BAB 4
IMPLEMENTASI DAN PENGUJIAN

Tahapan implementasi dan pengujian adalah tahap dimana perancangan diimplementasikan pada bahasa pemrograman yang digunakan. Setelah diimplementasikan kemudian dilakukan pengujian. Tahap pengujian dilakukan untuk melihat kekurangan-kekurangan yang mungkin terjadi pada aplikasi yang baru dibangun.

4.1. Implementasi

Implementasi dilakukan untuk menerapkan perancangan yang telah dilakukan pada tahap sebelumnya, sehingga pengguna pada aplikasi *supply chain management* ini dapat memberi masukan kepada pembangun aplikasi demi terbentuknya aplikasi yang sesuai dengan kebutuhan pengguna

4.1.1. Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam mengimplementasikan aplikasi ini adalah sebagai berikut:

Tabel 4.1 Spesifikasi Perangkat Keras

Perangkat Keras	Yang digunakan	Yang dibutuhkan
Processor	Intel i5 2,8 GHz	Intel Pentium 4 1,5 GHz
RAM (*Memory*)	4 GB	1 GB
Harddisk	500 Gb	512 GB
Monitor	1024x768	1024x600 (32 bit, 60 Hz)

4.1.2. Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam mengimplementasikan aplikasi ini adalah sebagai berikut:

Tabel 4.2 Spesifikasi Perangkat Lunak

Perangkat Keras	Yang digunakan	Yang dibutuhkan
Sistem Operasi	Window 8	Window XP
PDF Reader	Adobe reader 9	Adobe reader 9
DBMS	MySQL	MySQL

Browser	Google Chrome, Modzila Firefox, Opera, dan Internet Explorer	Google Chrome, Modzila Firefox, Opera
Code Editor	Adobe Dreamweaver CS3	Adobe Dreamweaver CS3

4.1.3. Implementasi Basis Data

Implementasi basis data merupakan penggambaran dalam pembuatan *database* pada aplikasi yang dibangun, implementasi basis data digambarkan dalam bahasa SQL (*Structured Query Language*) berikut ini:

Tabel 4.3 Implementasi basis data

No	Nama tabel	Hasil pembangkitan
1	bagian	CREATE TABLE IF NOT EXISTS `bagian` (`kode_bagian` varchar(5) NOT NULL, `nama_bagian` varchar(200) NOT NULL, `lokasi` varchar(500) NOT NULL, `ruang` varchar(10) NOT NULL, PRIMARY KEY (kode_barang))ENGINE=InnoDB DEFAULT CHARSET=latin1;
2	bahan_baku	CREATE TABLE IF NOT EXISTS `bahan_baku` (`kode_bb` varchar(20) NOT NULL, `nama_bahan_baku` varchar(500) NOT NULL, PRIMARY KEY (kode_bb))ENGINE=InnoDB DEFAULT CHARSET=latin1;
3	detail_pembelian	CREATE TABLE IF NOT EXISTS `detail_pembelian` (`purcash_order` varchar(16) NOT NULL, `kode_perencanaan` varchar(10) NOT NULL, `harga` int(20) NOT NULL, `leadtime` int(3) NOT NULL, `status` varchar(20) NOT NULL,

No	Nama tabel	Hasil pembangkitan
		FOREIGN KEY (purcash_order) REFERENCES pembelian(purcash_order) ON UPDATE CASCADE ON DELETE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1;
4	detail_pemesanan	CREATE TABLE IF NOT EXISTS `detail_pemesanan` (`material_requitetion` varchar(16) NOT NULL, `kode_bb` varchar(10) NOT NULL, `jumlah` int(10) NOT NULL, `status` varchar(20) NOT NULL, FOREIGN KEY (material_requitetion) REFERENCES pemesanan(material_requitetion) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
5	detail_penjualan	CREATE TABLE IF NOT EXISTS `detail_penjualan` (`delivery_order` varchar(16) NOT NULL, `kode_produk` varchar(10) NOT NULL, `jumlah` int(10) NOT NULL, FOREIGN KEY (delivery_order) REFERENCES penjualan(delivery_order) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (kode_produk) REFERENCES produk(kode_produk) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
6	detail_perencanaan	CREATE TABLE IF NOT EXISTS `detail_perencanaan` (`kode_perencanaan` varchar(16) NOT NULL, `purcash_requitetion` varchar(16) NOT NULL, `kode_bb` varchar(10) NOT NULL, `jumlah` int(10) NOT NULL,

No	Nama tabel	Hasil pembangkitan
		`status` varchar(20) NOT NULL, FOREIGN KEY (purcash_requiteion) REFERENCES perencanaan(purcash_requitetion) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (kode_bb) REFERENCES bahan_baku(kode_bb) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
7	gudang_lini_3	CREATE TABLE IF NOT EXISTS `gudang_lini_3` (`kode_gudang` varchar(50) NOT NULL, `nama_gudang` varchar(50) NOT NULL, `username` varchar(50) NOT NULL, `userpass` varchar(50) NOT NULL, `email` varchar(50) NOT NULL, `pimpinan` varchar(100) NOT NULL, `alamat` varchar(200) NOT NULL, `telepon` varchar(20) NOT NULL, `kapasitas_tampung` int(20) NOT NULL, PRIMARY KEY (kode_gudang))ENGINE=InnoDB DEFAULT CHARSET=latin1;
8	hapus_data	CREATE TABLE IF NOT EXISTS `hapus_data` (`kode_hapus` varchar(10) NOT NULL, `kode_data` varchar(20) NOT NULL, `tanggal_hapus` varchar(8) NOT NULL, `pelaku` varchar(100) NOT NULL, `alasan` varchar(500) NOT NULL, `nip` varchar(20) NOT NULL, PRIMARY KEY (kode_hapus))ENGINE=InnoDB DEFAULT CHARSET=latin1;
9	jabatan	CREATE TABLE IF NOT EXISTS `jabatan`

No	Nama tabel	Hasil pembangkitan
		(`kode_jabatan` varchar(5) NOT NULL, `nama_jabatan` varchar(20) NOT NULL, PRIMARY KEY (kode_jabatan))ENGINE=InnoDB DEFAULT CHARSET=latin1;
10	kebutuhan	CREATE TABLE IF NOT EXISTS `kebutuhan` (`kode_produk` varchar(20) NOT NULL, `kode_bb` varchar(20) NOT NULL, `kadar` int(3) NOT NULL, FOREIGN KEY (kode_produk) REFERENCES produk(kode_produk) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (kode_bb) REFERENCES bahan_baku(kode_bb) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
11	kendaraan	CREATE TABLE IF NOT EXISTS `kendaraan` (`no_polisi` varchar(15) NOT NULL, `jenis_angkutan` varchar(20) NOT NULL, `kapasitas` varchar(5) NOT NULL, `nama_supir` varchar(100) NOT NULL, PRIMARY KEY (no_polisi))ENGINE=InnoDB DEFAULT CHARSET=latin1;
12	pegawai	CREATE TABLE IF NOT EXISTS `pegawai` (`nip` varchar(20) NOT NULL, `nama_pegawai` varchar(100) NOT NULL, `alamat` varchar(200) NOT NULL, `telepon` varchar(20) NOT NULL, `username` varchar(50) NOT NULL,

No	Nama tabel	Hasil pembangkitan
		`userpass` varchar(50) NOT NULL, `email` varchar(50) NOT NULL, `kode_jabatan` varchar(5) NOT NULL, `kode_bagian` varchar(5) NOT NULL, PRIMARY KEY (nip) , FOREIGN KEY (kode_jabatan) REFERENCES jabatan(kode_jabatan) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (kode_bagian) REFERENCES bagian(kode_bagian) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
13	pembelian	CREATE TABLE IF NOT EXISTS `pembelian` (`purcash_order` varchar(16) NOT NULL, `kode_supplier` varchar(10) NOT NULL, `tanggal_pembelian` varchar(8) NOT NULL, `nip` varchar(20) NOT NULL, PRIMARY KEY (purcash_order) , FOREIGN KEY (kode_supplier) REFERENCES supplier(kode_supplier) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (nip) REFERENCES pegawai(nip) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
14	Pemesanan	CREATE TABLE IF NOT EXISTS `pemesanan` (`material_requitetion` varchar(16) NOT NULL, `tanggal_pemesanan` varchar(10) NOT NULL, `status` varchar(20) NOT NULL, `nip` varchar(20) NOT NULL,

No	Nama tabel	Hasil pembangkitan
		PRIMARY KEY (material_requitetion) , FOREIGN KEY (nip) REFERENCES pegawai(nip) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
15	Pengangkutan	CREATE TABLE IF NOT EXISTS `pengangkutan` (`no_polisi` varchar(15) NOT NULL, `kode_penjualan` varchar(10) NOT NULL, `tanggal_pengangkutan` varchar(8) NOT NULL, `kapasitas` int(5) NOT NULL, `status` varchar(20) NOT NULL, FOREIGN KEY (no_polisi) REFERENCES kendaraan(no_polisi) ON UPDATE CASCADE ON DELETE CASCADE, , FOREIGN KEY (kode_penjualan) REFERENCES detail_penjualan(kode_penjualan) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
16	Pengiriman	CREATE TABLE IF NOT EXISTS `pengiriman` (`kode_tujuan` varchar(5) NOT NULL, `nama_perusahaan` varchar(100) NOT NULL, `nama_pimpinan` varchar(100) NOT NULL, `alamat` varchar(200) NOT NULL, `telepon` varchar(20) NOT NULL)ENGINE=InnoDB DEFAULT CHARSET=latin1;
17	Penjualan	CREATE TABLE IF NOT EXISTS `penjualan` (`delivery_order` varchar(16) NOT NULL, `kode_gudang` varchar(5) NOT NULL, `tanggal_penjualan` varchar(8) NOT NULL,

No	Nama tabel	Hasil pembangkitan
		`nip` varchar(20) NOT NULL, `status` varchar(20) NOT NULL, PRIMARY KEY (delivery_order) , FOREIGN KEY (kode_gudang) REFERENCES gudang_lini_3(kode_gudang) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1
18	Perencanaan	CREATE TABLE IF NOT EXISTS `perencanaan` (`purcash_requitetion` varchar(16) NOT NULL, `tanggal_perencanaan` varchar(8) NOT NULL, `nip` varchar(20) NOT NULL, `status` varchar(20) NOT NULL, PRIMARY KEY (purcash_requitetion) , FOREIGN KEY (nip) REFERENCES pegawai(nip) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
19	produk	CREATE TABLE IF NOT EXISTS `produk` (`kode_produk` varchar(20) NOT NULL, `nama_produk` varchar(20) NOT NULL, `kapasitas` int(8) NOT NULL, PRIMARY KEY (kode_produk))ENGINE=InnoDB DEFAULT CHARSET=latin1;
20	Stok_bb	CREATE TABLE IF NOT EXISTS `stok_bb` (`kode_bb` varchar(10) NOT NULL, `nama_bb` varchar(100) NOT NULL, `persediaan` int(10) NOT NULL)ENGINE=InnoDB DEFAULT CHARSET=latin1;
21	Stok_produk	CREATE TABLE IF NOT EXISTS `stok_produk`

No	Nama tabel	Hasil pembangkitan
		(`kode_produk` varchar(20) NOT NULL, `tanggal_produksi` varchar(10) NOT NULL, `jumlah` int(10) NOT NULL, `status` varchar(20) NOT NULL, `nip_pelaku` varchar(20) NOT NULL, `verifikasi` varchar(20) NOT NULL, `nip_pengizin` varchar(20) NOT NULL)ENGINE=InnoDB DEFAULT CHARSET=latin1;
22	Supplier	CREATE TABLE IF NOT EXISTS `supplier` (`kode_supplier` varchar(10) NOT NULL, `nama_perusahaan` varchar(100) NOT NULL, `nama_pimpinan` varchar(100) NOT NULL, `alamat` varchar(200) NOT NULL, `telepon` varchar(20) NOT NULL, `nip` varchar(20) NOT NULL, PRIMARY KEY (kode_supplier) , FOREIGN KEY (nip) REFERENCES pegawai(nip) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;
23	target_produksi	CREATE TABLE IF NOT EXISTS `target_produksi` (`kode_produk` varchar(20) NOT NULL, `bulan` varchar(2) NOT NULL, `tahun` varchar(4) NOT NULL, `target` int(20) NOT NULL, FOREIGN KEY (kode_produk) REFERENCES produk(kode_produk) ON UPDATE CASCADE ON DELETE CASCADE)ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.1.4. Impelemntasi Antar Muka

Implementasi antar muka pada aplikasi *supply chain management* ini adalah sebagai berikut:

Tabel 4.4 Implementasi Antar Muka PSDM

Menu	Deskripsi	Nama File
Form login	Tampilan login pengguna	Login.php
Form Menu Bagian	Tampilan untuk mengelola data bagian	bagian.php
Form Menu Jabatan	Tampilan untuk mengelola data jabatan	jabatan.php
Form Menu Pegawai	Tampilan untuk mengolah data pegawai	pegawai.php

Tabel 4.5 Implementasi Antar Muka Kompartemen Produksi

Menu	Deskripsi	Nama File
Form login	Tampilan login pengguna	Login.php
Form Menu Bahan Baku	Tampilan untuk mengelola data bahan baku	bb.php
Form Menu Produk	Tampilan untuk mengelola data produk	produk.php
Form Menu Kebutuhan	Tampilan untuk mengolah data kebutuhan	kebutuhan.php

Tabel 4.6 Implementasi Antar Muka Produksi

Menu	Deskripsi	Nama File
Form login	Tampilan login pengguna	Login.php
Form Menu Jadwal Produksi	Tampilan untuk melihat data jadwal produksi	jadwal.php

Form Menu Pemesanan	Tampilan untuk mengelola data pemesanan	pemesanan.php
Form Menu Penyetokan	Tampilan untuk mengolah data penyetokan	penyetokan.php

Tabel 4.7 Implementasi Antar Muka Penjualan PSO

Menu	Deskripsi	Nama File
Form login	Tampilan login pengguna	Login.php
Form Menu Target Produksi	Tampilan untuk mengelola data target produksi	target.php
Form Menu Gudang Lini 3	Tampilan untuk mengelola data gudang lini 3	gudang.php
Form Menu Penjualan	Tampilan untuk mengolah data penjualan	penjualan.php

Tabel 4.8 Implementasi Antar Muka Sarana Penjualan

Menu	Deskripsi	Nama File
Form login	Tampilan login pengguna	Login.php
Form Menu Kendaraan	Tampilan untuk mengelola data kendaraan	kendaraan.php
Form Menu Pengangkutan	Tampilan untuk mengelola data pengangkutan	pengangkutan.php

4.2. Pengujian

Pengujian sistem dilakukan untuk menguji sistem dimana pada tahap ini dapat diketahui kesalahan - kesalahan sehingga dapat dilakukan perbaikan dan pengembangan sistem maka dapat diketahui kesesuaian sistem dengan kebutuhannya.

Pengujian yang digunakan untuk menguji sistem yang baru adalah dengan menggunakan metode pengujian *blackbox*. Pada pengujian *blackbox* lebih fokus pada persyaratan fungsional perangkat lunak dan pengujian ini berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Kesalahan *interface*
2. Fungsi - fungsi yang tidak sesuai dengan prosedur
3. Kesalahan dalam struktur data
4. Kesalahan kinerja sistem

4.2.1. Rencana Pengujian Blackbox

Pengujian blackbox yang akan dilakukan pada pengguna sistem informasi yang dibangun adalah sebagai berikut:

Tabel 4.9 Rencana Pengujian

Kasus Uji	Detail Pengujian	Jenis Pengujian
Login	Verifikasi data login	Black Box
Pengolahan data bagian	Tambah bagian	Black Box
	Ubah bagian	Black Box
	Hapus bagian	Black Box
Pengolahan data jabatan	Tambah jabatan	Black Box
	Ubah jabatan	Black Box
	Hapus jabatan	Black Box
Pengolahan data pegawai	Tambah pegawai	Black Box
	Ubah pegawai	Black Box
	Hapus pegawai	Black Box
Pengolahan data bahan baku	Tambah bahan baku	Black Box
	Ubah bahan baku	Black Box
	Hapus bahan baku	Black Box
Pengolahan data produk	Tambah produk	Black Box
	Ubah produk	Black Box
	Hapus produk	Black Box
Pengolahan data target produksi	Tambah target produksi	Black Box
	Ubah target produksi	Black Box
	Tambah data gudang	Black Box

Pengolahan data gudang lini 3	Ubah data gudang	Black Box
	Hapus data gudang	Black Box
Pengolahan data kendaraan	Tambah data kendaraan	Black Box
	Ubah data kendaraan	Black Box
	Hapus data kendaraan	Black Box

4.2.2. Kasus dan Hasil Pengujian Blackbox

Berdasarkan pada rencana pengujian blackbox maka dilakukan pengujian dan didapatkan hasil dari pengujian blackbox sebagai berikut:

4.2.2.1. Pengujian Login

Berikut dapat dilihat pengujian dari *login*, pemilik dan kepala produksi dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.10 Pengujian Login

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username Contoh masukan : Username: 03	Data username yang dimasukkan akan tampil pada masing-masing isian	Data username yang dimasukan tampil	[√] Diterima [ ] Ditolak
password contoh masukan: password : 03	Data password yang dimasukkan akan tampil pada masing-masing isian	Data password yang dimasukan tampil	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Username Contoh masukan: Username: Coba-Coba	Tampilkan pesan kesalahan "Maaf Username Anda Tidak Terdaftar"	Dapat menampilkan pesan kesalahan "Maaf Username Anda Tidak Terdaftar"	[√] Diterima [ ] Ditolak
password contoh masukan: password : Coba1	Tampilkan pesan kesalahan "Maaf Password Anda Salah"	Dapat menampilkan pesan kesalahan "Maaf Password Anda Salah"	[√] Diterima [ ] Ditolak

4.2.2.2. Tambah Bagian

Berikut dapat dilihat pengujian dari tambah bagian, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.11 Pengujian Tambah Data Bagian

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Bagian: 01 Nama Bagian: Pengelolaan Sumber Daya Manusia Lokasi: Gedung Administrasi Ruang: 211	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan "Tambah data berhasil"	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses penambahan bagian	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Kode Bagian: Nama Bagian: Lokasi: Ruang:	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

4.2.2.3. Ubah Bagian

Berikut dapat dilihat pengujian dari ubah bagian, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.12 Pengujian Ubah Data Bagian

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Bagian: 01 Nama Bagian: Pengelolaan Sumber Daya Manusia Lokasi: Gedung Administrasi Ruang: 212	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan “Ubah data berhasil”	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses simpan data dan menampilkan pesan “Ubah data berhasil”	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Kode Bagian: Nama Bagian: Lokasi: Ruang:	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

4.2.2.4. Tambah Jabatan

Berikut dapat dilihat pengujian dari tambah jabatan, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.13 Pengujian Tambah Data Jabatan

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Jabatan: 01 Nama Jabatan: Staff	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan “Tambah data berhasil”	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses penambahan bagian	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Jabatan: Nama Jabatan:	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

4.2.2.5. Ubah Jabatan

Berikut dapat dilihat pengujian dari ubah jabatan, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.14 Pengujian Ubah Data Jabatan

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Jabatan: 01 Nama Jabatan: Supervisor	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan "Ubah data berhasil"	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses simpan data dan menampilkan pesan "Ubah data berhasil"	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Jabatan: Nama Jabatan:	Tampilkan pesan kesalahan "data belum diisi"	Dapat menampilkan pesan kesalahan "data belum diisi"	[√] Diterima [ ] Ditolak

4.2.2.6. Tambah Pegawai

Berikut dapat dilihat pengujian dari tambah pegawai, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.15 Pengujian Tambah Data Pegawai

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
NIP : 1020042007001 Nama Pegawai : Ahmad Saeful Alamat Jln. Jenderal Sudirman No. 22 Cikampek Telepon : 0264 2303 001 Username : Ahmad Userpass : Ahmad123 Email : ahmad.saeful@yahoo.co.id Kode Jabatan : 01 Kode Bagian : 08	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan “Tambah data berhasil”	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses penambahan bagian	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
NIP : Nama Pegawai : Alamat Telepon :	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

Username : Userpass : Email : Kode Jabatan : Kode Bagian :			

4.2.2.7. Ubah Pegawai

Berikut dapat dilihat pengujian dari ubah pegawai, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.16 Pengujian Ubah Data Pegawai

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
NIP : 1020042007001 Nama Pegawai : Ahmad Saeful M Alamat Jln. Jenderal Sudirman No. 22 Cikampek Telepon : 0264 2303 011 Username : Ahmad Userpass : Ahmad123	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan “Ubah data berhasil”	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses simpan data dan menampilkan pesan “Ubah data berhasil”	[√] Diterima [ ] Ditolak

Email : ahmad.saeful@yahoo.co.id Kode Jabatan : 01 Kode Bagian : 09			
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
NIP : Nama Pegawai : Alamat Telepon : Username : Userpass : Email : Kode Jabatan : Kode Bagian :	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

4.2.2.8. Tambah Bahan Baku

Berikut dapat dilihat pengujian dari tambah bahan baku, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.17 Pengujian Tambah Data Bahan Baku

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Bahan Baku : 01 Nama Bahan Baku : Kcl	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan "Tambah data berhasil"	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses penambahan bagian	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Bahan Baku : Nama Bahan Baku :	Tampilkan pesan kesalahan "data belum diisi"	Dapat menampilkan pesan kesalahan "data belum diisi"	[√] Diterima [ ] Ditolak

4.2.2.9. Ubah Bahan Baku

Berikut dapat dilihat pengujian dari ubah bahan baku, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.18 Pengujian Ubah Data Bahan Baku

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Bahan Baku : 01 Nama Bahan Baku :	Jika form terisi atau benar dalam pengisian, maka	Semua form terisi atau benar dalam pengisian, lalu	[√] Diterima [ ] Ditolak

DAP	akan dapat melanjutkan proses simpan data dan menampilkan pesan "Ubah data berhasil"	dapat melanjutkan proses simpan data dan menampilkan pesan "Ubah data berhasil"	
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Bahan Baku : Nama Bahan Baku :	Tampilkan pesan kesalahan "data belum diisi"	Dapat menampilkan pesan kesalahan "data belum diisi"	[√] Diterima [ ] Ditolak

4.2.2.10. Tambah Produk

Berikut dapat dilihat pengujian dari tambah produk, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.19 Pengujian Tambah Data Produk

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Produk: 01 Nama Produk: Pupuk NPK 15:15:15 Kapasitas: 50 Kg	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan "Tambah data berhasil"	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses penambahan bagian	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			

Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Kode Produk: Nama Produk: Kapasitas:	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

4.2.2.11. Ubah Produk

Berikut dapat dilihat pengujian dari ubah produk, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.20 Pengujian Ubah Data Produk

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Produk: 01 Nama Produk: Pupuk NPK 15:15:15 Kapasitas: 5 Kg	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan “Ubah data berhasil”	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses simpan data dan menampilkan pesan “Ubah data berhasil”	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Produk: Nama Produk:	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

Kapasitas:			

4.2.2.12. Tambah Target Produksi

Berikut dapat dilihat pengujian dari tambah target produksi, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.21 Pengujian Tambah Data Target Produksi

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Bulan: 02 Tahun: 2016 Target Produksi: 1600	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan "Tambah data berhasil"	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses penambahan bagian	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Bulan: Tahun: Target Produksi:	Tampilkan pesan kesalahan "data belum diisi"	Dapat menampilkan pesan kesalahan "data belum diisi"	[√] Diterima [ ] Ditolak

4.2.2.13. Ubah Target Produksi

Berikut dapat dilihat pengujian dari ubah target produksi, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.22 Pengujian Ubah Data Target Produksi

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Target Produksi: 1800	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan "Ubah data berhasil"	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses simpan data dan menampilkan pesan "Ubah data berhasil"	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Target Produksi:	Tampilkan pesan kesalahan "data belum diisi"	Dapat menampilkan pesan kesalahan "data belum diisi"	[√] Diterima [ ] Ditolak

4.2.2.14. Tambah Gudang Lini 3

Berikut dapat dilihat pengujian dari tambah Gudang Lini 3, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.23 Pengujian Tambah Data Gudang Lini 3

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Kode Gudang: G01 Nama Gudang: Gudang Lini 3 Cab. Tasik Email: gudang_tasik@pupuk-kujang.co.id Pimpinan: Doni Subandrio Alamat: Jln. Maribaya No.3 Tasikmalaya Telepon: 0265 2349 887 Kapasitas Tampungan: 70	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan "Tambah data berhasil"	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses penambahan bagian	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Gudang: Nama Gudang: Email: Pimpinan: Alamat: Telepon:	Tampilkan pesan kesalahan "data belum diisi"	Dapat menampilkan pesan kesalahan "data belum diisi"	[√] Diterima [ ] Ditolak

Kapasitas Tampungan:			

4.2.2.15. Ubah Gudang Lini 3

Berikut dapat dilihat pengujian dari ubah gudang lini 3, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.24 Pengujian Ubah Data Gudang Lini 3

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Gudang: G01 Nama Gudang: Gudang Lini 3 Cab. Tasik Email: gudang_tasik@pupuk-kujang.co.id Pimpinan: Doni Subandrio Alamat: Jln. Maribaya No.3 Tasikmalaya Telepon: 0265 2349 887 Kapasitas Tampungan: 120	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan “Ubah data berhasil”	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses simpan data dan menampilkan pesan “Ubah data berhasil”	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Kode Gudang: Nama Gudang: Email: Pimpinan: Alamat: Telepon: Kapasitas Tampungan:	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

4.2.2.16. Tambah Kendaraan

Berikut dapat dilihat pengujian dari tambah kendaraan, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.25 Pengujian Tambah Data Kendaraan

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nomor Polisi: T 1111 PK Jenis Angkutan: Truck Kapasitas: 25 TON Nama Sopir: Zainal M.	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan “Tambah data berhasil”	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses penambahan bagian	[√] Diterima [ ] Ditolak

Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nomor Polisi: Jenis Angkutan: Kapasitas: Nama Sopir:	Tampilkan pesan kesalahan "data belum diisi"	Dapat menampilkan pesan kesalahan "data belum diisi"	[√] Diterima [ ] Ditolak

4.2.2.17. Ubah Kendaraan

Berikut dapat dilihat pengujian dari ubah kendaraan, dengan kasus dan hasil uji kemudian diamati dan menghasilkan kesimpulan dari pengujian tersebut:

Tabel 4.26 Pengujian Ubah Data Kendaraan

Kasus dan Hasil uji (Valid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nomor Polisi: T 1111 PK Jenis Angkutan: Truck Kapasitas: 25 TON Nama Sopir: Faizal M.	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses simpan data dan menampilkan pesan "Ubah data berhasil"	Semua form terisi atau benar dalam pengisian, lalu dapat melanjutkan proses simpan data dan menampilkan pesan "Ubah data berhasil"	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Invalid Data)			
Data Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Nomor Polisi: Jenis Angkutan: Kapasitas: Nama Sopir:	Tampilkan pesan kesalahan “data belum diisi”	Dapat menampilkan pesan kesalahan “data belum diisi”	[√] Diterima [ ] Ditolak

4.2.3. Kesimpulan Pengujian Blackbox

Berdasarkan hasil pengujian pada kasus uji yang telah dilakukan maka dapat disimpulkan bahwa secara umum sistem informasi supply chain management di PT Pupuk Kujang dapat berjalan sesuai dengan yang diharapkan.

4.2.4. Pengujian Beta

Setelah melakukan pengujian blackbox maka diperlukan juga pengujian beta dimana pengujian beta adalah pengujian yang dilakukan secara langsung ke lapangan dan hasilnya bersifat objektif.

Pengujian beta dilakukan dengan cara *interview* atau wawancara dengan mengunakan teknik wawancara terstruktur mengenai kepuasan pengguna, dengan menggunakan teknik jawaban terbuka yaitu peneliti memberikan jawaban sepenuhnya kepada orang yang menggunakan sistem.

Wawancara hanya dilakukan 2 user saja yang akan di wawancara tentang kekurangan dan kelebihan sistem yang dibangun yakni perencanaan pengendalian material dan penjualan *pso*.

Adapun pertanyaan pertanyaan yang diajukan dalam proses wawancara pengujian beta sistem informasi supply chain management di PT Pupuk Kujang secara terperinci dapat dilihat pada tabel berikut:

Tabel 4.27 Pertanyaan pengujian beta

Level Pengguna	Pertanyaan

Perencanaan Pengendalian Material (PPM)	1. Apakah sistem informasi *supply chain management* ini mudah digunakan? Jawaban: Menurut bagian perencanaan pengendalian material (ppm) bahwa sistem informasi *supply chain management* yang dibangun sederhana dan mudah untuk digunakan. 2. Apakah tampilan antar muka sistem informasi *supply chain management* ini terlihat menarik? Jawaban: Menurut bagian perencanaan pengendalian material (ppm) bahwa sistem informasi *supply chain management* yang dibangun tampilan antarmuka sederhana sangat mirip dengan aplikasi sebelumnya *(IFS)* namun sistem yang dibangun ini lebih interaktif dan fungsionalis. 3. Apakah sistem informasi *supply chain management* ini interaktif? Jawaban: Menurut bagian perencanaan pengendalian material (ppm) bahwa sistem informasi *supply chain management* yang dibangun lebih interaktif dari pada aplikasi sebelumnya *(IFS).* 4. Apakah sistem informasi *supply chain management* ini memberikan kemudahan pada bagian perencanaan pengendalian material? Jawaban: Menurut bagian perencanaan pengendalian material (ppm) bahwa sistem informasi *supply chain management* yang dibangun sangat memberikan kemudahan untuk bagian perencanaan pengendalian material (ppm).

	5. Menurut anda, apakah sistem informasi *supply chain management* ini menggunakan bahasa yang mudah untuk dimengerti? Jawaban: Menurut bagian perencanaan pengendalian material (ppm) bahwa sistem informasi *supply chain management* yang dibangun dikarenakan sistem yang dibangun sederhana dan berbahasa indonesia sehingga sangat mudah untuk dimengerti. 6. Apakah sistem informasi *supply chain management* ini memberikan kemudahan dalam meramalkan pembelian bahan baku bagi perencanaan pengendalian material? Jawaban: Menurut bagian perencanaan pengendalian material (ppm) bahwa sistem informasi *supply chain management* yang dibangun sangat membantu perencanaan pengendalian material (ppm) dalam meramalkan pembelian bahan baku. 7. Apakah sistem informasi *supply chain management* ini memberikan kemudahan dalam memantau stok bahan baku pada gudang material oleh perencanaan pengendalian material? Jawaban: Menurut bagian perencanaan pengendalian material (ppm) bahwa sistem informasi *supply chain management* yang dibangun sangat membantu perencanaan pengendalian material (ppm) dalam memantau stok bahan baku pada gudang material. 8. Apakah sistem informasi *supply chain management* yang dibangun mengenai perhitungan perencanaan persediaan bahan baku agar tidak kekurangan maupun kelebihan bahan baku sudah sesuai dengan yang dibutuhkan?

	Jawaban: Menurut bagian perencanaan pengendalian material (ppm) bahwa sistem informasi *supply chain management* yang dibangun mengenai perhitungan perencanaan persediaan bahan baku pada periode sebelumnya yang telah sesuai sistem yang dibangun, maka dapat dipastikan sistem yang dibangun telah cukup akurat dan sesuai dengan kebutuhan perencanaan pengendalian material (ppm).
Penjualan PSO	1. Apakah sistem informasi *supply chain management* ini mudah digunakan? Jawaban: Menurut bagian penjualan *pso* bahwa sistem informasi *supply chain management* yang dibangun sangat sederhana sehingga mudah untuk digunakan. 2. Apakah tampilan antar muka sistem informasi *supply chain management* ini terlihat menarik? Jawaban: Menurut bagian penjualan *pso* bahwa sistem informasi *supply chain management* yang dibangun tampilan antarmuka sistem ini sederhana dan perlu dibuat lebih menarik lagi. 3. Apakah sistem informasi *supply chain management* ini interaktif? Jawaban: Menurut bagian penjualan *pso* bahwa sistem informasi *supply chain management* yang dibangun cukup interaktif. 4. Apakah sistem informasi *supply chain management* ini memberikan kemudahan pada bagian penjualan PSO? Jawaban:

	Menurut bagian penjualan *pso* bahwa sistem informasi *supply chain management* yang dibangun sangat memberikan kemudahan untuk bagian penjualan *pso*. 5. Menurut anda, apakah sistem informasi *supply chain management* ini menggunakan bahasa yang mudah untuk dimengerti? Jawaban: Menurut bagian penjualan *pso* bahwa sistem informasi *supply chain management* yang dibangun dikarenakan sistem yang dibangun sederhana sehingga sangat mudah untuk dimengerti. 6. Apakah Sistem Informasi *supply chain management* ini memberikan kemudahan dalam mendistribusikan produk pupuk oleh penjualan PSO? Jawaban: Menurut bagian penjualan *pso* bahwa sistem informasi *supply chain management* yang dibangun cukup membantu dalam proses pendistribusian produk pupuk dari gudang lini 2 ke gudang lini 3 sebagai sasaran tujuan. 7. Apakah Sistem informasi *supply chain management* ini memberikan kemudahan dalam memantau stok produk pupuk pada gudang lini 3 oleh penjualan PSO? Jawaban: Menurut bagian penjualan *pso* bahwa sistem informasi *supply chain management* yang dibangun cukup membantu dalam memantau stok produk pupuk pada gudang lini 3.

Hasil wawancara yang dilakukan terhadap level pengguna bagian perencanaan pengendalian material dan bagian penjualan pso adalah bahwa sistem informasi *supply chain management* yang dibangun mengenai perhitungan pembelian bahan baku sudah cukup benar dan sesuai dengan yang dibutuhkan. Untuk pendistribusian produk pupuk sendiri sudah cukup membantu dalam menentukan pengiriman produk pupuk ke setiap gudang lini 3 sesuai dengan

target pemesanan permentan. Sedangkan untuk tampilan sistem yang dibangun sudah cukup menarik dan saran untuk kedepannya aplikasi yang dibangun harus lebih banyak fitur tambahan.

4.2.5. Kesimpulan Hasil Pengujian Beta

Berdasarkan pengujian beta yang telah dilakukan maka dapat ditarik sebuah kesimpulan bahwa sistem informasi *supply chain management* yang dibangun telah memenuhi kebutuhan pengguna yakni bagian perencanaan pengendalian material dapat terbantu dalam melakukan perhitungan pembelian bahan baku yang harus dibeli oleh bagian pengadaan agar kegiatan produksi terus berlanjut, bagian penjualan *pso* telah terbantu dalam menentukan pendistribusian produk pupuk ke setiap gudang lini 3 sesuai dengan target pemesanan yang ditentukan oleh permentan.