemesanan_supplierModel	<<Entity>>
<p>-ID_PESANSUPPLIER: integer, digunakan untuk menyimpan data numerik identitas pemesanan supplier.</p> <p>-KODE_PESANSUPPLIER: integer, digunakan untuk menyimpan data kode unik pemesanan supplier.</p> <p>-TGL_PESANSUPPLIER: date, digunakan untuk menyimpan data tanggal pemesanan supplier.</p> <p>-TOTAL_HARGA: decimal, digunakan untuk menyimpan data total harga.</p> <p>-STATUS_PESAN: string, digunakan untuk menyimpan data status pesan.</p> <p>-TGL_PENERIMAAN_SUP: date, digunakan untuk menyimpan data tanggal penerimaan supplier.</p>	
<p>+pemesananSup()</p> <p>Default konstruktor, digunakan untuk</p>	

menginisialisasikan semua atribut yang ada dikelas ini.

+getDataPemesananSup() : Tabel

Operasi ini digunakan untuk mengambil semua data dari tabel pemesanan.

+getPemesananSupByTgl(): void

Operasi ini digunakan untuk mencari data pemesanan berdasarkan tanggal.

+getJumlahPemesananSupByTgl(id_pemesanan): void

Operasi ini digunakan untuk mendapatkan jumlah pemesanan berdasarkan tanggal yang digunakan untuk kode pemesanan barang.

+insDataPemesananSup(pemesanan) : void

Operasi ini digunakan untuk memasukkan data pemesanan ke dalam database.

+ubahDataPemesananSup (pemesanan) : void

Operasi ini digunakan untuk mengedit data pemesanan yang ada di database

+batalDataPemesananSup(id_pemesanan) : void

Operasi ini digunakan untuk membatalkan data pemesanan.

+insertBarangSup(barang) : void

Operasi ini digunakan untuk menambah data barang baru yang diterima.

+updateStatusPemesananSup(id_pemesanan) : void

Operasi ini digunakan untuk mengubah status pemesanan barang.

**2.2.17.7 Specific Design Class
detilPemesananSupplierModel**

DetilPemesananSupplierModel	<<Entity>>
<p>-JUMLAH_PESANSUPPLIER: integer, digunakan untuk menyimpan data jumlah pemesanan supplier.</p> <p>-HARGA_SUPPLIER: decimal, digunakan untuk menyimpan data harga supplier.</p> <p>-SUBTOTAL: decimal, digunakan untuk menyimpan data sub total harga.</p> <p>-JUMLAH_BELUMDITERIMA: integer, digunakan untuk menyimpan data jumlah barang yang belum diterima.</p> <p>-TGL_PENERIMAAN_DETILSUP: date, digunakan untuk menyimpan data tanggal detail penerimaan supplier.</p>	
<p>+detilPemesananSup() Default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada dikelas ini.</p> <p>+getDetilPemesananSup () : Tabel Operasi ini digunakan untuk menampilkan data detail pemesanan.</p> <p>+insDataDetailPemesananSup (pemesanan) : void Operasi ini digunakan untuk memasukkan data pemesanan ke dalam database.</p> <p>+ubahDataDetilPemesananSup (pemesanan) : void Operasi ini digunakan untuk mengedit data detail pemesanan yang ada di database</p> <p>+hapusDataDetilPemesananSup (id_detil_pemesanan) : void Operasi ini digunakan untuk menghapus data detail pemesanan yang ada di database</p>	

**2.2.17.8 Specific Design Class
pemesananCabangModel**

Pemesanan_cabangModel	<<Entity>>
<p>-ID_PESANCABANG: integer, digunakan untuk menyimpan data numerik identitas pemesanan cabang.</p> <p>-KODE_PESANCABANG: integer, digunakan untuk menyimpan data kode unik pemesanan cabang.</p> <p>-TGL_PESANCABANG: date, digunakan untuk menyimpan data tanggal pemesanan cabang.</p> <p>-STATUS_PESAN: string, digunakan untuk menyimpan data status pesan.</p> <p>-TGL_PENERIMAAN_CAB: date, digunakan untuk menyimpan data tanggal penerimaan cabang.</p>	
<p>+pemesananCab() Default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada dikelas ini.</p> <p>+getDataPemesananCab () : Tabel Operasi ini digunakan untuk mengambil semua data dari tabel pemesanan.</p> <p>+getPemesananCabByTgl(): void Operasi ini digunakan untuk mencari data pemesanan berdasarkan tanggal.</p> <p>+getJumlahPemesananCabByTgl(id_pemesanan): void Operasi ini digunakan untuk mendapatkan jumlah pemesanan berdasarkan tanggal yang digunakan untuk kode pemesanan barang.</p> <p>+insDataPemesananCab (pemesanan) : void Operasi ini digunakan untuk memasukkan data pemesanan ke dalam database.</p> <p>+ubahDataPemesananCab (pemesanan) : void</p>	

Operasi ini digunakan untuk mengedit data pemesanan yang ada di database

+batalDataPemesananCab (id_pemesanan) : void

Operasi ini digunakan untuk membatalkan data pemesanan.

+insertBarangCab (barang) : void

Operasi ini digunakan untuk menambah data barang baru yang diterima.

+updateStatusPemesananCab (id_pemesanan) : void

Operasi ini digunakan untuk mengubah status pemesanan barang.

2.2.17.9 Specific Design Class detilPemesananCabangModel

DetilPemesananCabangModel	<<Entity>>
-JUMLAH_PESANCABANG: integer, digunakan untuk menyimpan data jumlah pemesanan cabang.	
-JUMLAH_BELUMDITERIMA: integer, digunakan untuk menyimpan data jumlah barang yang belum diterima.	
-TGL_PENERIMAAN_DETILCAB: date, digunakan untuk menyimpan data tanggal detil penerimaan cabang.	
+detilPemesananCab ()	
Default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada dikelas ini.	
+getDetilPemesananCab () : Tabel	
Operasi ini digunakan untuk menampilkan data detil pemesanan.	
+insDataDetailPemesananCab (pemesanan) : void	
Operasi ini digunakan untuk memasukkan data	

pemesanan ke dalam database.

```
+ubahDataDetilPemesananCab (pemesanan) : void
```

Operasi ini digunakan untuk mengedit data detil pemesanan yang ada di database

```
+hapusDataDetilPemesananCab (id_detil_pemesanan) : void
```

Operasi ini digunakan untuk menghapus data detil pemesanan yang ada di database

2.2.17.10 Specific Design Class trans_penjualanModel

Trans_penjualanModel	<<Entity>>
<p>-ID_TRANSAKSIJUAL: integer, digunakan untuk menyimpan data numerik identitas penjualan.</p> <p>-NO_FAKTUR: string, digunakan untuk menyimpan data kode unik transaksi penjualan.</p> <p>-TGL_TRANSAKSI: datetime, digunakan untuk menyimpan data tanggal penjualan barang.</p> <p>-TOTAL_HARGA: string, digunakan untuk menyimpan data total harga penjualan barang.</p> <p>-DISCOUNT: decimal, digunakan untuk menyimpan data discount penjualan.</p> <p>-TOTAL_STLH_DISC: string, digunakan untuk menyimpan data total harga setelah diskon penjualan barang.</p> <p>-KEMBALIAN: decimal, digunakan untuk menyimpan data kembalian dari pembayaran.</p>	
<pre>+penjualan()</pre> <p>Default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada di kelas ini.</p> <pre>+getDataPenjualan() : Tabel</pre>	

Operasi ini digunakan untuk mengambil semua data dari tabel penjualan.

+getPenjualanByTgl(): void

Operasi ini digunakan untuk mencari data penjualan berdasarkan tanggal.

+getJumlahPenjualanByTgl(id_penjualan): void

Operasi ini digunakan untuk mendapatkan jumlah penjualan berdasarkan tanggal yang digunakan untuk kode penjualan barang.

+insDataPenjualan(penjualan) : void

Operasi ini digunakan untuk memasukkan data transaksi penjualan ke dalam database.

+ubahDataPenjualan (penjualan) : void

Operasi ini digunakan untuk mengedit data transaksi penjualan yang ada di database

+batalDataPenjualan(id_penjualan) : void

Operasi ini digunakan untuk membatalkan data penjualan.

2.2.17.11 Specific Design Class detilTrans_penjualanModel

DetilTrans_penjualanModel	<<Entity>>
-JUMLAH_BARANG: integer, digunakan untuk menyimpan data jumlah barang yang dibeli.	
-HARGA_BARANG: string, digunakan untuk menyimpan data harga satuan barang.	
-DISC_ITEM: decimal, digunakan untuk menyimpan data discount item.	
-HARGA_STLH_DISC: string, digunakan untuk menyimpan data harga setelah diskon penjualan barang.	
-SUBTOTAL: string, digunakan untuk menyimpan data	

sub total harga penjualan barang.

+detilpenjualan()

Default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada dikelas ini.

+getDetilPenjualan() : Tabel

Operasi ini digunakan untuk menampilkan data detil penjualan.

+insDataDetailPenjualan(penjualan) : void

Operasi ini digunakan untuk memasukkan data penjualan ke dalam database.

+ubahDataDetilPenjualan(penjualan) : void

Operasi ini digunakan untuk mengedit data detil pemesanan yang ada di database

+hapusDataDetilPenjualan(id_detil_penjualan) : void

Operasi ini digunakan untuk menghapus data detil penjualan yang ada di database

+batalDataDetailPenjualan(id_detil_penjualan) : void

Operasi ini digunakan untuk membatalkan data detil penjualan.

3 PERANCANGAN DATA

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas User

Nama	Tipe	Panjang	Keterangan
ID_USER	int		Kode dari user, primary key
ID_ROLE	int		Kode role, foreign key
ID_CABANG	Int		Kode cabang, foreign key
USERNAME	Varchar	15	Nama sandi dari

ProgramStudiTeknikInformatika	DPPL-SISIP	101/151
Dokumeninidan informasi yang dimilikinya adalah milikProgramStudiTeknikInformatika-UAJYdanbersifatrahasia. Dilarang untukme-reproduksidokumen initanpadiketahuiolehProgramStudiTeknikInformatika		

			user
PASSWORD	Varchar	50	Kata sandi dari user
NAMA_USER	Varchar	50	Nama dari user
JENIS_KELAMIN	Varchar	15	Jenis Kelamin dari user
ALAMAT_USER	Varchar	200	Alamat dari user
EMAIL_USER	Varchar	100	Email dari user
FOTO_USER	Varchar	300	Foto dari user
IS_DELETED	bit		Kode status hapus semu dari user

3.1.2 Deskripsi Entitas Role

Nama	Tipe	Panjang	Keterangan
ID_ROLE	int		Kode dari role, primary key
NAMA_ROLE	varchar	30	Nama dari role

3.1.3 Deskripsi Entitas Cabang

Nama	Tipe	Panjang	Keterangan
ID_CABANG	Int		Kode dari pegawai, primary key
NAMA_CABANG	varchar	200	Alamat dari pegawai
ALAMAT_CABANG	varchar	20	Jenis Kelamin dari pegawai
NOTEKP_CABANG	varchar	300	Foto dari pegawai
IS_PUSAT	bit		Jika true (1) maka cabang tersebut menjadi pusat, jika false (0) maka tidak menjadi pusat
IS_DELETED	bit		Kode status hapus semu dari cabang

3.1.4 Deskripsi Entitas Supplier

Nama	Tipe	Panjang	Keterangan
ID_SUPPLIER	Int		Kode dari supplier, primary key
NAMA_SUPPLIER	Varchar	100	Nama dari supplier
ALAMAT_SUPPLIER	Varchar	200	Alamat dari supplier

NOTEPL_SUPPLIER	Varchar	30	No telp dari supplier
IS_DELETED	bit		Kode status hapus semu dari supplier

3.1.5 Deskripsi Entitas Kategori Barang

Nama	Tipe	Panjang	Keterangan
ID_KATEGORI_BARANG	Int		Kode dari kategori barang, primary key
NAMA_KATEGORI_BARANG	Varchar	100	Nama dari kategori barang
IS_DELETED	bit		Kode status hapus semu dari kategori barang

3.1.6 Deskripsi Entitas Barang

Nama	Tipe	Panjang	Keterangan
ID_BARANG	Int		Kode dari barang, primary key
ID_KATEGORI_BARANG	Int		Kode dari kategori barang, foreign key
NAMA_BARANG	varchar	100	Nama dari barang
STOK_BARANG	Int		Jumlah stok dari barang
SATUAN	varchar	15	Satuan barang (kilogram atau liter)
FOTO_BARANG	varchar	300	Foto dari barang
IS_DELETED	bit		Kode status hapus semu dari barang

3.1.7 Deskripsi Entitas Harga Satuan Barang

Nama	Tipe	Panjang	Keterangan
ID_HARGA_SATUAN_BARANG	Int		Kode dari harga

			satuan barang, primary key
ID_BARANG	Int		Kode dari barang, foreign key
ID_CABANG	Int		Kode dari cabang, foreign key
KODE_BARANG	Int		Kode barang untuk barcode
KEMASAN	Float		Kemasan dari barang (2.5, 5, 25 kg dst.)
HARGA_PER_KEMASAN	Decimal	18,0	Harga dari satuan barang
STOK	Int		Stok dari masing-masing satuan barang
IS_DELETED	bit		Kode status hapus semu dari harga satuan barang

3.1.8 Deskripsi Entitas Harga Satuan Barang

Nama	Tipe	Panjang	Keterangan
ID_BARANG_SUPPLIER	Int		Kode dari harga barang supplier, primary key
ID_SUPPLIER	Int		Kode dari supplier, foreign key
ID_BARANG	Int		Kode dari barang, foreign key

KODE_BARANG	Int		Kode barang untuk barcode
KEMASAN	Float		Kemasan dari barang (2.5, 5, 25 kg dst.)
HARGA_PER_KEMASAN	Decimal	18,0	Harga dari satuan barang
IS_DELETED	bit		Kode status hapus semu dari harga satuan barang

3.1.9 Deskripsi Entitas Pemesanan ke Supplier

Nama	Tipe	Panjang	Keterangan
ID_PESANSUPPLIER	Int		Kode dari pemesanan supplier, primary key
ID_SUPPLIER	Int		Kode supplier, foreign key
ID_USER	Int		Kode dari user, foreign key
KODE_PESANSUPPLIER			Kode unik dari pemesanan ke supplier
TGL_PESANSUPPLIER	Datetime		Tanggal saat input pemesanan ke supplier
TOTAL_HARGA	Decimal	18,0	Total harga dari pemesanan
STATUS_PESAN	Varchar	25	Status pesan dari pemesanan 'terkirim ke supplier'
TGL_PENERIMAAN_SUP	Date		Tanggal penerimaan barang dari supplier

3.1.10 Deskripsi Entitas Detil Pemesanan ke Supplier

Nama	Tipe	Panjang	Keterangan
ID_PESANSUPPLIER	Int		Kode dari pemesanan ke supplier, primary key
ID_BARANG	Int		Kode dari barang, primary key
ID_BARANG_SUPPLIER	Int		Kode dari barang supplier, primary kry
JUMLAH_PESANSUPPLIER	Int		Jumlah barang yang dipesan
HARGA_SUPPLIER	Decimal	18,0	Harga per satuan barang dari supplier
SUBTOTAL	Decimal	18,0	Sub total harga dari pemesanan
JUMLAH_BELUMDITERIMA	Int		Jumlah barang yang belum diterima
TGL_PENERIMAAN_DETILSUP	Date		Tanggal penerimaan detil pemesanan ke supplier

3.1.11 Deskripsi Entitas Pemesanan ke Pusat/Antar Cabang

Nama	Tipe	Panjang	Keterangan
ID_PESANCABANG	Int		Kode dari pemesanan cabang, primary key
ID_CABANGFROM	Int		Kode cabang asal stok, foreign key

ID_CABANGTO	Int		Kode cabang tujuan stok, foreign key
ID_USER	Int		Kode dari user, foreign key
KODE_PESANCABANG	Varchar	25	Kode unik dari pemesanan ke cabang
TGL_PESACABANG	Date		Tanggal saat input pemesanan ke cabang
STATUS_PESAN	Varchar	50	Status pesan dari pemesanan 'terkirim ke cabang'
TGL_PENERIMAAN_CAB	Date		Tanggal penerimaan barang dari cabang

3.1.12 Deskripsi Entitas Detil Pemesanan ke Pusat/Antar Cabang

Nama	Tipe	Panjang	Keterangan
ID_PESANCABANG	Int		Kode dari pemesanan ke pusat atau antar cabang, primary key
ID_BARANG	Int		Kode dari barang, primary key
ID_HARGA_SATUAN_BARANG	Int		Kode dari harga satuan barang, primary key
JUMLAH_PESANCABANG	Int		Jumlah barang yang dipesan
JUMLAH_BELUMDITERIMA	Int		Jumlah barang yang belum diterima
TGL_PENERIMAAN_DETILCAB	Date		Tanggal

			penerimaan detil pemesanan ke cabang
--	--	--	---

3.1.13 Deskripsi Entitas Transaksi Penjualan

Nama	Tipe	Panjang	Keterangan
ID_TRANSAKSIJUAL	Int		Kode dari transaksi penjualan, primary key
ID_CABANG	Int		Kode dari cabang, foreign key
ID_USER	Int		Kode dari user, foreign key
NO_FAKTUR	Varchar	30	No faktur transaksi
TGL_TRANSAKSI	Date		Tanggal dari penjualan
TOTAL_HARGA	Decimal	18,0	Total harga dari penjualan
DISCOUNT	Decimal	18,0	Diskon dari penjualan
TOTAL_STLH_DISC	Decimal	18,0	Total harga setelah didiskon dari penjualan
KEMBALIAN	Decimal	18,0	Sisa uang yang harus dikembalikan kepada konsumen

3.1.14 Deskripsi Entitas Detil Transaksi Penjualan

Nama	Tipe	Panjang	Keterangan
ID_TRANSAKSIJUAL	Int		Kode dari detil penjualan, primary key
ID_BARANG	Int		Kode dari barang, primary key
ID_HARGA_SATUAN_BARANG	Int		Kode dari harga satuan barang, primary key
JUMLAH_BARANG	Int		Jumlah dari barang yang penjualan
HARGA_BARANG	Decimal	18,0	Harga dari satu

			satuan barang
DISC_ITEM	Decimal	18,0	Diskon dari tiap item barang
HARGA_STLH_DISC	Decimal	18,0	Harga setelah didiskon
SUBTOTAL	Decimal	18,0	Sub total harga dari penjualan

3.1.15 Deskripsi Entitas Histori Stok Barang

Nama	Tipe	Panjang	Keterangan
ID_HISTORI_STOK	Int		Kode dari histori stok, primary key
ID_CABANG	Int		Kode dari cabang, foreign key
ID_HARGA_SATUAN_BARANG	Int		Kode dari harga satuan barang, foreign key
STOK_LAMA	Int		Stok lama dari barang
STOK_BARU	Int		Stok baru dari barang
TGL_HISTORY_STOCK	Date		Tanggal history dari stok, per bulan
REMARK_HISTORI_STOK	Varchar	100	Catatan perubahan stok

3.1.16 Deskripsi Entitas Histori Harga Barang

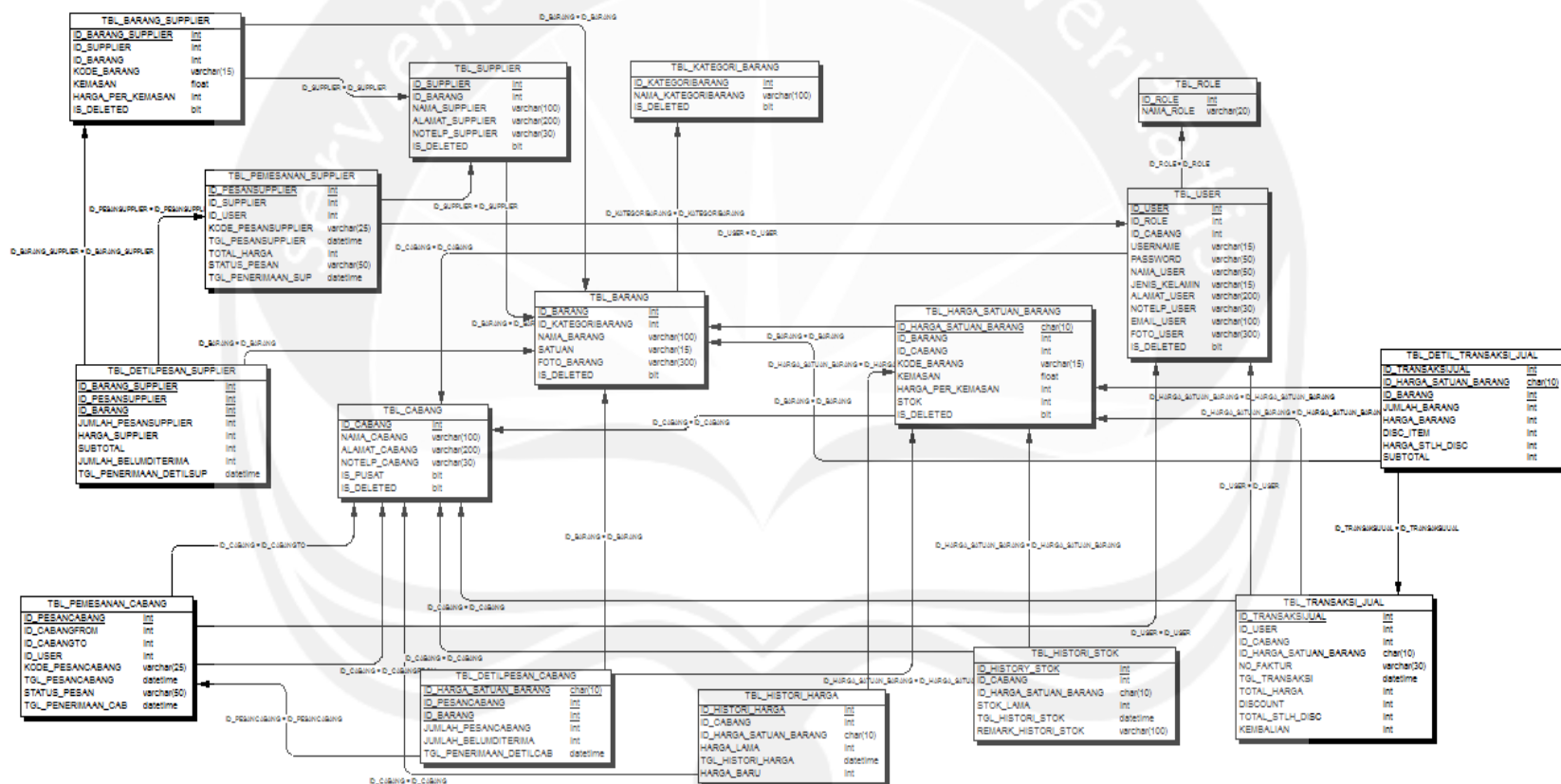
Nama	Tipe	Panjang	Keterangan
ID_HISTORI_HARGA	Int		Kode dari histori harga, primary key
ID_CABANG	Int		Kode dari cabang,

			foreign key
ID_HARGA_SATUAN_BARANG	Int		Kode dari harga satuan barang, foreign key
HARGA_LAMA	Decimal	18,0	Harga lama dari barang
HARGA_BARU	Decimal	18,0	Harga baru dari barang
TGL_HISTORY_HARGA	Date		Tanggal history dari harga, jika terjadi perubahan harga

3.1.17 Deskripsi Entitas Bulan

Nama	Tipe	Panjang	Keterangan
ID_BULAN	Int		Kode dari bulan, primary key
NAMA_BULAN	Varchar	15	Nama dari bulan dalam satu tahun

3.2 Physical Data Model

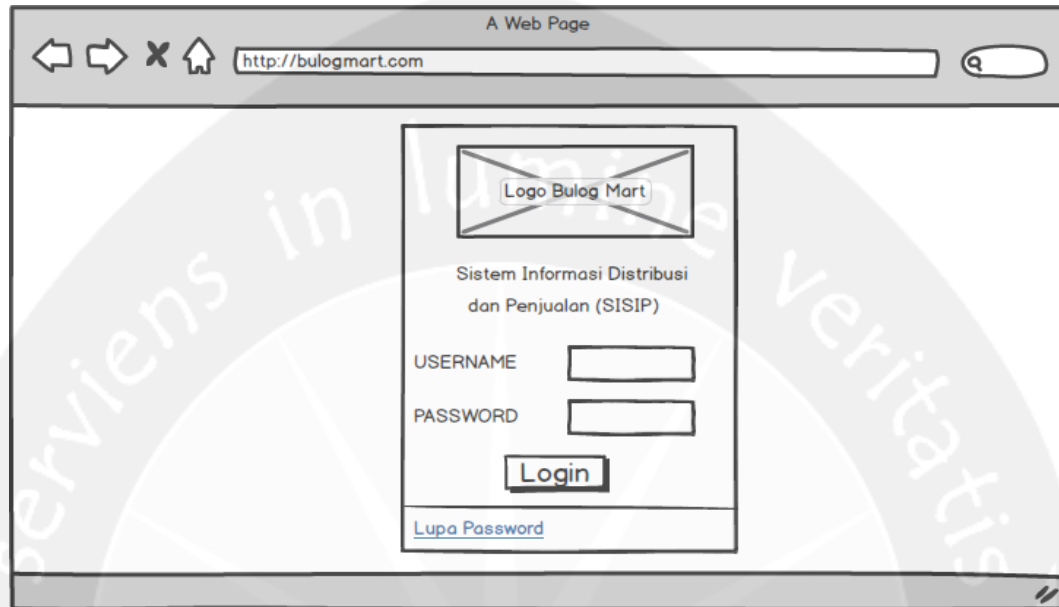


Gambar 2. 69 Physical Data Model SISIP

Program Studi Teknik Informatika	DPPL-SISIP	111/151
Dokumen ini adalah informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa dipadiketahui oleh Program Studi Teknik Informatika		

4 Perancangan Antarmuka

4.1 Antarmuka Halaman Login Web

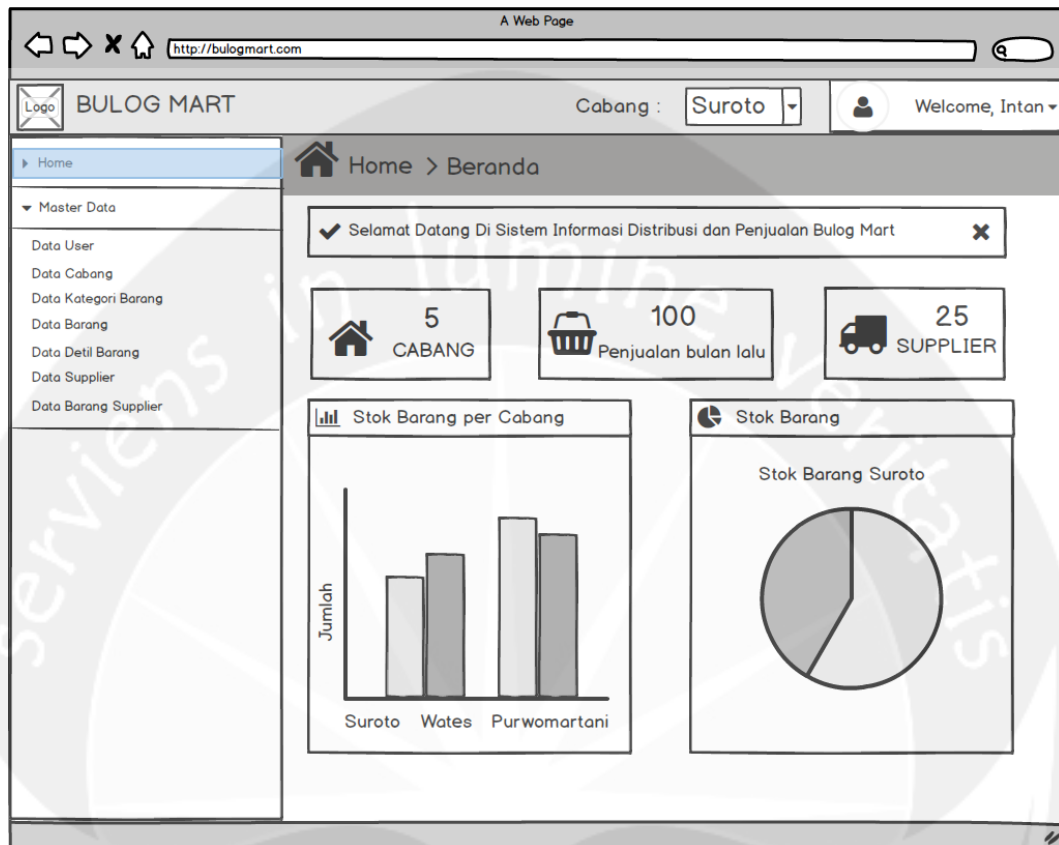


Gambar 4. 1 Rancangan Antarmuka *Login Web*

Antarmuka gambar 4.5 adalah rancangan antarmuka ketika pengguna melihat halaman awal pada aplikasi web. Pengguna dapat melakukan *login* ke dalam sistem. Pengguna memasukkan *username* dan *password* ke dalam *textbox* yang sudah disediakan, apabila pengguna lupa dengan *password*, pengguna dapat menghubungi *administrator* untuk mereset *password*, atau pengguna dapat mengklik link lupa *password* kemudian *password* akan dikirimkan secara otomatis ke *email* dari pengguna. Setelah *login* sistem akan menampilkan tampilan halaman beranda web dengan tampilan menu sesuai dengan *rolenya* masing-masing. Untuk *role* yang pertama yaitu *administrator* tampilan halaman beranda pada web akan terlihat seperti gambar 4.6.

ProgramStudiTeknikInformatika	DPPL-SISIP	112/151
Dokumeninidan informasi yang dimilikinya adalah milikProgramStudiTeknikInformatika-UAJYdanbersifatrahasia. Dilarang untukme-reproduksidokumen initanpadiketahuiolehProgramStudiTeknikInformatika		

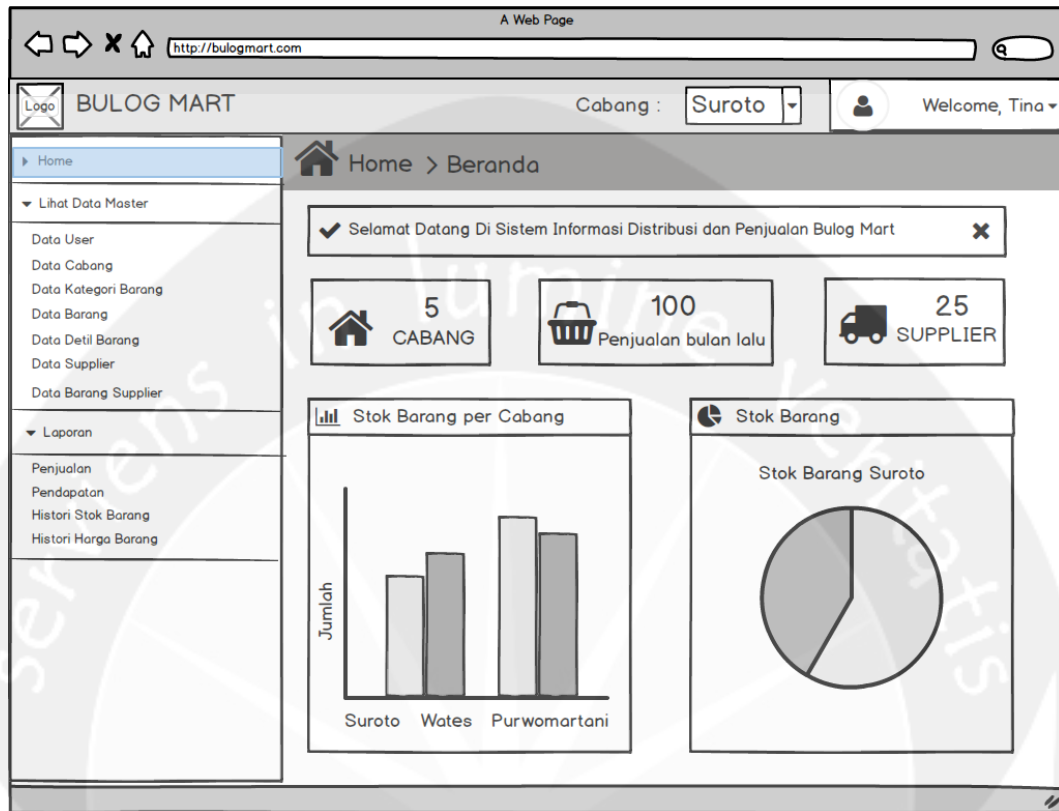
4.2 Antarmuka Halaman Beranda Web untuk Administrator



Gambar 4.2 Rancangan Antarmuka Halaman Beranda Web untuk Administrator

Pada gambar halaman beranda untuk administrator menunjukkan bahwa admin dapat mengakses semua menu untuk mengontrolnya. Menu yang disediakan secara lengkap adalah menu master data beserta sub menu data user, data cabang, data kategori barang, data barang, data detil barang, data *supplier*, dan data barang *supplier*.

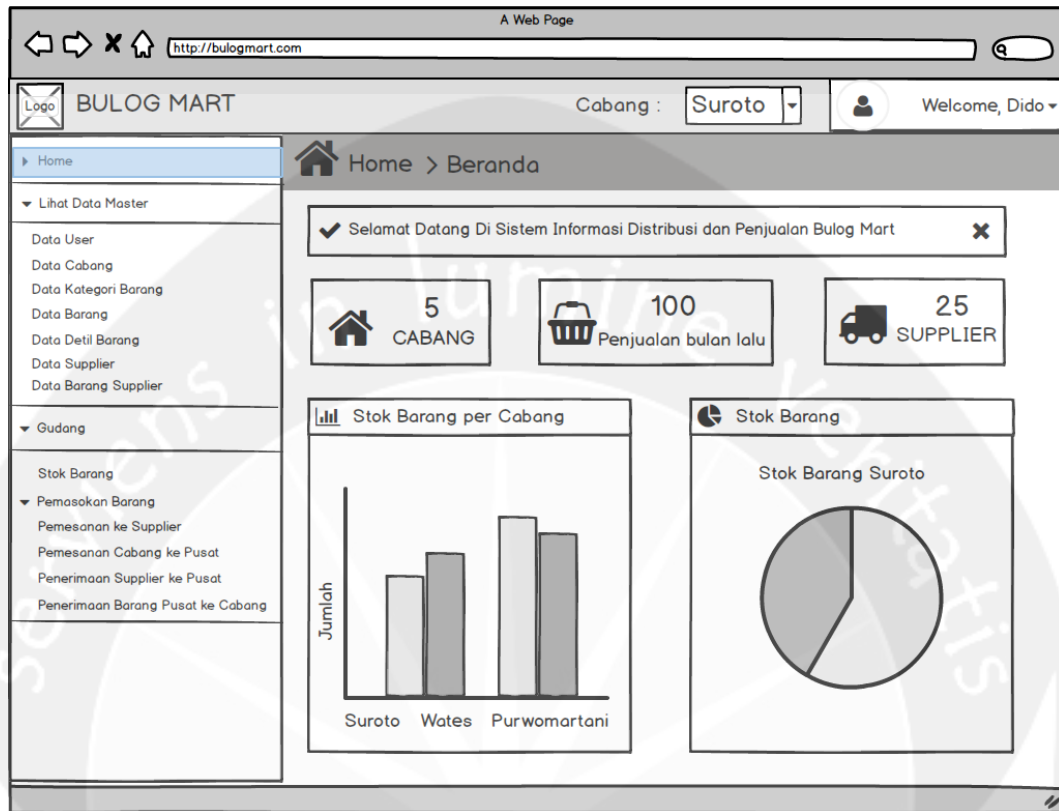
4.3 Antarmuka Halaman Beranda Web untuk KASI PPU



Gambar 4.3 Rancangan Antarmuka Halaman Beranda Web untuk KASI PPU

Pada gambar halaman beranda untuk administrator menunjukkan bahwa admin dapat mengakses semua menu untuk mengontrolnya. Menu yang disediakan adalah menu view data beserta sub menu data user, data cabang, data kategori barang, data barang, data detail barang, dan data supplier. Kemudian menu laporan dengan sub menu laporan untuk penjualan, laporan untuk pendapatan, histori stok, dan histori harga.

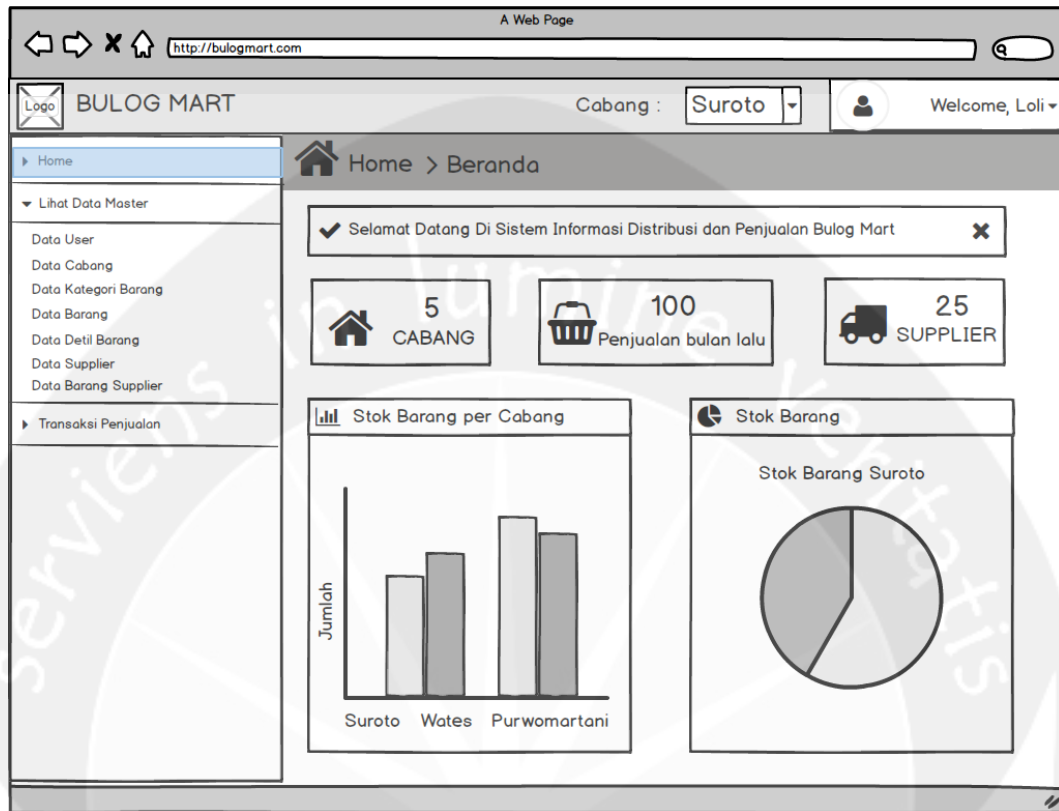
4.4 Antarmuka Halaman Beranda Web untuk Gudang



Gambar 4.4 Rancangan Antarmuka Halaman Beranda Web untuk Gudang

Pada gambar halaman beranda untuk administrator menunjukkan bahwa admin dapat mengakses semua menu untuk mengontrolnya. Menu yang disediakan adalah menu view data beserta sub menu data user, data cabang, data kategori barang, data barang, data detail barang, dan data *supplier*. Kemudian menu gudang, yang terdiri dari sub menu stok barang dan pemasokan barang. Di dalam sub menu pemasokan barang terdapat sub sub menu pemesanan ke *supplier*, pemesanan cabang ke pusat, penerimaan *supplier* ke pusat, dan penerimaan barang pusat ke cabang.

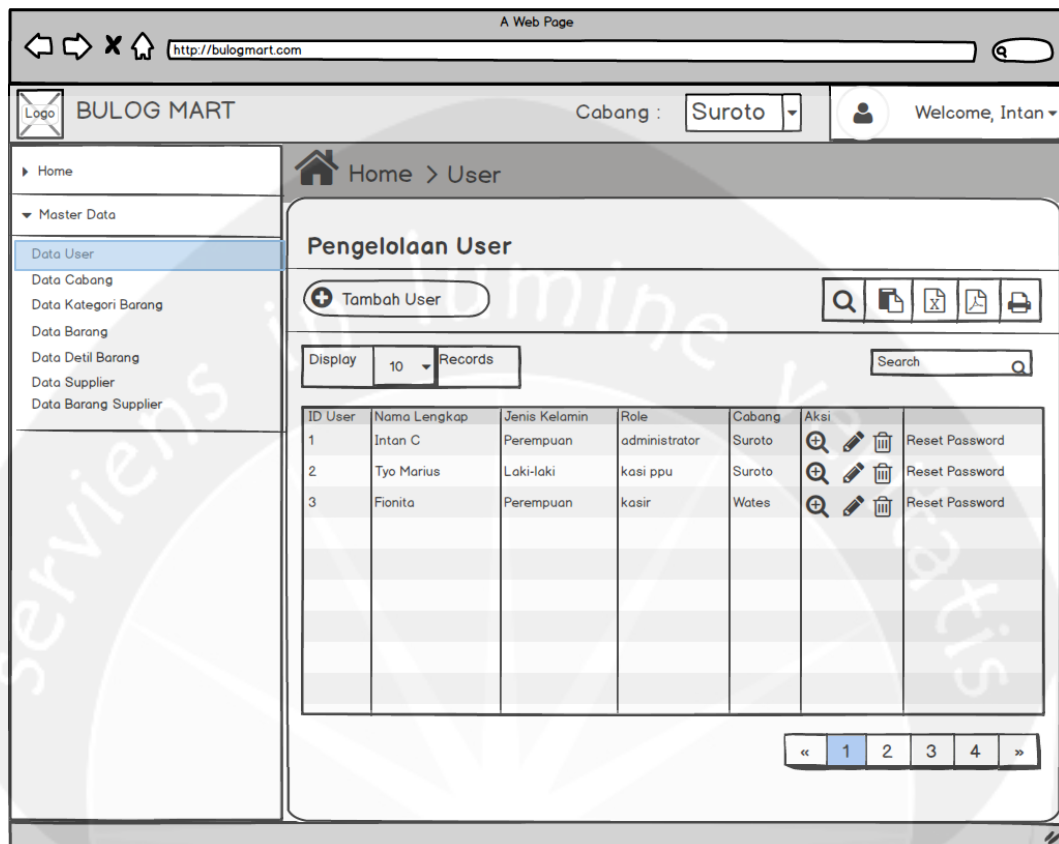
4.5 Antarmuka Halaman Beranda Web untuk Kasir



Gambar 4.5 Rancangan Antarmuka Halaman Beranda Web untuk Kasir

Pada gambar halaman beranda untuk administrator menunjukkan bahwa admin dapat mengakses semua menu untuk mengontrolnya. Menu yang adalah menu *view* data beserta sub menu data user, data cabang, data kategori barang, data barang, data detail barang, dan data *supplier*. Menu selanjutnya adalah transaksi penjualan.

4.6 Antarmuka Halaman Pengelolaan Data User



Gambar 4.6 Rancangan Antarmuka Halaman User

Gambar di atas menunjukkan antarmuka untuk pengelolaan user yang hanya dapat di kelola oleh admin. Admin dapat melihat data user, menambah data user, mengubah data user, menghapus data user, dan mereset password user tertentu. Yang pertama adalah pengelolaan untuk menambah data user. Pengguna dapat melakukan tambah data user dengan menekan tombol tambah user yang telah disediakan. Kemudian setelah menekan tombol tambah user maka form akan tampil seperti gambar 4.11 di bawah ini.

Gambar 4.7 Rancangan Antarmuka Tambah User

Gambar di atas merupakan rancangan antarmuka setelah pengguna menekan tombol tambah user. Form tambah user terdiri dari beberapa data yang harus diisi oleh pengguna. Dalam hal ini pengguna adalah administrator. Admin harus memilih cabang di mana user di tempatkan, kemudian setelah itu admin menetapkan *username* untuk user. Selanjutnya adalah memilih *role* dari user yang terdiri dari kasi ppu, kasir, atau gudang. Admin juga harus mengisi nama lengkap dari user baru yang akan ditambahkan, mengisi jenis kelamin laki-laki atau perempuan, mengisi alamat user, mengisi

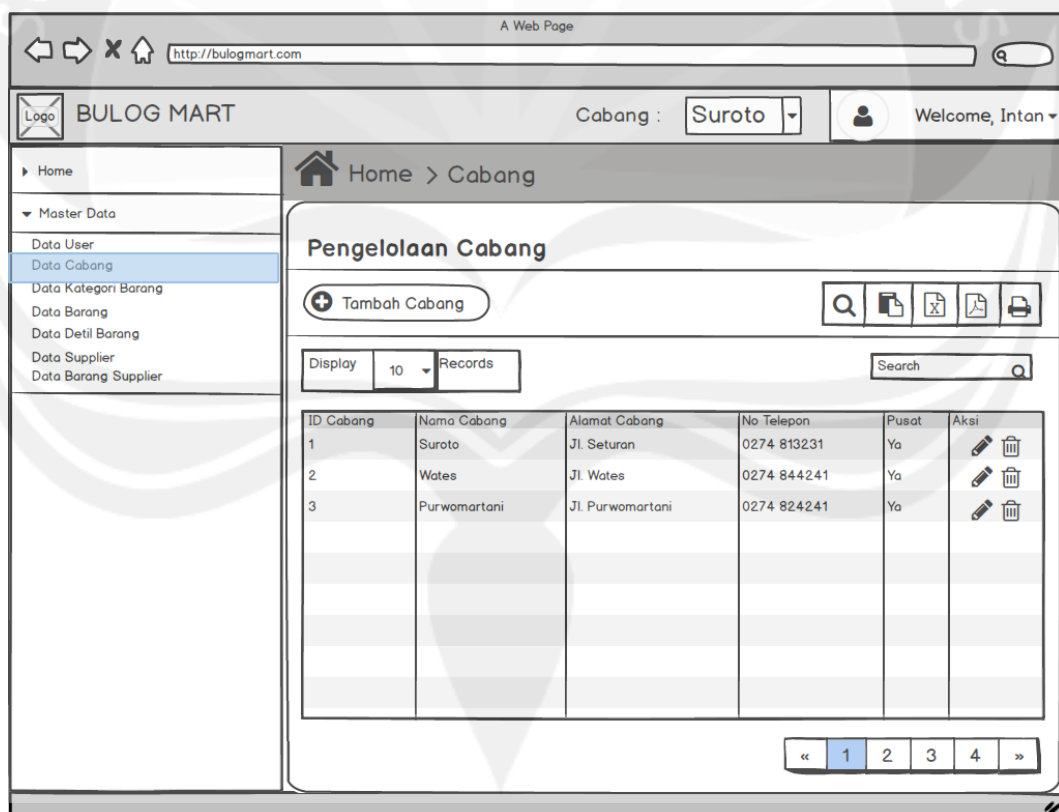
no telepon, mengisi *email*, dan mengupload foto user. Jika semua bagian data user telah terisi maka yang terakhir adalah pengguna menekan tombol simpan. Jika data user berhasil tersimpan ke dalam *database* maka secara otomatis data user baru akan terlihat pada tabel user. Jadi setelah pengguna menekan tombol simpan, maka halaman web akan kembali pada *view* tabel data user seperti pada gambar 4.10.

Gambar 4.8 Rancangan Antarmuka Ubah Data User

Gambar di atas merupakan rancangan antarmuka setelah pengguna menekan gambar pensil yang ada pada

kolom aksi untuk mengubah data user tertentu. Form ubah data user terdiri dari beberapa data yang harus diisi oleh pengguna. Dalam hal ini pengguna adalah administrator. Pengguna mengubah data sesuai keperluan, setelah itu jika semua bagian data user telah terisi maka yang terakhir adalah pengguna menekan tombol simpan. Jika data user berhasil tersimpan ke dalam database maka secara otomatis data user yang diubah akan terlihat pada tabel user. Jadi setelah pengguna menekan tombol simpan, maka halaman web akan kembali pada view tabel data user seperti pada gambar 4.10.

4.7 Antarmuka Halaman Pengelolaan Cabang



Gambar 4.9 Rancangan Antarmuka Halaman Cabang

Gambar di atas adalah rancangan antarmuka untuk pengelolaan data cabang yang hanya dapat dilakukan oleh admin. Admin dapat menambah data cabang, mengubah data cabang tertentu, dan menghapus data cabang tertentu. Jika admin ingin menambah data cabang maka admin dapat menekan tombol tambah cabang yang telah tersedia.

The screenshot shows a web browser window with the URL `http://bulogmart.com`. The page header includes the logo 'BULOG MART', a dropdown menu for 'Cabang' set to 'Suroto', and a user greeting 'Welcome, Intan'. The breadcrumb trail is 'Home > Cabang > Tambah Cabang'. The main content area is titled 'Tambah Cabang' and contains the following form elements:

- Nama Cabang:** A text input field.
- Alamat Cabang:** A text input field.
- Pusat Cabang:** A checkbox with the label 'Berilah tanda centang (v) jika cabang merupakan pusat'.

At the bottom of the form are two buttons: 'Simpan' and 'Batal'.

Gambar 4.10 Rancangan Antarmuka Halaman Tambah Cabang

Gambar di atas menunjukkan *form* untuk menambah data cabang yang baru. Pengguna yaitu admin wajib mengisi beberapa data yang telah disediakan. Admin harus mengisi nama cabang dan alamat cabang, setelah itu jika cabang tersebut berfungsi sebagai pusat maka pada pusat cabang di beri tanda centang. Dalam hal ini

cabang yang berfungsi sebagai pusat hanya satu cabang, tidak boleh ada dua. Setelah informasi terisi semua, data akan disimpan dengan menekan tombol Simpan. Kemudian jika berhasil tersimpan data cabang baru akan muncul pada tabel view cabang seperti gambar 4.14 .

Gambar 4.11 Rancangan Antarmuka Halaman Ubah Data Cabang

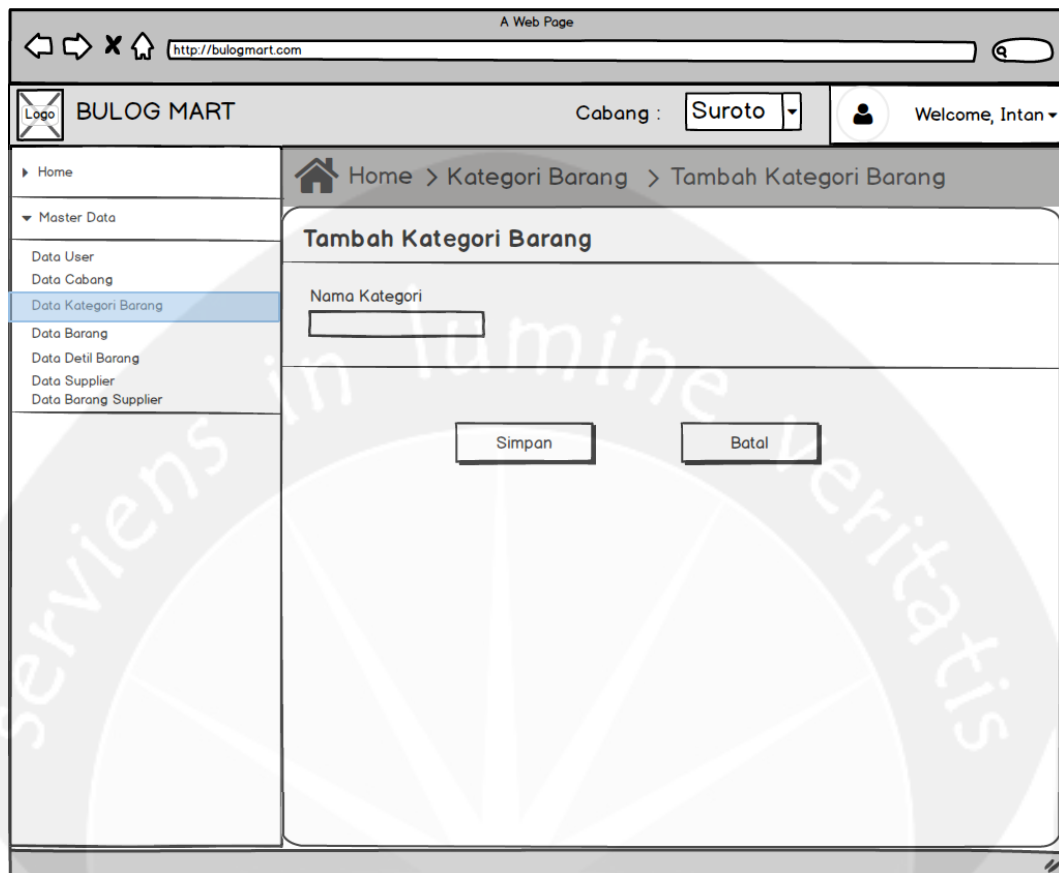
Gambar di atas menunjukkan *form* untuk mengubah data cabang tertentu. Pengguna mengubah data cabang sesuai keperluan. Setelah informasi terisi semua, data akan disimpan dengan menekan tombol Simpan. Kemudian jika berhasil tersimpan data cabang yang diubah akan muncul pada tabel *view* cabang seperti gambar 4.14 .

4.8 Antarmuka Halaman Kategori Barang



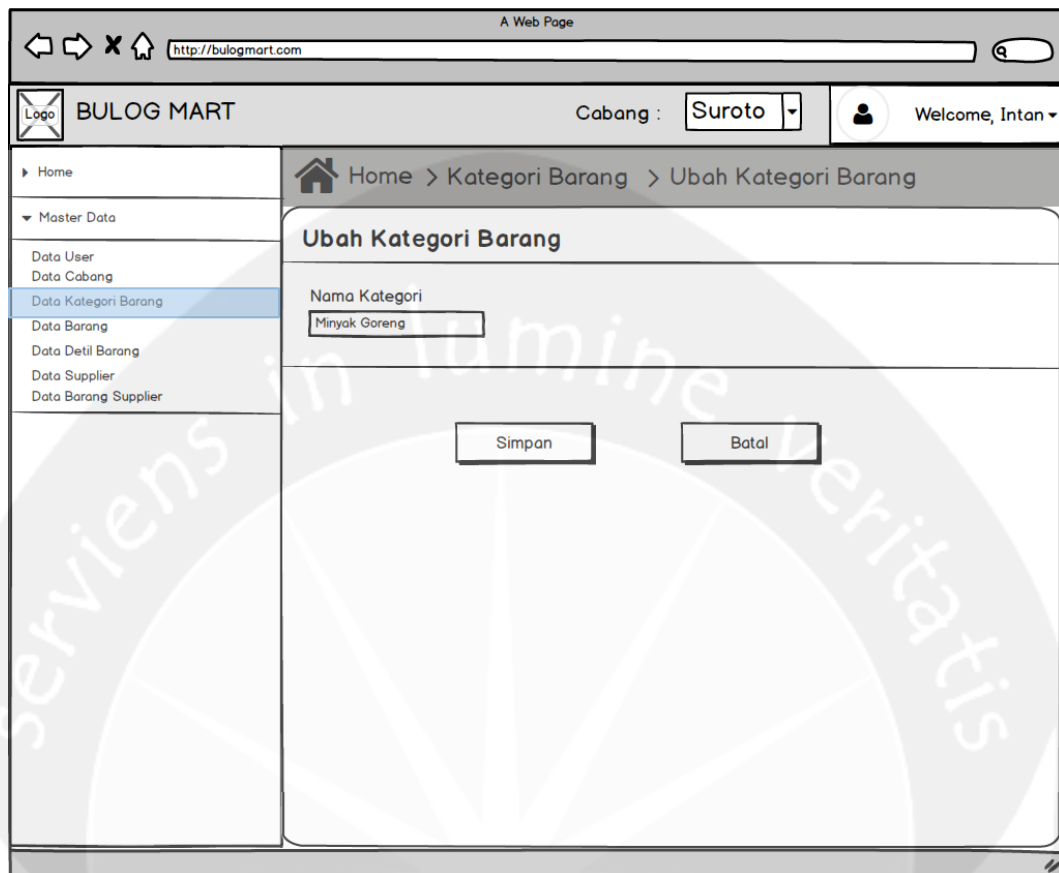
Gambar 4.12 Rancangan Antarmuka Pengelolaan Kategori Barang

Gambar di atas menunjukkan rancangan antarmuka jika kita mengklik menu kategori barang. Di halaman tersebut terdapat tabel yang akan menampilkan data kategori barang yang ada dalam *database* kategori barang. Pengguna yaitu admin dapat melakukan tambah data kategori, mengubah data kategori tertentu, dan menghapus data kategori tertentu. Yang pertama admin dapat menambah data kategori barang baru dengan menekan tombol tambah kategori yang telah disediakan, dan akan muncul *form* untuk tambah kategori seperti gambar 4.18.



Gambar 4.13 Rancangan Antarmuka Tambah Kategori Barang

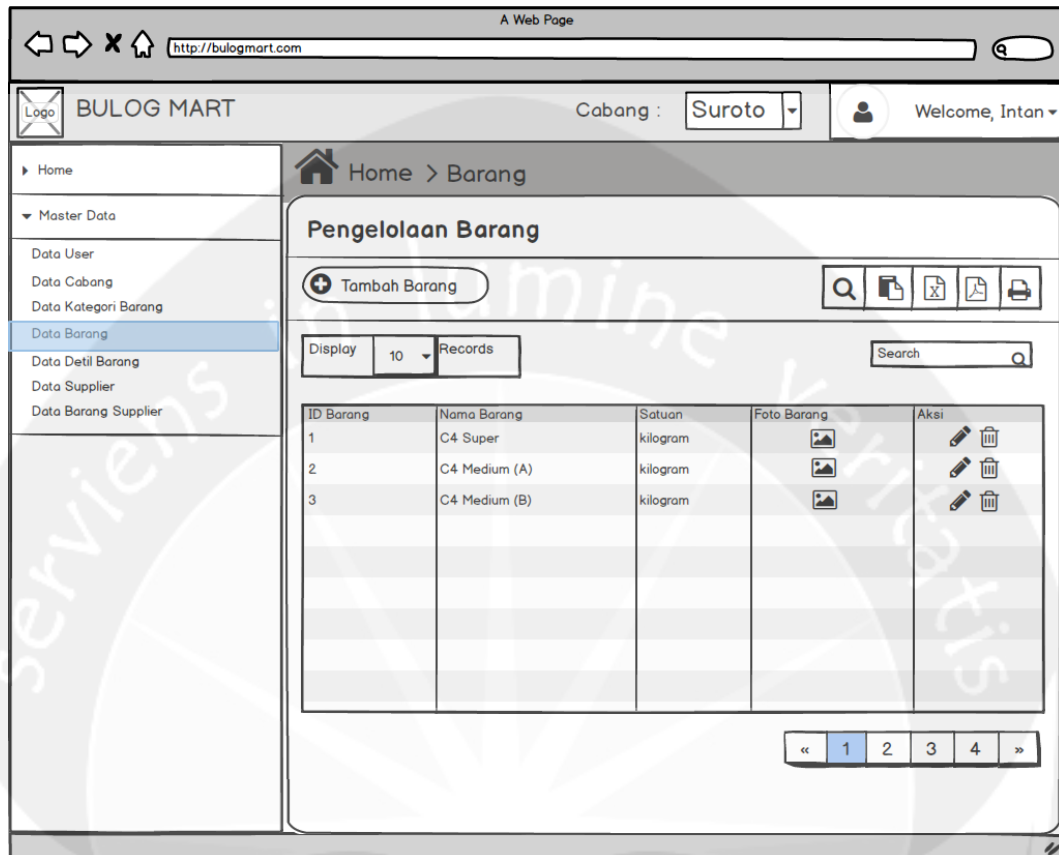
Gambar di atas menunjukkan data yang harus diisi oleh admin yaitu nama kategori barang, setelah terisi maka untuk menyimpannya admin harus menekan tombol simpan. Data kategori yang baru akan terlihat pada tabel seperti pada gambar 4.17.



Gambar 4.14 Rancangan Antarmuka Ubah Data Kategori Barang

Gambar di atas menunjukkan data yang akan diubah oleh admin yaitu nama kategori barang, setelah terisi maka untuk menyimpannya admin harus menekan tombol simpan. Data kategori yang baru akan terlihat pada tabel seperti pada gambar 4.17.

4.9 Antarmuka Halaman Pengelolaan Barang

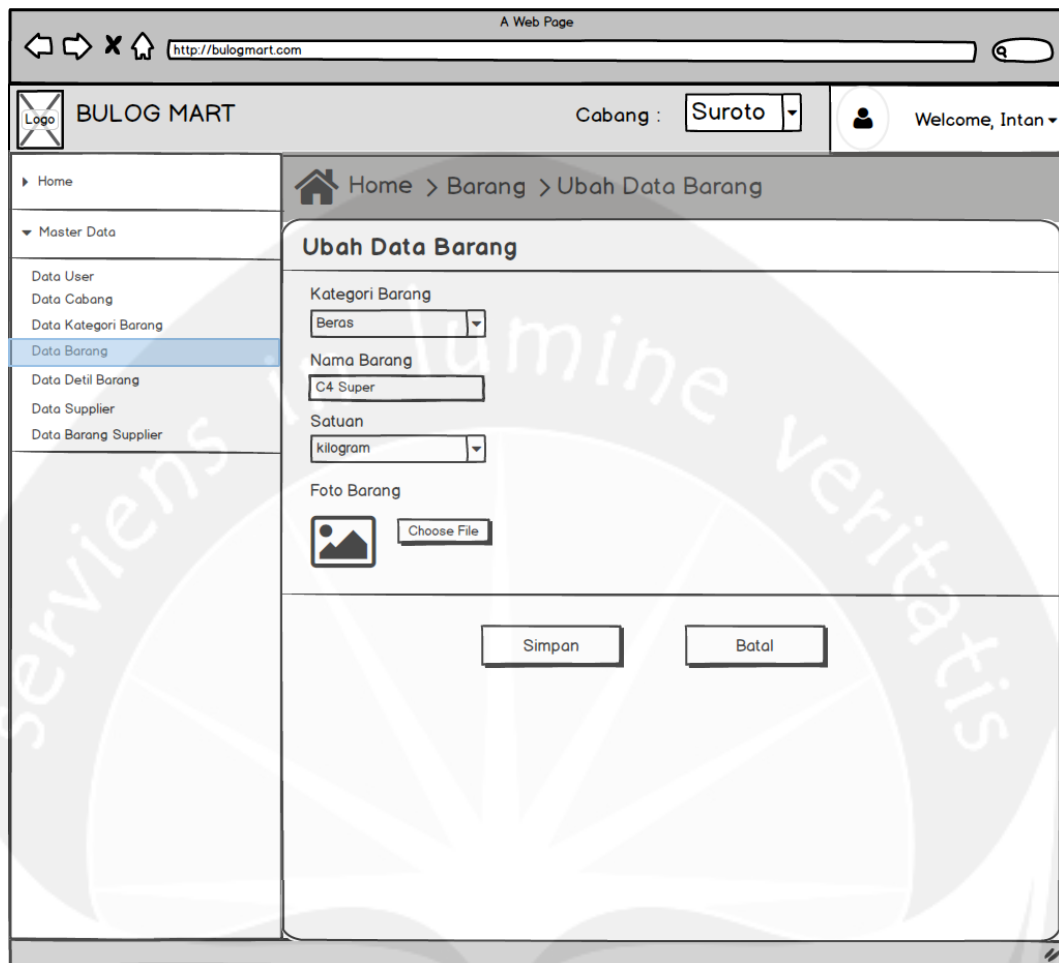


Gambar 4.15 Rancangan Antarmuka Pengelolaan Barang

Gambar di atas adalah rancangan antarmuka jika kita memilih menu barang. Pengguna yaitu admin dapat melakukan tambah data barang baru, mengubah data barang tertentu, dan menghapus data barang tertentu. Pengguna yaitu admin dapat melakukan tambah barang dengan cara menekan tombol tambah barang, kemudian akan muncul *form* barang seperti gambar 4.21 di bawah ini.

Gambar 4.16 Rancangan Antarmuka Tambah Barang

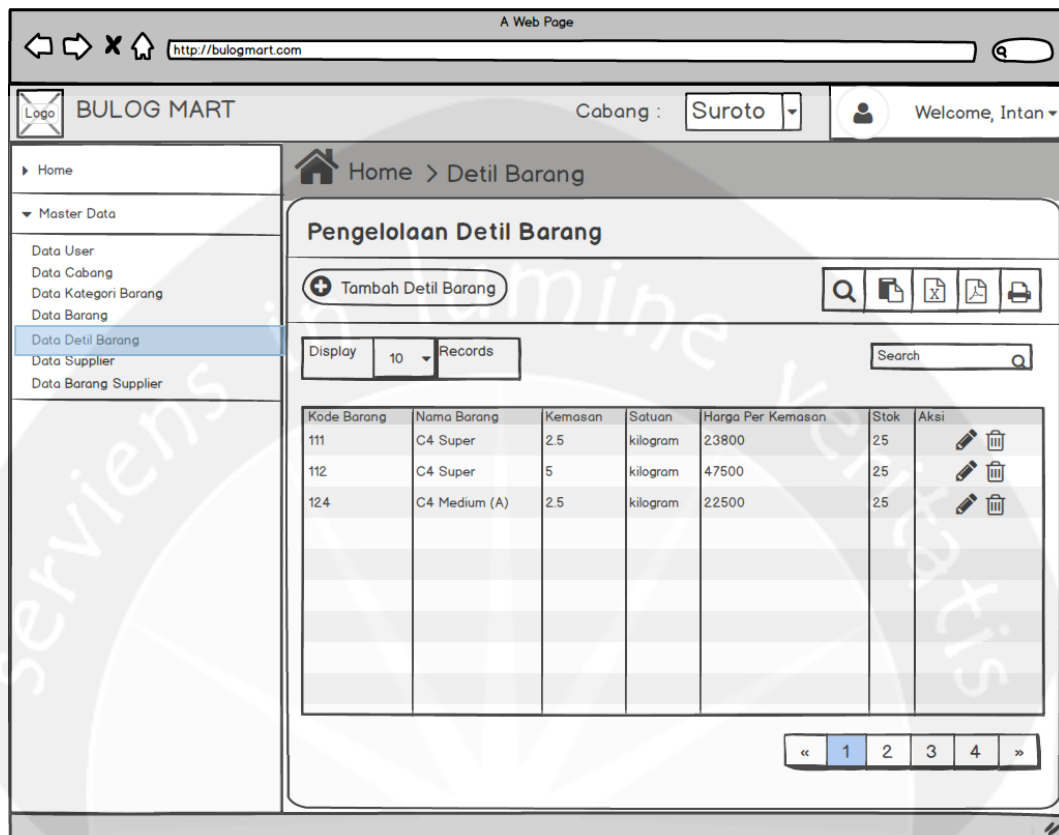
Gambar di atas adalah rancangan antarmuka untuk *form* tambah barang. Admin harus mengisi informasi yang ada, yang pertama dengan memilih kategori barang yaitu beras atau gula pasir atau minyak goreng. Kemudian mengisikan nama barang, memilih satuannya kilogram atau liter. Yang terakhir adalah mengupload foto barang. Kemudian jika semua informasi sudah di lengkapi, untuk meyimpan data barang baru klik tombol simpan. Jika data barang berhasil di simpan dalam *database*, maka data barang baru akan tertampil pada *view* data barang seperti pada gambar 4.20.



Gambar 4.17 Rancangan Antarmuka Ubah Data Barang

Gambar di atas adalah rancangan antarmuka untuk *form* ubah data barang tertentu setelah pengguna mengklik ikon pensil pada kolom aksi. Admin mengubah data sesuai keperluan, kemudian jika semua informasi sudah di lengkapi, untuk meyimpan data barang baru klik tombol simpan. Jika data barang berhasil di simpan dalam *database*, maka data barang yang diubah akan tertampil pada *view* data barang seperti pada gambar 4.20.

4.10 Antarmuka Halaman Detil Barang



Gambar 4.18 Rancangan Antarmuka Pengelolaan Detil Barang

Gambar di atas menunjukkan rancangan antarmuka dari pengelolaan detil barang yang dikelola oleh Admin. Admin dapat melakukan tambah detil barang, ubah data detil barang, dan hapus data detil barang. Tambah detil barang dapat dilakukan dengan cara menekan tombol tambah data detil barang. Kemudian akan muncul *form* seperti gambar 4.24 di bawah ini.

The screenshot shows a web browser window with the URL `http://bulogmart.com`. The page title is "A Web Page". The header includes the "BULOG MART" logo, a branch selector set to "Suroto", and a user greeting "Welcome, Intan". The main content area is titled "Tambah Detil Barang" and contains the following form fields:

- Kode Barang**: A text input field.
- Nama Barang**: A dropdown menu with "C4 Super" selected.
- Kemasan**: A text input field.
- Harga Per Kemasan**: A text input field.
- Jumlah Stok**: A spinner control set to "1".

At the bottom of the form are two buttons: "Simpan" and "Batal". A sidebar menu on the left lists "Master Data" items, with "Data Detil Barang" highlighted. A breadcrumb trail at the top of the content area reads "Home > Detil Barang > Tambah Detil Barang".

Gambar 4.19 Rancangan Antarmuka Tambah Detil Barang

Gambar di atas menunjukkan rancangan antarmuka untuk *form* tambah detil barang. Kode barang akan di *generate* secara otomatis, pilih nama barang yang tersedia, masukkan data kemasan pada kasus ini kemasan hanya ada 2.5 kg atau 5 kg atau 25 kg untuk beras. Masukkan harga per kemasan, dan jumlah stok barang tersebut. Setelah semua informasi telah diisi, untuk menyimpan data detil barang yang baru admin harus mengklik tombol simpan. Jika data detil barang baru berhasil disimpan maka akan muncul pada tabel detil

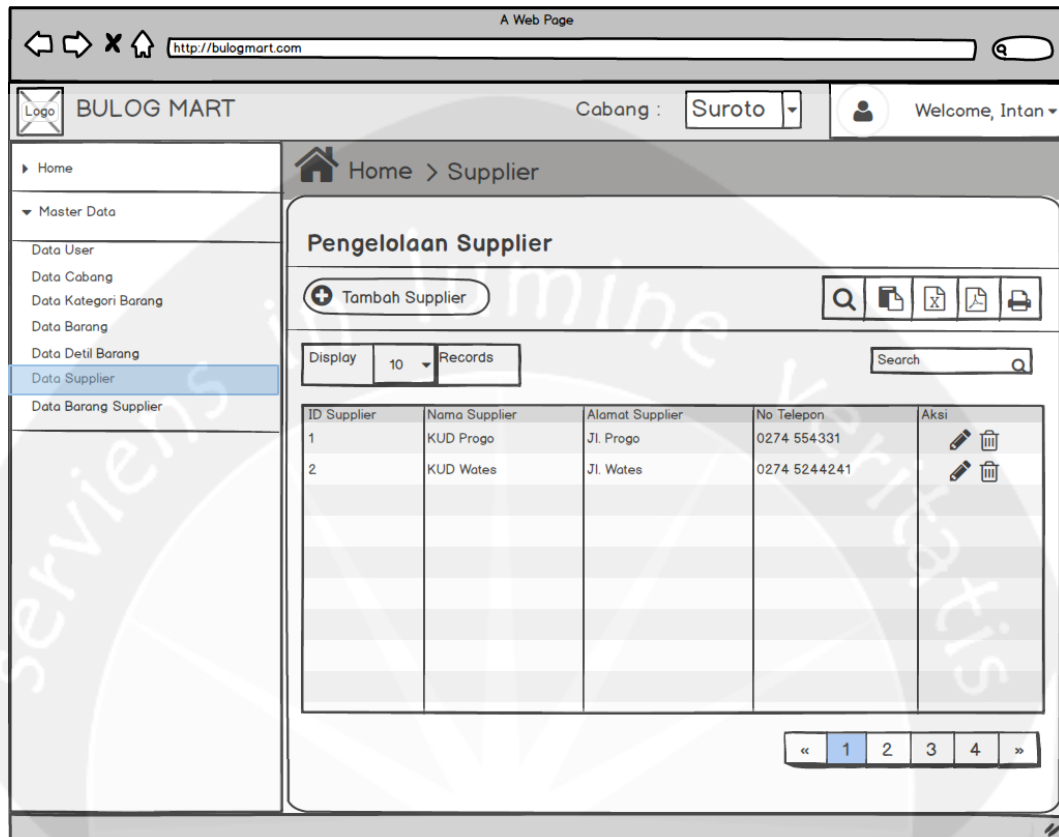
barang yang ada di awal saat memilih menu detail barang seperti gambar 4.23.

The screenshot shows a web browser window with the URL <http://bulogmart.com>. The page title is "A Web Page". The header includes the "BULOG MART" logo, a branch selector set to "Suroto", and a user greeting "Welcome, Intan". The breadcrumb trail is "Home > Detil Barang > Ubah Detil Barang". The left sidebar menu includes "Home", "Master Data", "Data User", "Data Cabang", "Data Kategori Barang", "Data Barang", "Data Detil Barang" (highlighted), "Data Supplier", and "Data Barang Supplier". The main form area is titled "Ubah Detil Barang" and contains the following fields: "Kode Barang" (text input with value "111"), "Nama Barang" (dropdown menu with value "C4 Super"), "Kemasan" (text input with value "2.5"), "Harga Per Kemasan" (text input with value "20000"), and "Jumlah Stok" (spin button with value "1"). At the bottom of the form are two buttons: "Simpan" and "Batal".

Gambar 4.20 Rancangan Antarmuka Ubah Data Detil Barang

Gambar di atas menunjukkan rancangan antarmuka untuk *form* ubah detail barang setelah mengklik ikon pensil pada kolom aksi. Admin mengubah data seperlunya, setelah semua informasi telah diisi, untuk menyimpan data detail barang yang diubah admin harus mengklik tombol simpan. Jika data detail barang yang diubah berhasil disimpan maka akan muncul pada tabel detail barang yang ada di awal saat memilih menu detail barang seperti pada gambar 4.23.

4.11 Antarmuka Halaman Pengelolaan *Supplier*



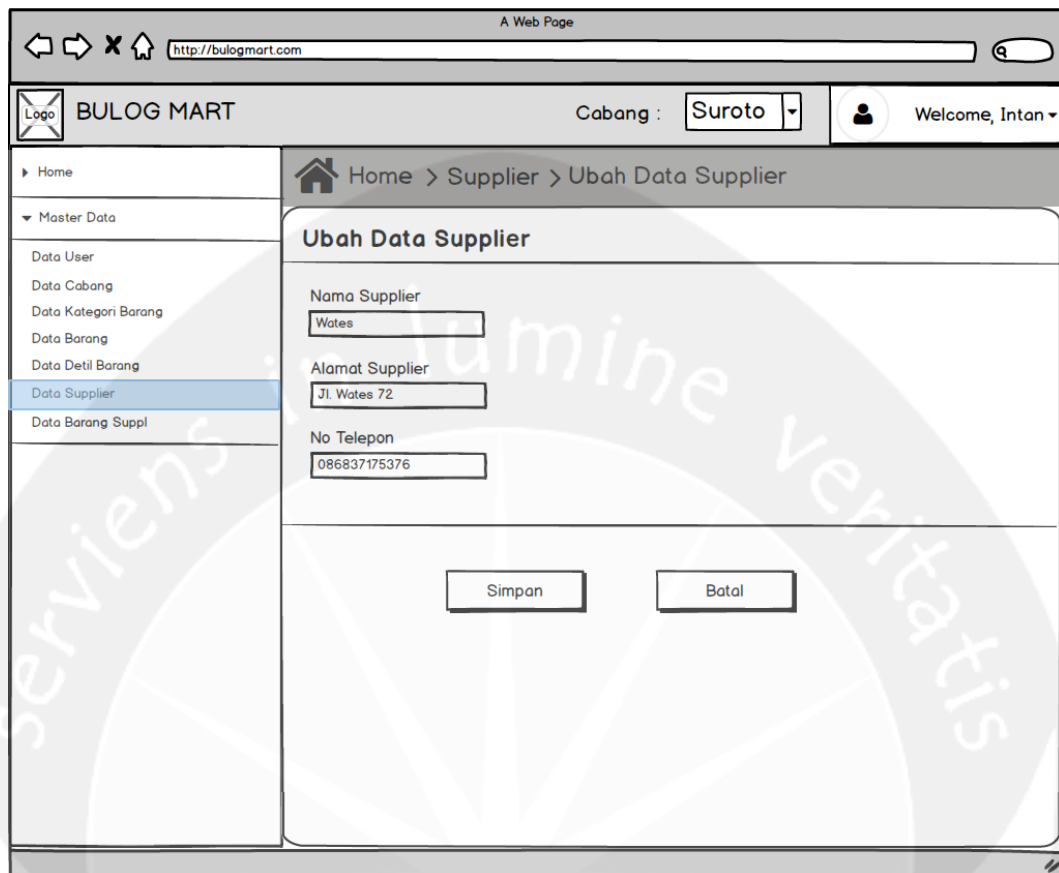
Gambar 4.21 Rancangan Antarmuka Pengelolaan *Supplier*

Gambar di atas menunjukkan rancangan antarmuka untuk pengelolaan *supplier* yang hanya bisa dilakukan oleh Admin saja. Pengelolaan tersebut terdiri dari tambah *supplier*, ubah data *supplier*, dan hapus data *supplier* tertentu. Jika ingin menambah *supplier* baru, Admin dapat menekan tombol tambah *supplier* yang sudah disediakan. *Form* tambah *supplier* akan muncul seperti gambar di bawah ini.

The screenshot shows a web browser window with the URL `http://bulogmart.com`. The page title is "BULOG MART". The branch is set to "Suroto" and the user is logged in as "Intan". The navigation path is "Home > Supplier > Tambah Supplier". The main form is titled "Tambah Supplier" and contains three input fields: "Nama Supplier", "Alamat Supplier", and "No Telepon". Below the form are two buttons: "Simpan" and "Batal". A left sidebar menu lists "Master Data" items, with "Data Supplier" highlighted.

Gambar 4.22 Rancangan Antarmuka Tambah *Supplier*

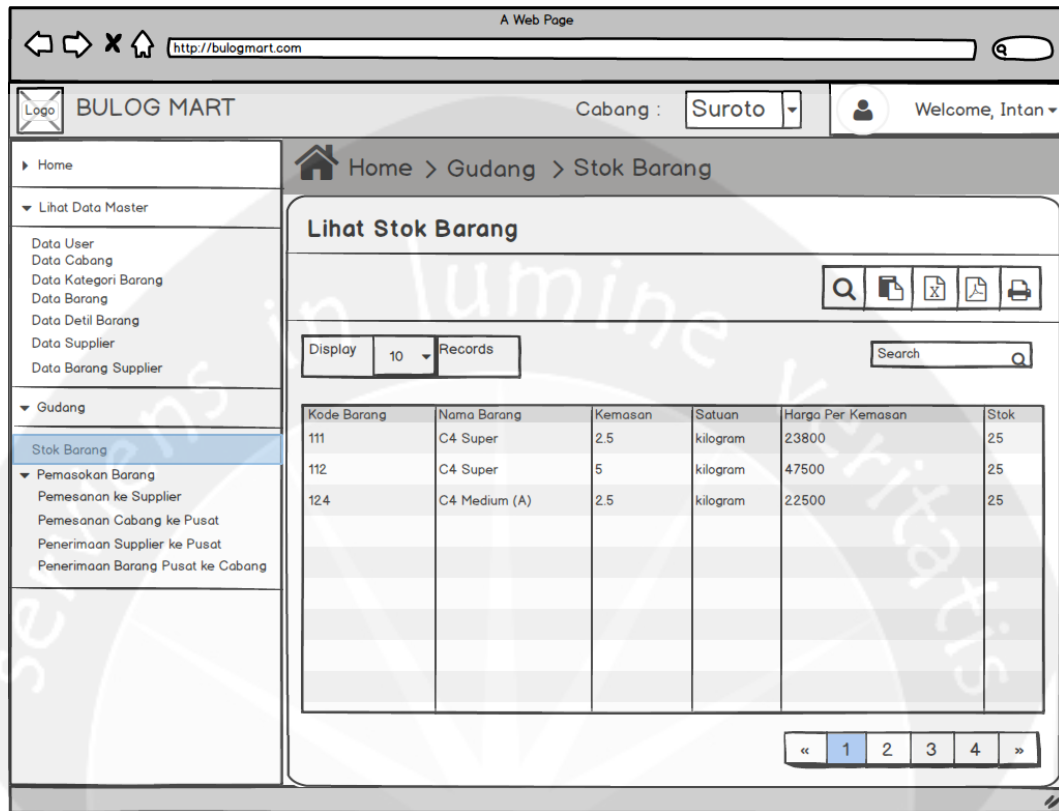
Gambar di atas menunjukkan rancangan antarmuka tambah *supplier*. *Form* tersebut terdiri dari nama *supplier*, alamat *supplier*, dan no telepon *supplier*. Admin harus mengisi informasi tersebut, kemudian klik tombol simpan jika ingin menyimpan data *supplier* yang baru. Jika data *supplier* yang baru telah berhasil disimpan maka akan muncul ke dalam tabel *supplier* yang terlihat saat Admin memilih menu data *supplier*.



Gambar 4.23 Rancangan Antarmuka Ubah Data Supplier

Gambar di atas menunjukkan rancangan antarmuka ubah data *supplier*. Admin mengubah data kemudian klik tombol simpan jika ingin menyimpan data *supplier* yang sudah diubah. Jika data *supplier* yang diubah telah berhasil disimpan maka akan muncul ke dalam tabel *supplier* yang terlihat saat Admin memilih menu data *supplier* seperti pada gambar 4.26.

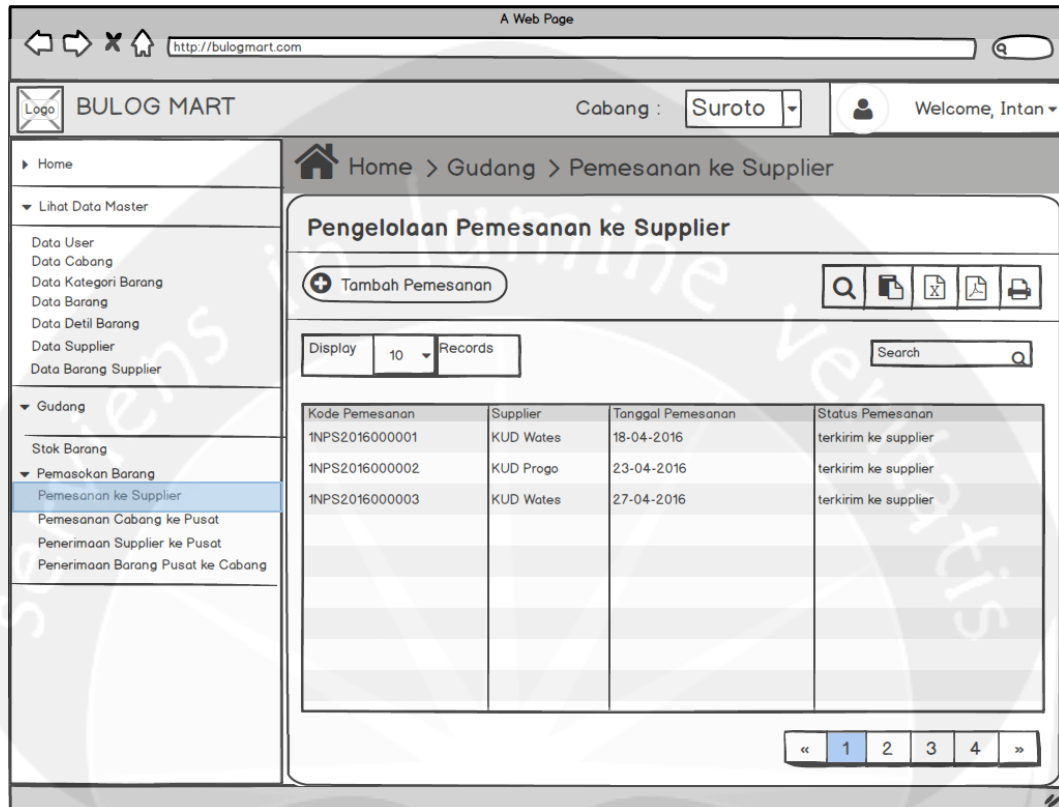
4.12 Antarmuka Halaman View Stok Barang oleh Gudang



Gambar 4.24 Rancangan Antarmuka Lihat Stok Barang

Gambar di atas menunjukkan rancangan antarmuka untuk melihat stok barang pada gudang tertentu. Antarmuka tersebut hanya tersedia pada menu gudang yang hanya bisa diakses oleh pihak Gudang Bulog Mart baik cabang maupun pusat. Sub menu stok barang ini memudahkan pihak gudang untuk melihat stok terakhir dari cabangnya.

4.13 Antarmuka Halaman Pengelolaan Pemesanan Barang ke Supplier



Gambar 4.25 Rancangan Antarmuka Pengelolaan Pemesanan Barang ke *Supplier*

Gambar di atas adalah rancangan antarmuka untuk pengelolaan pemesanan barang ke *supplier*. Proses pemesanan barang ke *supplier* ini hanya dapat diakses oleh pihak gudang pusat. Pihak gudang dapat memesan ke *supplier* dengan cara menekan tombol tambah pemesanan yang sudah tersedia. Kemudian *form* untuk tambah pemesanan ke *supplier* akan terlihat seperti gambar 4.31 di bawah ini.

The screenshot shows a web browser window with the URL `http://bulogmart.com`. The page title is "A Web Page". The header includes the "BULOG MART" logo and the branch name "Suroto". The user is logged in as "Intan". The breadcrumb trail is "Home > Gudang > Pemesanan ke Supplier > Tambah Pemesanan".

The main form is titled "Tambah Pemesanan ke Supplier". It contains the following fields:

- Kode Pemesanan: 1NPS2016000004
- Supplier: KUD Progo
- Tanggal Pemesanan: / /

Below these fields is the "Tambah Detil Pemesanan" section, which includes:

- Kode Barang: [input field]
- Nama Barang: [input field]
- Jumlah Stok: 1
- Tombol: "Tambah Barang", "Pesan Barang", "Batal"

On the right side of the "Tambah Detil Pemesanan" section, there is a table with the following data:

No	Nama Barang	Kemasan	Jumlah	Sub Total	Aksi
1	C4 Super	2.5 kilogram	3	45.000	[edit] [delete]
2	C4 Super	5 kilogram	1	30.000	[edit] [delete]

The table also includes a search bar and a "Display 10 Records" indicator.

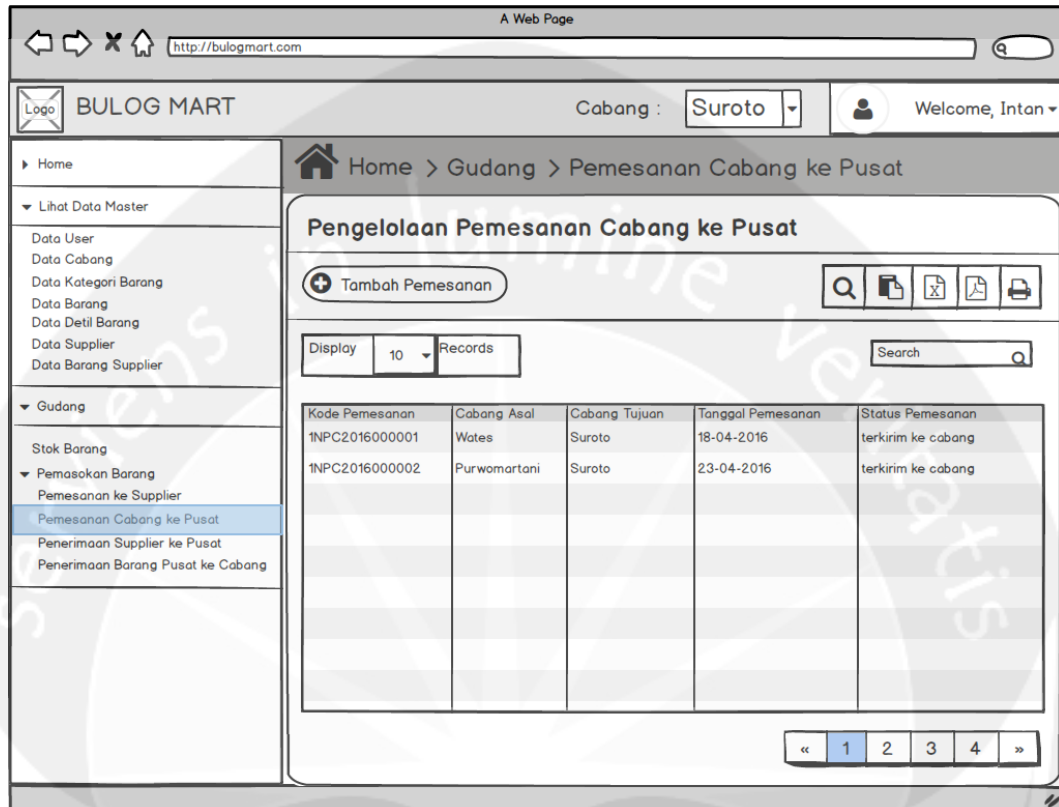
Gambar 4.26 Rancangan Antarmuka Tambah Pemesanan Barang ke Supplier

Gambar di atas menunjukkan rancangan antarmuka *form* untuk menambah pemesanan barang ke *supplier*. Seperti yang terlihat di gambar bahwa kode pemesanan akan ter-*generate* secara otomatis, kemudian gudang memilih *supplier* yang akan menjadi tempat pemesanan barang, tanggal pemesanan juga akan ter-*generate* secara otomatis sesuai tanggal dilakukan pemesanan. Masuk ke tambah detil pemesanan, pihak gudang dapat memasukkan kode barang, jika tidak tahu kode barangnya gudang dapat mencari barang tersebut dengan mengklik tombol

cari barang yang ada di sebelah *textbox* kode barang. Kemudian akan muncul *pop up* untuk memilih barang yang ada di *supplier* tersebut. Gudang lalu memilih salah satu barang dengan cara mengklik barang tersebut, jumlah stok yang akan dipesan diisi sesuai dengan kebutuhan pesanan. Lalu klik tombol tambah barang, dan barang yang akan dipesan akan muncul pada tabel di sebelahnya. Jika ingin menambah daftar barang yang akan dipesan, gudang memasukkan kode barang atau mencari barang yang akan dipesan selanjutnya. Alurnya sama seperti memesan barang pertama, jika pemesanan dirasa sudah cukup. Hal terakhir adalah menyimpan data pemesanan dan memprosesnya dengan cara mengklik tombol pesan barang. Setelah mengklik tombol pesan barang, maka nota pembelian barang ke *supplier* akan muncul. Karena *supplier* berada di luar sistem maka dibutuhkan nota pembelian yang nantinya akan dicetak dan ditandatangani oleh pihak *supplier*. Nota pembelian ke *supplier* terdiri dari informasi tanggal kode pemesanan, tanggal pemesanan, pemesan, beserta daftar barang yang akan dibeli ke *supplier*.

Program Studi Teknik Informatika	DPPL-SISIP	138/151
Dokumen ini adalah 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		

4.14 Antarmuka Halaman Pengelolaan Pemesanan Barang Cabang ke Pusat



Gambar 4.27 Rancangan Antarmuka Pengelolaan Pemesanan Barang Cabang ke Pusat

Gambar di atas menunjukkan rancangan antarmuka untuk pengelolaan pemesanan barang cabang ke pusat. Pemesanan barang dari cabang ke pusat tidak menyertakan harga dari barang tersebut. Dalam kasus Bulog Mart pemesanan barang dapat dilakukan dari cabang satu ke cabang yang lainnya. Pihak gudang pada cabang tertentu dapat melakukan pemesanan dengan cara menekan tombol tambah pemesanan yang sudah tersedia. Setelah itu, akan masuk ke dalam *form* tambah pemesanan cabang ke pusat.

The screenshot shows a web browser window with the URL <http://bulogmart.com>. The page title is "A Web Page". The header includes the "BULOG MART" logo, a branch selection dropdown set to "Suroto", and a user greeting "Welcome, Intan". The breadcrumb trail is "Home > Gudang > Pemesanan Cabang ke Pusat > Tambah Pemesanan".

The main content area is titled "Tambah Pemesanan Cabang ke Pusat" and contains the following form elements:

- Kode Pemesanan:** A text input field containing "1NPC2016000004".
- Cabang Asal:** A dropdown menu with "Wates" selected.
- Cabang Tujuan:** A dropdown menu with "Suroto" selected.
- Tanggal Pemesanan:** A date input field with a calendar icon.

Below these fields is a section titled "Tambah Detil Pemesanan" with a search bar and a table:

No	Nama Barang	Kemasan	Jumlah	Aksi
1	C4 Super	2.5 kilogram	3	
2	C4 Super	5 kilogram	1	

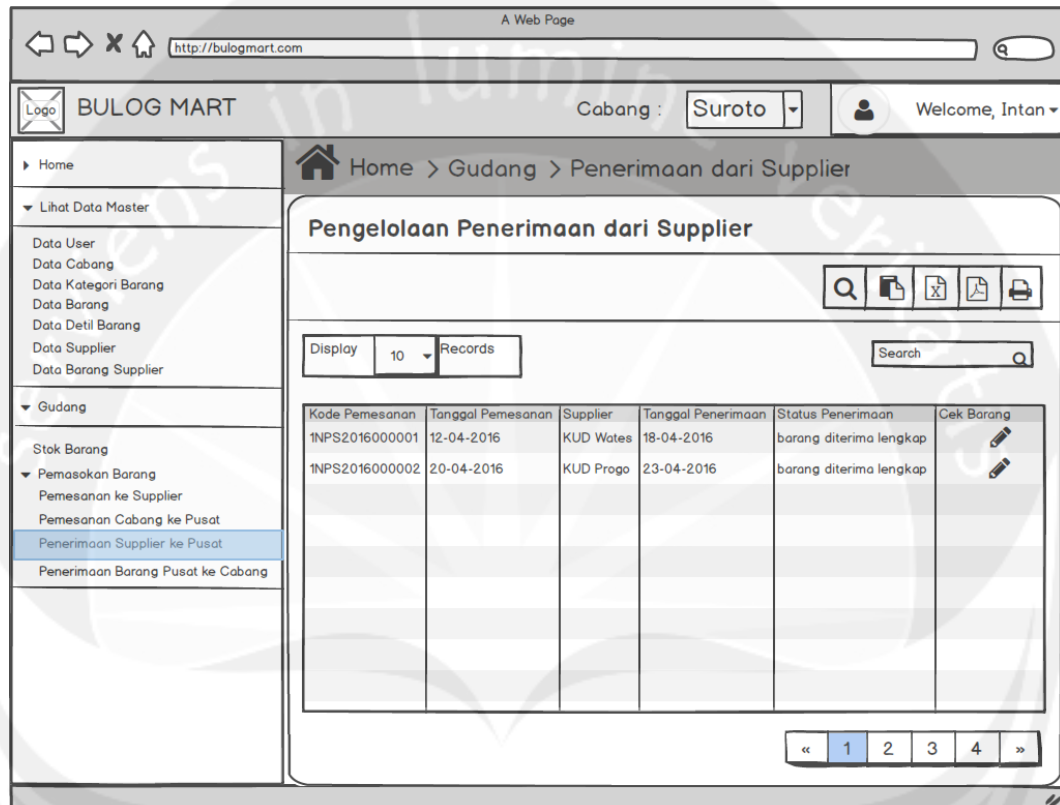
At the bottom of the form are two buttons: "Pesan Barang" and "Batal".

Gambar 4.28 Rancangan Antarmuka Tambah Pemesanan Barang Cabang ke Pusat

Gambar di atas menunjukkan rancangan antarmuka untuk tambah pemesanan barang. Kode pemesanan akan ter-generate secara otomatis, cabang asal akan disesuaikan dengan pihak gudang pada cabang tertentu. Selanjutnya gudang harus memilih cabang tujuan. Tanggal pemesanan akan ter-generate secara otomatis. Pada area tambah detil barang, gudang akan menambah daftar barang yang akan dipesan. Jika sudah maka yang terakhir pihak gudang akan mengklik tombol pesan barang, dan data pemesanan akan bertambah kemudian muncul nota pemesanan

dari cabang ke pusat atau cabang ke cabang lainnya pada Divre DIY.

4.15 Antarmuka Halaman Pengelolaan Penerimaan Barang dari Supplier



Gambar 4.29 Rancangan Antarmuka Pengelolaan Penerimaan dari Supplier

Gambar di atas merupakan rancangan antarmuka untuk pengelolaan penerimaan barang dari *supplier*. Pengelolaan ini hanya dapat diakses oleh pihak gudang. Jika pihak gudang ingin mengecek barang yang sudah diterima serta belum diterima, gudang dapat memilih data sesuai dengan kode pemesanannya dengan mengklik icon pensil pada kolom cek barang untuk menambah data

penerimaan. Selanjutnya akan muncul *form* seperti di bawah ini.

The screenshot shows a web browser window with the URL <http://bulogmart.com>. The page title is "A Web Page". The header includes the BULOG MART logo, a branch selector set to "Suroto", and a user greeting "Welcome, Intan". The breadcrumb trail is "Home > Gudang > Penerimaan dari Supplier > Checking & Ubah Status Terima".

The main content area is titled "Checking & Ubah Status Terima" and contains the following elements:

- Form Fields:**
 - Kode Pemesanan:
 - Supplier:
 - Tanggal Pemesanan: (with a calendar icon)
- Daftar Barang Table:**

Nama Barang	Kemasan	Jumlah	Cek Lengkap	Kekurangan
C4 Super	2.5 kilogram	5	<input type="checkbox"/>	<input type="text" value="3"/> (with up/down arrows)
C4 Super	5 kilogram	3	<input checked="" type="checkbox"/>	<input type="text" value="0"/> (with up/down arrows)
- Buttons:** A "Simpan" button is located at the bottom of the form.

The left sidebar contains a navigation menu with categories like "Lihat Data Master", "Gudang", and "Pemasakan Barang".

Gambar 4.30 Rancangan Antarmuka Checking Barang dari Supplier dan Ubah Status Terima

Gambar di atas merupakan rancangan antarmuka untuk menambah data penerimaan dari *supplier*. Data Pemesanan termasuk kode pemesanan, *supplier*, dan tanggal penerimaan akan ter-generate secara otomatis. Pada tabel daftar barang, terdapat daftar barang sesuai dengan pemesanan ke *supplier*. Lakukan *checking* barang dengan cara mencentang (v) barang yang sudah diterima pada kolom *check*, jika barang yang diterima kurang maka

gudang menuliskan jumlah kekurangan barang yang diterima. Jika sudah maka gudang bisa menyimpan data penerimaan dengan mengklik tombol Simpan. Untuk status penerimaan jika barang sudah tercentang semua atau tidak ada kekurangan barang maka status akan berubah menjadi barang diterima lengkap, jika tidak semua tercek atau ada kekurangan barang yang diterima maka status penerimaan akan diubah menjadi barang diterima belum lengkap.

4.16 Antarmuka Halaman Pengelolaan Penerimaan Barang dari Pusat ke Cabang

The screenshot shows a web browser window with the URL <http://bulogmart.com>. The page title is "BULOG MART" and the user is logged in as "Welcome, Intan". The breadcrumb trail is "Home > Gudang > Penerimaan Pusat ke Cabang". The main content area is titled "Pengelolaan Penerimaan Pusat ke Cabang" and features a table with the following data:

Kode Pemesanan	Tanggal Pemesanan	Cabang Asal	Tanggal Penerimaan	Status Penerimaan	Cek Barang
1NPC2016000001	16-04-2016	Suroto	18-04-2016	barang diterima lengkap	
1NPC2016000002	22-04-2016	Suroto	23-04-2016	barang diterima lengkap	

The interface also includes a search bar, a display/records selector (set to 10 records), and a pagination control at the bottom showing page 1 of 4.

Gambar 4.31 Rancangan Antarmuka Pengelolaan Penerimaan Pusat ke Cabang

Gambar di atas menunjukkan rancangan antarmuka untuk mengelola data penerimaan dari pusat ke cabang. Pengelolaan penerimaan ini hanya dapat diakses oleh pihak gudang cabang. Gudang hanya perlu memilih data pesanan sesuai kode pesanan kemudian klik icon pensil pada kolom cek barang untuk mengecek barang yang diterima kemudian akan muncul form seperti gambar di bawah ini.

The screenshot shows a web browser window with the URL 'http://bulogmart.com'. The page title is 'A Web Page'. The header includes the 'BULOG MART' logo, a 'Cabang' dropdown menu set to 'Suroto', and a user greeting 'Welcome, Intan'. The breadcrumb trail is 'Home > Gudang > Penerimaan dari Pusat ke Cabang > Checking & Ubah Status Terima'. The main content area is titled 'Checking & Ubah Status Terima' and contains the following form elements:

- Kode Pemesanan:** Input field with value '1NPS2016000004'.
- Cabang Pengirim:** Input field with value 'Suroto'.
- Tanggal Penerimaan:** Input field with value '1 / 1' and a calendar icon.
- Daftar Barang:** A table with the following data:

Nama Barang	Kemasan	Jumlah	Cek Lengkap	Kekurangan
C4 Super	2.5 kilogram	5	<input type="checkbox"/>	3
C4 Super	5 kilogram	3	<input checked="" type="checkbox"/>	0
- Simpan:** A button at the bottom of the form.

Gambar 4.32 Rancangan Antarmuka Tambah Penerimaan Pusat ke Cabang

Gambar di atas menunjukkan rancangan antarmuka untuk mengecek barang yang diterima dari pusat ke cabang atau dari cabang ke cabang. Kode pemesanan,

cabang pengirim, dan tanggal penerimaan akan terisi secara otomatis sesuai dengan dibuatnya data penerimaan barang. Pada tabel daftar barang akan muncul data barang apa yang dipesan, kemudian pihak gudang akan mencek barang yang sudah diterima, jika barang yang dikirim kurang maka akan ditulis kekurangannya pada kolom kekurangan. Jika sudah maka klik tombol simpan untuk menyimpan data penerimaan barang, status penerimaan akan secara otomatis terbentuk, jika barang belum diterima sepenuhnya maka statusnya "barang diterima belum lengkap", jika barang lengkap maka status penerimaannya adalah "barang diterima lengkap".

Program Studi Teknik Informatika	DPPL-SISIP	145/151
Dokumen ini adalah 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		

4.17 Antarmuka Transaksi Penjualan

The screenshot shows the BULOG MART sales transaction interface. The page is titled "A Web Page" and the URL is "http://bulogmart.com". The header includes the BULOG MART logo, the branch name "Suroto", and a user greeting "Welcome, Intan". The left sidebar contains a navigation menu with options like "Home", "Lihat Data Master", "Transaksi Penjualan", and "Laporan". The main content area is divided into three sections: "Data Transaksi", "Input Data Barang", and "Daftar Barang".

Data Transaksi

No. Faktur: INJ2016000001
 Tanggal Penjualan: / /
 Kasir: Fionita
 Total Harga: 0.00
 Bayar: 0.00
 Kembali: 0.00

Input Data Barang

Kode Barang: Cari Barang
 Nama Barang:
 Jumlah Stok: 1
 Diskon Item:

Daftar Barang

	Kode Barang	Nama Barang	Jumlah	Harga	Disc Item	SubTotal	Total
	111	Beras C4 Super 2.5 kilogram	2	23500	0	47000	47000

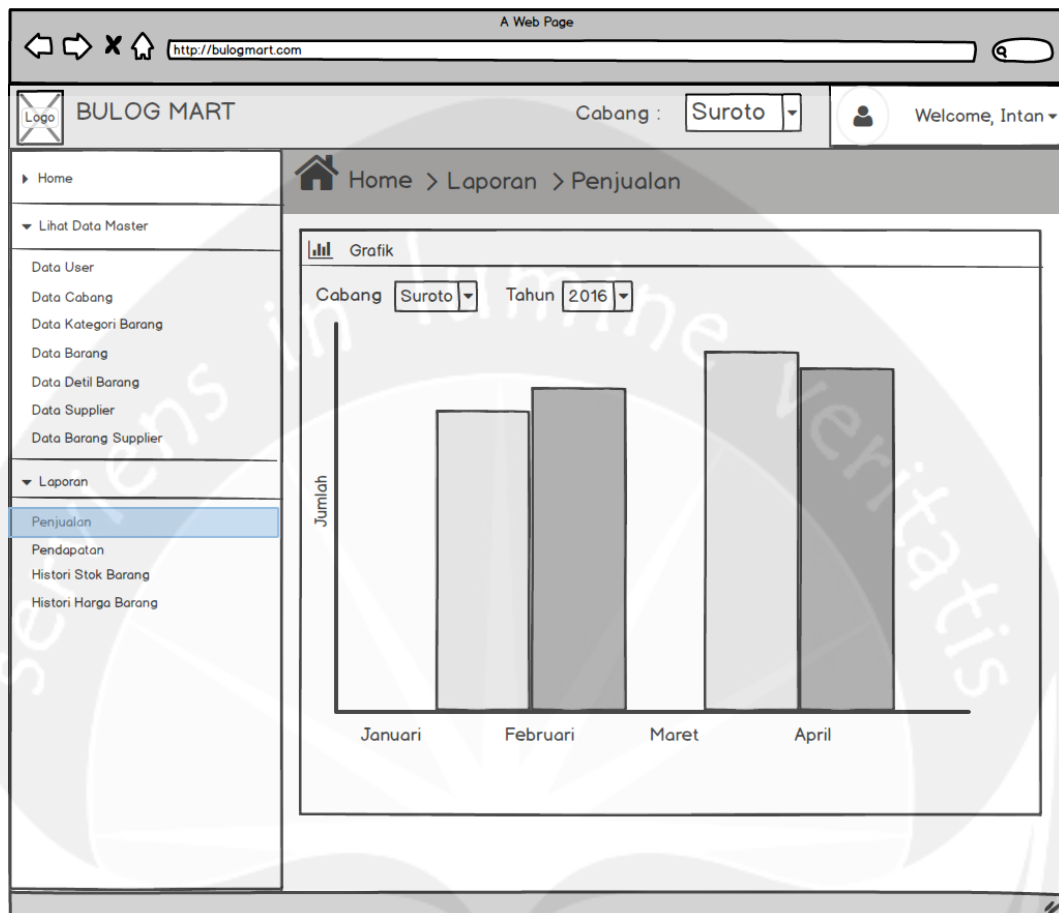
Gambar 4.33 Rancangan Antarmuka Transaksi Penjualan

Gambar di atas menunjukkan rancangan antarmuka untuk transaksi penjualan yang ada pada masing-masing cabang Bulog Mart Divre DIY. Dapat terlihat pada gambar di atas terdapat tiga bagian penting yaitu data transaksi, input data barang, dan daftar barang. Pada bagian data transaksi no faktur akan ter-generate secara otomatis, tanggal penjualan juga sesuai dengan transaksi penjualan yang sedang terjadi, nama kasir juga akan tertampil sesuai dengan pengguna yang login. Pada bagian input data barang, kasir dapat men-scan barcode barang tertentu dan kemudian harga akan muncul

secara otomatis atau dengan memasukkan kode barang ke dalam *textbox* yang tersedia atau dengan mengklik tombol cari barang. Saat tombol cari barang diklik akan muncul *pop-up* yang menampilkan daftar barang pada gudang Bulog Mart tersebut. Klik pada *row* barang tertentu jika ingin memilihnya. Masukkan jumlah barang yang akan dibeli, harga akan muncul secara otomatis, dan diskon akan dimasukkan manual dengan potongan harga tertentu. Jika sudah, klik tambah barang untuk menambah barang dalam daftar barang. Barang yang ditambahkan akan tertampil pada tabel daftar barang. Jika ingin mengubah jumlah barang maka klik gambar pensil yang ada di dalam tabel kemudian ubah jumlah barang yang akan dibeli. Setelah semua barang sudah diinputkan, setelah itu total harga yang dibeli akan tertera dan kasir harus memasukkan input pembayaran, pada *textbox* kembalian akan tertampil hasil pengurangan dari pembayaran dengan total harga. Yang terakhir untuk menyimpan data transaksi jual, kasir harus menekan tombol selesaikan transaksi, maka akan muncul nota pembayaran yang telah dilakukan, dan data transaksi akan tersimpan di *database*.

Program Studi Teknik Informatika	DPPL-SISIP	147/151
Dokumen ini adalah 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		

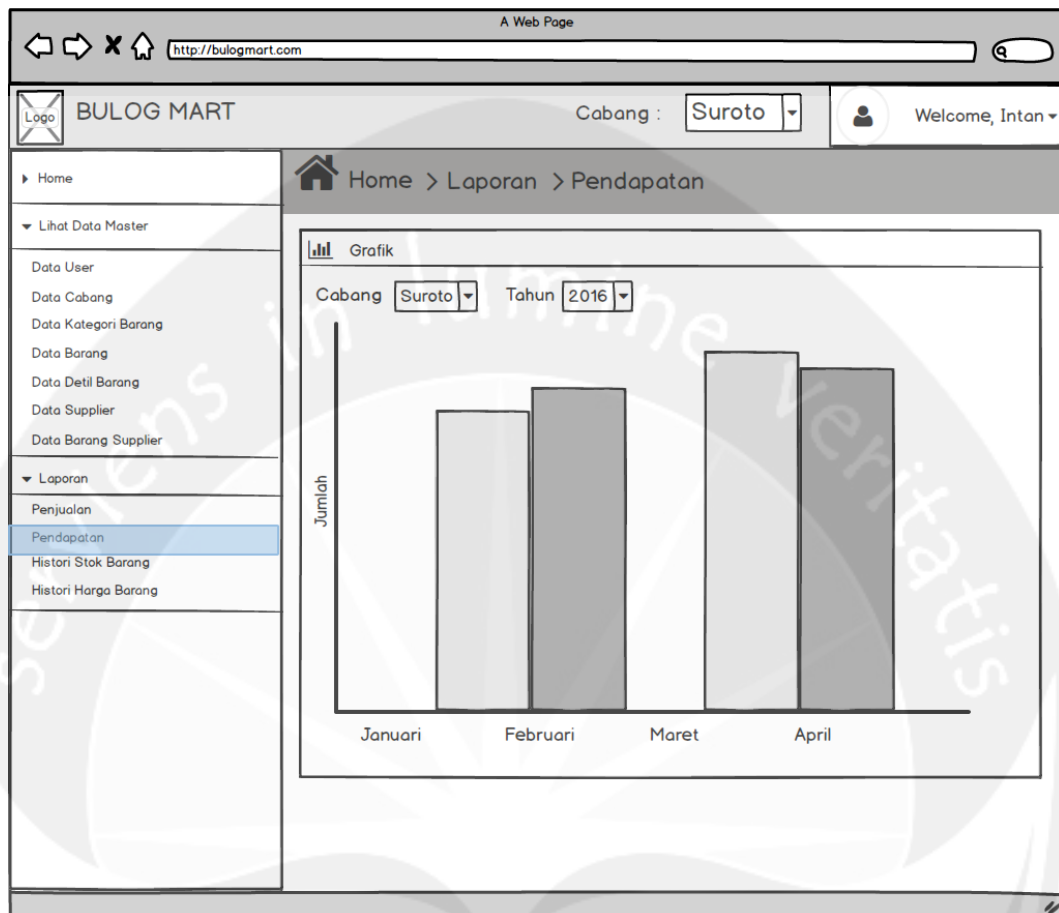
4.18 Antarmuka Laporan Penjualan



Gambar 4.34 Rancangan Antarmuka Transaksi Penjualan

Gambar di atas menunjukkan rancangan antarmuka untuk laporan penjualan. Laporan penjualan hanya dapat diakses oleh KASI PPU (Kepala Seksi Perencanaan dan Pengembangan Usaha). Laporan penjualan berada dalam menu Laporan, ada satu macam tampilan untuk menampilkan laporan penjualan yaitu dalam bentuk grafik. Laporan penjualan ditampilkan berdasarkan cabang dari Bulog Mart DIY, yaitu Suroto, Wates, dan Purwomartani, dan tahun berapa.

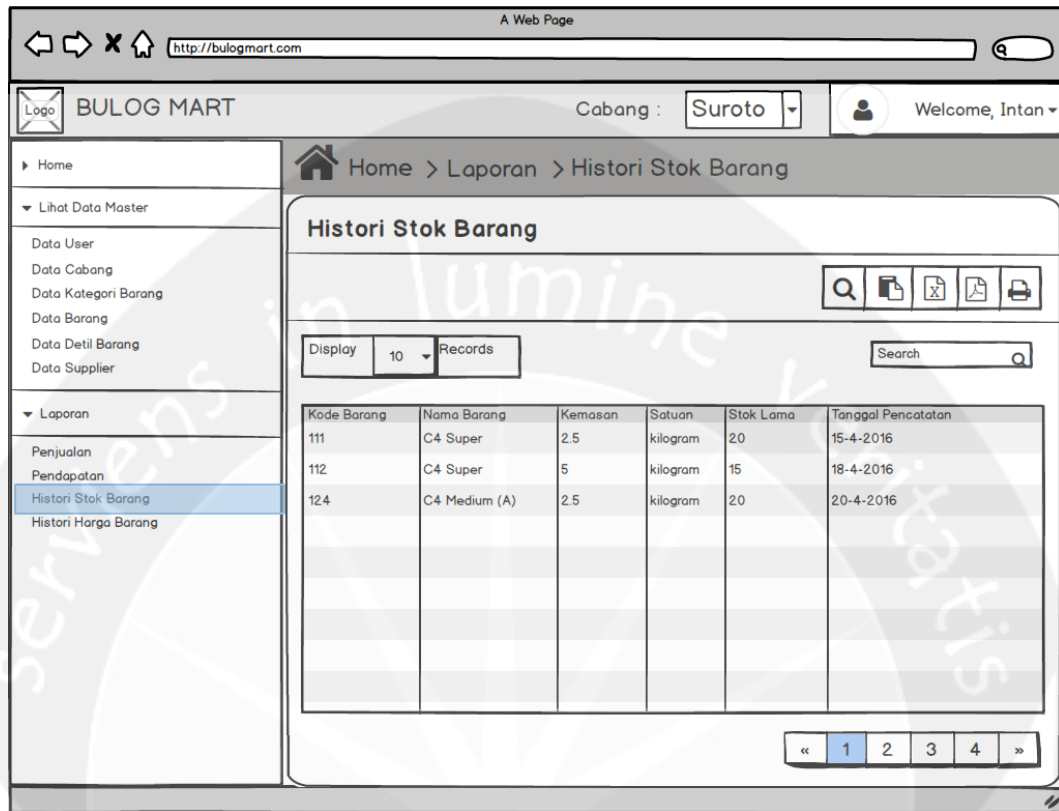
4.19 Antarmuka Laporan Pendapatan



Gambar 4.35 Rancangan Antarmuka Transaksi Penjualan

Gambar di atas menunjukkan rancangan antarmuka untuk laporan pendapatan. Laporan pendapatan hanya dapat diakses oleh KASI PPU (Kepala Seksi Perencanaan dan Pengembangan Usaha). Laporan pendapatan berada dalam menu Laporan, ada satu macam tampilan untuk menampilkan laporan pendapatan yaitu dalam bentuk grafik. Laporan pendapatan ditampilkan berdasarkan cabang dari Bulog Mart DIY, yaitu Suroto, Wates, dan Purwomartani, dan tahun berapa.

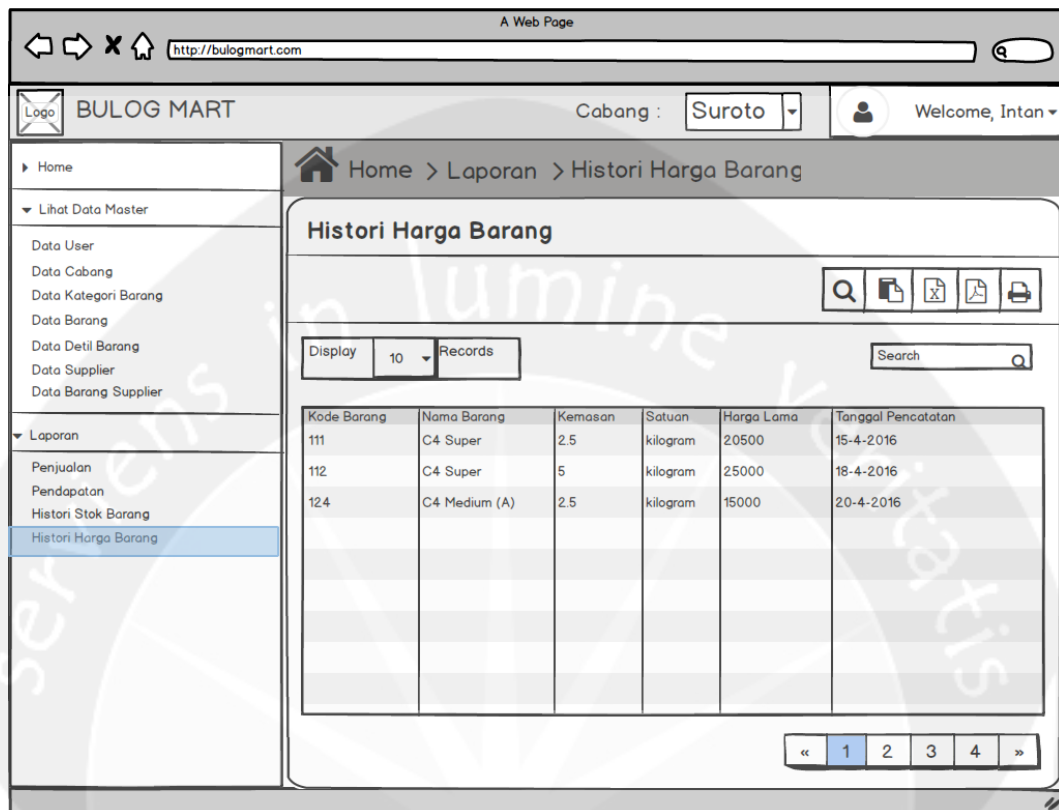
4.20 Antarmuka Laporan Histori Stok Barang



Gambar 4.36 Rancangan Antarmuka Laporan Histori Stok Barang

Gambar di atas menunjukkan rancangan antarmuka untuk laporan histori stok barang. Laporan ini hanya bisa diakses oleh KASI PPU (Kepala Seksi Perencanaan dan Pengembangan Usaha). Submenu histori stok barang akan tersedia pada saat pengguna mengakses menu laporan. Pada antarmuka histori stok barang ditampilkan informasi stok barang yang lama. Fungsi dari laporan ini adalah mencatat stok lama pada barang tertentu saat terjadi perubahan jumlah stok.

4.21 Antarmuka Laporan Histori Harga Barang



Gambar 4.37 Rancangan Antarmuka Laporan Histori Harga Barang

Gambar di atas menunjukkan rancangan antarmuka untuk laporan histori harga barang. Laporan ini hanya bisa diakses oleh KASI PPU (Kepala Seksi Perencanaan dan Pengembangan Usaha). Submenu histori harga barang akan tersedia pada saat pengguna mengakses menu laporan. Pada antarmuka histori harga barang ditampilkan informasi harga barang yang lama. Fungsi dari laporan ini adalah mencatat harga lama pada barang tertentu saat terjadi perubahan harga.