BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Tahap Implementasi sistem merupakan tahap penciptaan perangkat lunak, tahap kelanjutan perancangan sistem. Tahap ini merupakan tahap dimana sistem siap untuk dijalankan. Tahap ini pula merupakan tahap menerjemah perancangan dari tahap analisis yang sebelumnya sudah dilakukan kedalam bahasa yang sapat mengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sesungguhnya.

4.1.1 Implementasi Perangkat Keras

Perangkat keras (*Hardwere*) yang diperlukan untuk mengimplementasikan Sistem Inventori dan Distribusi Dengan Menggunakan Pendekatan *Supply Chain Management* Pada CV.CIPTA MANDIRI CIMAHI adalah sebagai berikut:

a. *Processor* 2.2 Ghz
b. RAM 1 GB
c. *HardDisk* 80 GB
d. Monitor 17 “
e. Mouse dan Keyboard

4.1.2 Implementasi Perangkat Lunak

Perangkat Lunak (*Softwere*) yang diperlukan untuk mengembangkan pembangunan Sistem Inventori dan Distribusi Dengan Menggunakan Pendekatan *Supply Chain Management* Pada CV.CIPTA MANDIRI CIMAHI adalah sebagai berikut:

a. Sistem Operasi Windows 7
b. Web Browser
c. *Xamp* sebagai *webserver*

4.1.3 Implementasi Form

Implementasi form dilakukan terhadap setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk file program. Berikut ini adalah implementasi form untuk pengguna.

1. Implementasi form untuk Admin

Implementasi halaman di bawah ini ditunjukan untuk admin yang dibuat untuk dapat memonitoring dan melaakukan peramalan dapat dilihat pada tabel 4.1 sebagai berikut

Tabel 4.1 Implementasi Form

Menu	Deskripsi	Nama file
Login User	Fille program untuk proses user login	Login_user.php Login.php Logout.php
Dasboard	File program untuk proses dasboard	Ops.php Main.php
Pengolahan data	File program untuk proses olah data	Daftar_stok.php Daftar_pembelian.php Daftar_permintaan.php Daftar_penjualan.php Daftar_distribusi.php
Data master	File program untuk proses data master	Daftar_supplier.php Daftar_barang.php

		Daftar_cabang.php
Data pengguna	File program untuk proses data pengguna	Hapus_admin.php Ubah_admin.php Hapus_user.php Ubah_user.php Tambah_admin.php
Peramalan	File program untuk proses peramalan	Data_peramalan.php
Laporan	File program untuk proses laporan	Laporan_penjualan.php Laporan_peramalan.php Laporan_distribusi.php Laporan_pembelian.php Laporan_stok.php

2. Implementasi form untuk Gudang

Tabel 4.2 Implementasi Form Untuk Gudang

Menu	Deskripsi	Nama file
Login User	File program untuk proses user login	Login_user.php Login.php Logout.php
Pengolahan data	File program untuk proses pengolahan data	Daftar_stok.php Daftar_pembelian.php
Peramalan	File program untuk proses peramalan	Data_peramalan.php
Laporan	Fileprogram untuk proses laporan	lap_pembelan,php laporan_stok.php

3. Implementasi form untuk Distribusi

Tabel 4.3 Implementasi Form Untuk Distribusi

Menu	Deskripsi	Nama file
Login User	File program untuk proses user login	Login_user.php Login.php Logout.php
Pengolahan data	File program untuk proses pengolahan data	Daftar_permintaan.php Daftar_penjualan.php Daftar_distribusi.php
Laporan	File program untuk proses laporan	Laporan_penjualan.php Laporan_peramalan.php Laporan_distribusi.php

4.1.4 Implementasi Basis Data

Implementasi basis data dibangun berdasarkan perancangan basis data yang dibuat sebelumnya. Basis data yang dibutuhkan akan diimplementasikan menggunakan perangkat lunak *MySQL*. Berikut ini akan dijelaskan bagaimana proses pembuatan basis data beserta table-tabel yang mendukung sistem.

1. Pembuatan Database

```
CREATE DATABASE `db_inventory_scm`
```

2. Pembuatan table admin

```
CREATE TABLE IF NOT EXISTS `tb_admin` (
  `kode_admin` varchar(30) NOT NULL,
  `username_admin` varchar(75) NOT NULL,
  `password_admin` varchar(75) NOT NULL,
```

```
  `nama_admin` varchar(75) NOT NULL,
  `email_admin` varchar(75) NOT NULL,
  `level` varchar(30) NOT NULL,
  PRIMARY KEY (`kode_admin`),
  UNIQUE KEY `username` (`username_admin`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_user`
 ADD CONSTRAINT `tb_user_ibfk_1` FOREIGN KEY (`kode_cabang`)
REFERENCES `tb_cabang` (`kode_cabang`) ON DELETE CASCADE ON
UPDATE CASCADE,
 ADD CONSTRAINT `tb_user_ibfk_2` FOREIGN KEY (`kode_admin`)
REFERENCES `tb_supplier` (`kode_admin`);
```

3. Pembuatan table barang

```
CREATE TABLE IF NOT EXISTS `tb_barang` (
`kode_barang` varchar(30) NOT NULL,
`nama_barang` varchar(75) NOT NULL,
`deskripsi_barang` varchar(250) NOT NULL,
 `stok_barang` int(11) NOT NULL,
`kode_admin` varchar(30) NOT NULL,
PRIMARY KEY (`kode_barang`),
KEY `kode_satuan` (`kode_admin`),
KEY `admin_id` (`kode_admin`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `tb_barang`
ADD CONSTRAINT `tb_barang_ibfk_1` FOREIGN KEY (`kode_admin`)
REFERENCES `tb_admin` (`kode_admin`);
```

4. Pembuatan table cabang

```
CREATE TABLE IF NOT EXISTS `tb_cabang` (
`kode_cabang` varchar(30) NOT NULL,
`nama_cabang` varchar(75) NOT NULL,
`alamat_cabang` varchar(200) NOT NULL,
`telepon_cabang` varchar(20) NOT NULL,
`kode_admin` varchar(35) NOT NULL,
PRIMARY KEY (`kode_cabang`),
KEY `kode_admin` (`kode_admin`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_cabang`
ADD CONSTRAINT `tb_cabang_ibfk_1` FOREIGN KEY (`kode_admin`)
REFERENCES `tb_admin` (`kode_admin`);
```

5. Pembuatan table detail pembelian

```
CREATE TABLE IF NOT EXISTS `tb_detailpembelian` (
`num_beli` int(11) NOT NULL AUTO_INCREMENT,
`qty_beli` int(11) NOT NULL,
`kode_barang` varchar(30) NOT NULL,
`faktur_beli` varchar(30) NOT NULL,
PRIMARY KEY (`num_beli`),
```

```
KEY `kode_barang` (`kode_barang`,`faktur_beli`),
KEY `kode_jenistransaksibeli` (`faktur_beli`),
KEY `faktur_beli` (`faktur_beli`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

ALTER TABLE `tb_detailpembelian`
ADD CONSTRAINT `tb_detailpembelian_ibfk_1` FOREIGN KEY (`kode_barang`) REFERENCES `tb_barang` (`kode_barang`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `tb_detailpembelian_ibfk_2` FOREIGN KEY (`faktur_beli`) REFERENCES `tb_pembelian` (`faktur_beli`) ON DELETE CASCADE ON UPDATE CASCADE;
```

6. Pembuatan table detail distribusi

```
CREATE TABLE IF NOT EXISTS `tb_detail_distribusi` (
`id_distribusi` varchar(20) NOT NULL,
`kode_barang` varchar(50) NOT NULL,
`jumlah` int(11) NOT NULL,
KEY `id_distribusi` (`id_distribusi`),
KEY `kode_barang` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_detail_distribusi`
ADD CONSTRAINT `tb_detail_distribusi_ibfk_2` FOREIGN KEY (`kode_barang`) REFERENCES `tb_barang` (`kode_barang`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `tb_detail_distribusi_ibfk_1` FOREIGN KEY (`id_distribusi`) REFERENCES `tb_distribusi_brg` (`id_distribusi`) ON DELETE CASCADE ON UPDATE CASCADE;
```

7. Pembuatan table detail pemesanan

```
CREATE TABLE IF NOT EXISTS `tb_detail_pemesanan` (
`kode_pesan` varchar(30) NOT NULL,
`kode_barang` varchar(30) NOT NULL,
`qty` int(11) NOT NULL,
KEY `kode_pesan` (`kode_pesan`,`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_detail_pemesanan`
ADD CONSTRAINT `tb_detail_pemesanan_ibfk_1` FOREIGN KEY (`kode_pesan`) REFERENCES `tb_pemesanan` (`kode_pesan`) ON DELETE CASCADE ON UPDATE CASCADE;
```

8. Pembuatan table detail penjualan

```
CREATE TABLE IF NOT EXISTS `tb_detail_penjualan` (
`id_transaksi` varchar(16) NOT NULL,
`kode_barang` varchar(16) NOT NULL,
`jumlah` int(11) NOT NULL,
KEY `id_transaksi` (`id_transaksi`),
KEY `kode_barang` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `tb_detail_penjualan`
ADD CONSTRAINT `tb_detail_penjualan_ibfk_3` FOREIGN KEY (`id_transaksi`) REFERENCES `tb_penjualan` (`id_transaksi`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `tb_detail_penjualan_ibfk_2` FOREIGN KEY (`kode_barang`) REFERENCES `tb_barang` (`kode_barang`) ON DELETE CASCADE ON UPDATE CASCADE;
```

9. Pembuatan table detail permintaan

```
CREATE TABLE IF NOT EXISTS `tb_detail_permintaan` (
`id_permintaan` varchar(16) NOT NULL,
`kode_barang` varchar(16) NOT NULL,
`jumlah` int(11) NOT NULL,
KEY `id_permintaan` (`id_permintaan`),
KEY `kode_barang` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;



ALTER TABLE `tb_detail_permintaan`
ADD CONSTRAINT `tb_detail_permintaan_ibfk_1` FOREIGN KEY (`id_permintaan`) REFERENCES `tb_permintaan` (`id_permintaan`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `tb_detail_permintaan_ibfk_2` FOREIGN KEY (`kode_barang`) REFERENCES `tb_barang` (`kode_barang`) ON DELETE CASCADE ON UPDATE CASCADE;
```

10. Pembuatan table distribusi barang

```
CREATE TABLE IF NOT EXISTS `tb_distribusi_brg` (
`id_distribusi` varchar(20) NOT NULL,
`kode_cabang` varchar(30) NOT NULL,
`tgl_jual` date NOT NULL DEFAULT '0000-00-00',
`tgl_distribusi` date NOT NULL DEFAULT '0000-00-00',
`status` tinyint(3) NOT NULL,
PRIMARY KEY (`id_distribusi`),
KEY `kode_cabang` (`kode_cabang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_distribusi_brg`
ADD CONSTRAINT `tb_distribusi_brg_ibfk_1` FOREIGN KEY (`kode_cabang`) REFERENCES `tb_cabang` (`kode_cabang`) ON DELETE CASCADE ON UPDATE CASCADE;
```

11. Pembuatan table jenis transaksi beli

```
CREATE TABLE IF NOT EXISTS `tb_jenistransaksibeli` (
`kode_jenistransaksibeli` varchar(30) NOT NULL,
`jenis_transaksibeli` varchar(75) NOT NULL,
PRIMARY KEY (`kode_jenistransaksibeli`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

12. Pembuatan table jenis transaksi jual

```
CREATE TABLE IF NOT EXISTS `tb_jenistransaksijual` (
`kode_jenistransaksijual` varchar(35) NOT NULL,
`jenis_transaksijual` varchar(75) NOT NULL,
```

```
PRIMARY KEY (`kode_jenistransaksijual`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

13. Pembuatan table log stok beli

```
CREATE TABLE IF NOT EXISTS `tb_logstokbeli` (
  `num_log` int(11) NOT NULL AUTO_INCREMENT,
  `tgl_update` date NOT NULL,
  `jam_update` time NOT NULL,
  `stok_awal` int(11) NOT NULL,
  `stok_akhir` int(11) NOT NULL,
  `kode_barang` varchar(30) NOT NULL,
  `kode_admin` varchar(30) NOT NULL,
  `faktur_beli` varchar(30) NOT NULL,
  PRIMARY KEY (`num_log`),
  KEY `kode_barang` (`kode_barang`,`kode_admin`,`faktur_beli`),
  KEY `kode_admin` (`kode_admin`),
  KEY `faktur_beli` (`faktur_beli`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;



ADD CONSTRAINT `tb_logstokbeli_ibfk_1` FOREIGN KEY (`kode_barang`) REFERENCES `tb_barang` (`kode_barang`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `tb_logstokbeli_ibfk_2` FOREIGN KEY (`kode_admin`) REFERENCES `tb_admin` (`kode_admin`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `tb_logstokbeli_ibfk_3` FOREIGN KEY (`faktur_beli`) REFERENCES `tb_pembelian` (`faktur_beli`) ON DELETE CASCADE ON UPDATE CASCADE;
```

14. Pembuatan table log stok jual

```
CREATE TABLE IF NOT EXISTS `tb_logstokjual` (
 `num_log` int(11) NOT NULL AUTO_INCREMENT,
 `tgl_update` datetime NOT NULL,
 `stok_awal` int(11) NOT NULL,
 `stok_akhir` int(11) NOT NULL,
 `kode_barang` varchar(30) NOT NULL,
 `id_transaksi` varchar(30) NOT NULL,
 PRIMARY KEY (`num_log`),
 KEY `kode_barang` (`kode_barang`,`id_transaksi`),
 KEY `faktur_jual` (`id_transaksi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=9
;


ALTER TABLE `tb_logstokjual`
 ADD CONSTRAINT `tb_logstokjual_ibfk_1` FOREIGN KEY (`kode_barang`) REFERENCES `tb_barang` (`kode_barang`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `tb_logstokjual_ibfk_3` FOREIGN KEY (`id_transaksi`) REFERENCES `tb_penjualan` (`id_transaksi`) ON DELETE CASCADE ON UPDATE CASCADE;
```

15. Pembuatan table pembelian

```
CREATE TABLE IF NOT EXISTS `tb_pembelian` (
  `faktur_beli` varchar(30) NOT NULL,
  `tgl_beli` date NOT NULL,
  `kode_supplier` varchar(30) NOT NULL,
  `kode_jenistransaksibeli` varchar(30) NOT NULL,
  PRIMARY KEY (`faktur_beli`),
  KEY `kode_supplier` (`kode_supplier`,`kode_jenistransaksibeli`),
  KEY `kode_jenistransaksibeli` (`kode_jenistransaksibeli`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_pembelian`
  ADD CONSTRAINT `tb_pembelian_ibfk_1` FOREIGN KEY (`kode_supplier`) REFERENCES `tb_supplier` (`kode_supplier`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `tb_pembelian_ibfk_2` FOREIGN KEY (`kode_jenistransaksibeli`) REFERENCES `tb_jenistransaksibeli` (`kode_jenistransaksibeli`) ON DELETE CASCADE ON UPDATE CASCADE;
```

16. Pembuatan table pemesanan

```
CREATE TABLE IF NOT EXISTS `tb_pemesanan` (
  `kode_pesan` varchar(30) NOT NULL,
  `kode_suplier` varchar(30) NOT NULL,
  `tgl_beli` date NOT NULL,
  `status` tinyint(1) NOT NULL,
```

```
  PRIMARY KEY (`kode_pesan`),
  KEY `kode_suplier` (`kode_suplier`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_pemesanan`
ADD CONSTRAINT `tb_pemesanan_ibfk_1` FOREIGN KEY (`kode_suplier`) REFERENCES `tb_supplier` (`kode_supplier`) ON DELETE CASCADE ON UPDATE CASCADE;
```

17. Pembuatan table penjualan

```
CREATE TABLE IF NOT EXISTS `tb_penjualan` (
  `id_transaksi` varchar(30) NOT NULL,
  `tgl_jual` date NOT NULL,
  `kode_cabang` varchar(30) NOT NULL,
  `kode_admin` varchar(16) NOT NULL,
  `kode_jenistransaksijual` varchar(30) NOT NULL,
  PRIMARY KEY (`id_transaksi`),
  KEY `kode_cabang` (`kode_cabang`,`kode_jenistransaksijual`),
  KEY `kode_jenistransaksijual` (`kode_jenistransaksijual`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_penjualan`
 ADD CONSTRAINT `tb_penjualan_ibfk_1` FOREIGN KEY (`kode_cabang`) REFERENCES `tb_cabang` (`kode_cabang`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `tb_penjualan_ibfk_2` FOREIGN KEY
```

```
(`kode_jenistransaksijual`) REFERENCES `tb_jenistransaksijual`
(`kode_jenistransaksijual`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

18. Pembuatan table peramalan

```
 CREATE TABLE IF NOT EXISTS `tb_peramalan` (
   `id_peramalan` int(11) NOT NULL AUTO_INCREMENT,
   `kode_barang` varchar(50) NOT NULL,
   `by_month` bigint(20) NOT NULL,
   `jumlah` int(11) NOT NULL,
   `kode_cabang` varchar(30) NOT NULL,
   `tgl_peramalan` date NOT NULL,
   PRIMARY KEY (`id_peramalan`),
   KEY `kode_cabang` (`kode_cabang`),
   KEY `kode_barang` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=115 ;
```

19. Pembuatan table permintaan

```
CREATE TABLE IF NOT EXISTS `tb_permintaan` (
  `id_permintaan` varchar(30) NOT NULL,
  `tgl_jual` datetime NOT NULL,
  `kode_cabang` varchar(30) NOT NULL,
  `status` tinyint(4) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_permintaan`),
  KEY `kode_cabang` (`kode_cabang`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_permintaan`
ADD CONSTRAINT `tb_permintaan_ibfk_1` FOREIGN KEY (`kode_cabang`) REFERENCES `tb_cabang` (`kode_cabang`) ON DELETE CASCADE ON UPDATE CASCADE;
```

20. Pembuatan table supplier

```
CREATE TABLE IF NOT EXISTS `tb_supplier` (
 `kode_supplier` varchar(30) NOT NULL,
 `nama_supplier` varchar(75) NOT NULL,
 `alamat_supplier` varchar(300) NOT NULL,
 `kode_admin` varchar(30) NOT NULL,
 PRIMARY KEY (`kode_supplier`),
 KEY `admin_id` (`kode_admin`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_supplier`
ADD CONSTRAINT `tb_supplier_ibfk_1` FOREIGN KEY (`kode_admin`) REFERENCES `tb_admin` (`kode_admin`);
```

21. Pembuatan table user

```
CREATE TABLE IF NOT EXISTS `tb_user` (
  `kode_user` varchar(30) NOT NULL,
  `username_user` varchar(75) NOT NULL,
  `password_user` varchar(75) NOT NULL,
```

```
  `nama_user` varchar(75) NOT NULL,
  `email_user` varchar(75) NOT NULL,
  `kode_cabang` varchar(30) NOT NULL,
  `kode_admin` varchar(30) NOT NULL,
  PRIMARY KEY (`kode_user`),
  KEY `kode_cabang` (`kode_cabang`,`kode_admin`),
  KEY `kode_admin` (`kode_admin`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_user`
  ADD CONSTRAINT `tb_user_ibfk_1` FOREIGN KEY (`kode_cabang`) REFERENCES `tb_cabang` (`kode_cabang`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `tb_user_ibfk_2` FOREIGN KEY (`kode_admin`) REFERENCES `tb_supplier` (`kode_admin`);
```

22. Pembuatan table temp_permintaan

```
CREATE TABLE IF NOT EXISTS `temp_permintaan` (
  `kode` int(11) NOT NULL AUTO_INCREMENT,
  `kode_barang` varchar(30) NOT NULL,
  `kode_cabang` varchar(30) NOT NULL,
  `nama_barang` varchar(100) NOT NULL,
  `jumlah` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

23. Pembuatan table temp_pesan

```
CREATE TABLE IF NOT EXISTS `temp_pesan` (
  `kode` int(11) NOT NULL AUTO_INCREMENT,
  `kode_barang` varchar(30) NOT NULL,
  `kode_supplier` varchar(30) NOT NULL,
  `nama_barang` varchar(100) NOT NULL,
  `jumlah` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

4.2 Pengujian

Tahap selanjutnya adalah tahap pengujian sistem. Tahap ini merupakan hal terpenting yang bertujuan untuk menemukan kesalahan - kesalahan ataupun kekurangan – kekurangan pada aplikasi yang dibangun. Pengujian ini bermaksud untuk mengetahui apakah aplikasi yang dibuat telah memenuhi criteria yang sesuai dengan tujuan perancangan aplikasi atau belum.

Pengujian perangkat lunak ini menggunakan pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

4.2.1 Rencana Pengujian Sistem

Rencana pengujian yang akan dilakukan dengan menguji sistem secara alpha menggunakan metode black-box dan beta dengan wawancara. Pengujian sistem ini menggunakan data uji berdasarkan data yang terdapat di CV.CIPTA MANDIRI. Rencana pengujian selengkapnya terlihat pada table 4.4 di bawah ini.

Tabel 4.4 Rencana Pengujian

Item Pengujian	Detail Pengujian	Jenis Pengujian
Login pengguna	Verifikasi *username*,	*Black Box*

	password	
Lupa password	Verifikasi *password* ke *email* pengguna	*Black Box*
Pengolahan data pengguna	Tambah data	*Black Box*
	Ubah data pengguna	*Black Box*
	Hapus data pengguna	*Black Box*
Pengolahan data barang	Tambah data barang	*Black Box*
	Ubah data barang	*Black Box*
	Hapus data barang	*Black Box*
	Cari data barang	*Black Box*
Pengolahan data pembelian	Tambah data pembelian	*Black Box*
	Hapus data pembelian	*Black Box*
	Cari data pembelian	*Black Box*
Pengolahan data permintaan	Tambah data permintaan	*Black Box*
	Cari data permintaan	*Black Box*
Pengolahan data supplier	Tambah data supplier	*Black Box*
	Ubah data supplier	*Black Box*
	Hapus data supplier	*Black Box*
Pengolahan data cabang	Tambah data cabang	*Black Box*
	Ubah data cabang	*Black Box*
	Hapus data cabang	*Black Box*
	Cari data cabang	*Black Box*

4.2.2 Kasus dan Hasi Pengujian

Hasil implementasi perangkat lunak sistem inventori dan distribusi di CV.CIPTA MANDIRI ini diuji pada pengguna admin, bagian gudang dan bagian distribusiyang berbasis web melalui spesifikasi perangkat sistem agar sistem dapat berjalan dengan baik.

Pengujian perangkat lunak Sistem Inventori dan Distribusi menggunakan Supply Chain Management pada CV.CIPTA MANDIRI menggunakan data uji berdasarkan halaman yang terdapat didalam aplikasi.

1. Pengujian Login

Tabel 4.5 Pengujian Login Pengguna sebagai Admin

Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
username Contoh masukan : *username* : admin	Data *username* yang dimasukan akan tampil pada masing – masing isian	Data *username* yang dimasukan tampil	[√] Diterima [ ] Ditolak
Password masukan : Password : admin	Data password yang dimasukan akan tampil pada masing – masing isian	Data password yang dimasukan tampil	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data salah)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
email Contoh masukan :	Tampilkan pesan kesalahan email	Dapat menampilkan pesan	[√] Diterima [ ] Ditolak

ekky@yahoo.com	dan password yang tidak terdaftar	kesalahan	
Password Contoh masukan : 1234	Tampilkan pesan kesalahan email dan password tidak terdaftar	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

2. Pengujian Lupa Password

Tabel 4.6 Pengujian Lupa Password

Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Username Contoh masukan : Username : mimin@gmail.com	Data email yang dimasukan akan dikirim verifikasi untuk mengganti password	Data verifikasi terkirim ke email yang dimasukan	[√] Diterima [ ] Ditolak

Kasus dan Hasil uji (Data salah)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
email Contoh masukan : ekky@yahoo.com	Tampilkan pesan kesalahan email tidak terdaftar	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

3. Pengujian Pengolahan Data Pengguna

Tabel 4.7 Pengujian pengolahan Data Pengguna

Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
username : admin2 password : admin2 konfigurasi password : admin2	Data yang dimasukan apabila benar akan menambahkan data pengguna	Berhasil menambah data pengguna	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data salah)			
username : admin@gmail.com password : konfigurasi password: Jabatan :	Tampilkan pesan kesalahan field tidak boleh kosong dan pesan mohon koreksi kembali data yang dimasukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Ubah data pengguna			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
username : admin2 password : admin2 konfigurasi password : admin2	Mengubah data yang dimasukan apabila benar akan mengubah data pengguna	Berhasil mengubah data pengguna	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			

email : admin@gmail.com password : konfigurasi password :	Tampilkan pesan kesalahan field tidak boleh kosong	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Hapus Data Pengguna			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
username : admin2 password : admin2 konfigurasi password : admin2	Tampilkan pesan, yakin akan menghapus? Ya/Tidak menghapus data telah dipilih	Berhasil menghapus data	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
username : admin2 password : konfigurasi password :	Tampilkan pesan data gagal dihapus	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Cari data Pengguna			
Kasus dan Hasil uji (Data Normal)			
Data masukan	Keluaran yang diharapkan	pengamatan	Kesimpulan
username : admin2	Pengguna yang	Berhasil mencari	[√] Diterima

	akan ditemukan	pengguna	[ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
username : admin2	Tampilkan pesan data pengguna tidak ditemukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

4. Pengujian Data Barang

Tabel 4.8 Pengujian Data Barang

Tambah Data Barang			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Nama Barang : Master Rem Kode barang : BRG005 Jenis barang : Rem Stok minimal : 100	Data yang dimasukan apabila benar akan menambahkan data barang	Berhasil menambah data barang	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data salah)			
Nama Barang : Master Rem Kode barang : Jenis barang : Stok minimal :	Tampilkan pesan kesalahan field tidak boleh kosong dan pesan mohon koreksi kembali data yang dimasukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

Ubah Data Barang			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Nama Barang : Master Rem Kode barang : BRG005 Jenis barang : Rem Stok minimal : 100	Mengubah data yang dimasukan apabila benar akan mengubah data barang	Berhasil mengubah data barang	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Nama Barang : Master Rem Kode barang : BRG005 Jenis barang : Rem Stok minimal :	Tampilkan pesan kesalahan field tidak boleh kosong	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Hapus Data Barang			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Nama Barang : Master Rem Kode barang : BRG005 Jenis barang : Rem	Tampilkan pesan, yakin akan menghapus? Ya/Tidak menghapus data	Berhasil menghapus data	[√] Diterima [ ] Ditolak

Stok minimal : 100	telah dipilih		
Kasus dan Hasil uji (Data Salah)			
Nama Barang : Master Rem Kode barang : BRG005 Jenis barang : Rem Stok minimal :	Tampilkan pesan data gagal dihapus	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Cari data Barang			
Kasus dan Hasil uji (Data Normal)			
Data masukan	Keluaran yang diharapkan	pengamatan	Kesimpulan
Nama Barang : Master Rem Kode barang : BRG005 Jenis barang : Rem Stok minimal : 100	Barang yang akan ditemukan	Berhasil menampilkan barang yang dicari	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Nama Barang : Master Rem Kode barang : BRG005 Jenis barang : Rem Stok minimal :	Tampilkan pesan data barang tidak ditemukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

5. Pengujian Pengolahan Data Pembelian

Tabel 4.9 Pengujian Pengolahan Data pembelian

Tambah Data Pembelian			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Nama Supplier : Mitra Jaya Nama Barang : Master Rem Jumlah barang: 250	Data yang dimasukan apabila benar akan menambahkan data pembelian	Berhasil menambah data pembelian	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data salah)			
Nama Supplier : Mitra Jaya Nama Barang : Jumlah barang:	Tampilkan pesan kesalahan field tidak boleh kosong	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Hapus Data Pembelian			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Nama Supplier : Mitra Jaya Nama Barang : Master Rem Jumlah barang: 250	Tampilkan pesan, yakin akan menghapus? Ya/Tidak menghapus data telah dipilih	Berhasil menghapus data	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			

Nama Supplier : Mitra Jaya Nama Barang : Jumlah barang:	Tampilkan pesan data gagal dihapus	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Cari data pembelian			
Kasus dan Hasil uji (Data Normal)			
Data masukan	Keluaran yang diharapkan	pengamatan	Kesimpulan
Cari data pembelian sesuai status Contoh masukan : Dibeli	Data pembelian yang dicari ditemukan	Berhasil menampilkan data pembelian yang dicari	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Cari data pembelian sesuai status Contoh masukan : Mitra	Tampilkan pesan data pembelian tidak ditemukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

6. Pengujian Pengolahan Daftar Permintaan

Tabel 4.10 Pengujian Pengolahan Daftar Permintaan

Tambah Data Permintaan			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Nama Cabang : barokah	Data yang dimasukan apabila	Berhasil menambah data permintaan	[√] Diterima [ ] Ditolak

Nama Barang : Clutch Cover Jumlah barang: 250 Tanggal permintaan : 29 Juli 2013	benar akan menambahkan data permintaan		
Kasus dan Hasil uji (Data salah)			
Nama Cabang : barokah Nama Barang : Jumlah barang: Tanggal permintaan :	Tampilkan pesan kesalahan field tidak boleh kosong	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Cari data pembelian			
Kasus dan Hasil uji (Data Normal)			
Data masukan	Keluaran yang diharapkan	pengamatan	Kesimpulan
Cari data pembelian sesuai nama cabang Contoh masukan : Barokah	Data permintaan yang dicari ditemukan	Berhasil menampilkan data permintaan yang dicari	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Cari data pembelian sesuai nama cabang Contoh masukan :	Tampilkan pesan data permintaan tidak ditemukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

7. Pengujian Pengolahan Data Supplier

Tabel 4.11 Pengujian Pengolahan Data Supplier

Tambah Data Supplier			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Nama Supplier : Mitra Jaya Alamat Supplier : Jl.padalarang	Data yang dimasukan apabila benar akan menambahkan data supplier	Berhasil menambah data supplier	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data salah)			
Nama Supplier : Mitra Jaya Alamat Supplier :	Tampilkan pesan kesalahan field tidak boleh kosong	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Hapus Data Supplier			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Mencari Data supplier yang akan dihapus	Data supplier yang telah dipilih sudah dihapus	Berhasil menghapus data	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Mencari Data supplier yang akan dihapus	Tampilkan pesan data gagal dihapus	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Cari data supplier			
Kasus dan Hasil uji (Data Normal)			

Data masukan	Keluaran yang diharapkan	pengamatan	Kesimpulan
Cari data sesuai nama supplier Contoh masukan : Irian	Data supplier yang dicari ditemukan	Berhasil menampilkan data supplier yang dicari	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Cari data sesuai nama supplier Contoh masukan :	Tampilkan pesan data pembelian tidak ditemukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

8. Pengujian Pengolahan Data Cabang

Tabel 4.12 Pengujian Pengolahan Data Cabang

Tambah Data Cabang			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Nama Cabang : Family Alamat cabang : Jl.padalarang No.telp cabang : 123456	Data yang dimasukan apabila benar akan menambahkan data cabang	Berhasil menambah data cabang	[√] Diterima [ ] Ditolak

Kasus dan Hasil uji (Data salah)			
Nama Supplier : Mitra Jaya Alamat Supplier :	Tampilkan pesan kesalahan field tidak boleh kosong	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Ubah Data Cabang			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Memilih data cabang yang akan di ubah	Data cabang yang telah dipilih sudah diubah	Berhasil mengubah data cabang	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Memilih data cabang yang akan di ubah	Tampilkan pesan data gagal dihapus	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Hapus Data Cabang			
Kasus dan Hasil uji (Data Normal)			
Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Mencari Data cabang yang akan dihapus	Data cabang yang telah dipilih sudah dihapus	Berhasil menghapus data	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Mencari Data cabang yang akan dihapus	Tampilkan pesan data gagal dihapus	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

Cari data cabang			
Kasus dan Hasil uji (Data Normal)			
Data masukan	Keluaran yang diharapkan	pengamatan	Kesimpulan
Cari data sesuai nama cabang Contoh masukan : Irian	Data cabang yang dicari ditemukan	Berhasil menampilkan data cabang yang dicari	[√] Diterima [ ] Ditolak
Kasus dan Hasil uji (Data Salah)			
Cari data sesuai nama cabang Contoh masukan :	Tampilkan pesan data cabang tidak ditemukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

4.2.3 Kasus dan Hasil Pengujian Beta

Pengujian beta dilakukan di lingkungan pengguna. Pengujian ini merupakan pengujian yang bersifat langsung di lingkungan yang sebenarnya. Peneliti mendapat penilaian sistem secara langsung ke lapangan yaitu perusahaan yang bersangkutan dengan wawancara langsung kepada pengguna yang bersifat pertayaan menegnai kepuasan pengguna.

4.2.3.1 Wawancara

Berikut adalah hasil wawancara yang telah dilakukan terhadap admin, bagian gudang, bagian distribusi, dan pemilik pada CV.CIPTA MANDIRI yang menggunakan aplikasi sistem Inventori dan Distribusi :

A. Wawancara pada bagian Gudang

1. Apakah aplikasi dapat membantu dalam memberikan informasi persediaan barang?

Jawaban: Aplikasi dapat membantu dalam memberikan informasi mengenai produk yang tersedia.

2. Apakah aplikasi ini dapat membantu dalam pengolahan data barang dan permintaan barang?

   Jawaban: Aplikasi ini membantu kami dalam memasukan data barang dan pengiriman barang kepada cabang.

3. Apakah aplikasi yang dibangun sudah sesuai dengan kebutuhan mengenai informasi yang dibutuhkan untuk mengolah data bagian gudang?

   Jawaban: Sudah sesuai dengan kebutuhan kami khususnya bagian gudang ditambah kami mendapat seluruh informasi berupa laporan yang dibutuhkan disini.

4. Apakah aplikasi ini dapat membantu dalam meramalkan barang yang tersedia di gudang sebelum didistribusikan?

   Jawaban: Aplikasi ini sangat membantu kami sehingga mengetahui perkiraan berapa jumlah barang yang dibutuhkan untuk jumlah barang kedepannya.

5. Apakah aplikasi yang dibangun mudah digunakan?

   Jawaban: Aplikasi ini cukup mudah digunakan.

B. Wawancara pada bagian Distribusi

1. Apakah aplikasi dapat membantu meramalkan barang yang harus didistribusikan?

   Jawaban: Aplikasi ini sangat membantu kami dalam melakukan pengiriman barang sehingga membuat kami mempunyai gambaran.

2. Apakah aplikasi ini membantu bagian distribusi dalam pengiriman sesuai permintaan cabang?

   Jawaban: Aplikasi ini sangat membantu kami dalam proses pengiriman sesuai dengan jadwal dan jumlah barang yang akan dikirimkan sesuai keinginan cabang.

3. Apakah aplikasi yang dibangun sudah sesuai dengan kebutuhan bagian distribusi?

   Jawaban: Ya, kami sangat membutuhkan aplikasi seperti ini untuk memperlancar kinerja perusahaan serta memberikan informasi yang dibutuhkan bagian distribusi.

4. Apakah aplikasi yang dibangun mudah digunakan?

   Jawaban: Aplikasi ini cukup mudah saat digunakan dan dapat dimengerti dengan cepat.

4.2.4 Kesimpulan Hasil Pengujian Beta

Berdasarkan pengujian beta yang telah dilakukan di CV.CIPTA MANDIRI yaitu dengan wawancara kepada seluruh pengguna yang bersifat pertanyaan di lapangan, diperoleh kesimpulan bahwa sistem yang telah dibangun ini bernilai sudah sangat membantu dalam hal keseluruhan seperti dalam proses pengolahan data barang, informasi pengadaan, melakukan peramalan penjualan, peramalan stok barang yang dibutuhkan memberikan informasi apa yang dilakukan pengguna dalam menjalankan sistem dan memberikan kemudahan bagi perusahaan.