BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini dilakukan implementasi dan pengujian yang merupakan tahap penerjemahan kebutuhan pembangunan aplikasi kedalam representasi perangkat lunak sesuai dengan analisis yang telah dilakukan terhadap system yang dibangun. Setelah implementasi maka dilakukan pengujian system yang baru dimana akan dilihat kekuranga-kekurangan pada aplikasi yang baru untuk selanjutnya diadakan pengembangan system.

Tahapan ini biasa dilakukan setelah mampu menyelesaikan tahapan analisis dan perancangan, sehingga selanjutnya dapat diimplementasikan kedalam bahasa pemrograman.

4.1 Implementasi

Implementasi merupakan tahap dimana sistem merupakan hasil terjemahan dari perancangan dan analisis yang dituangkan kedalam bahasa yang dapat dimengerti mesin dalam keadaan yang sesungguhnya, sehingga mampu menunjukkan bahwa sistem telah siap untuk dipergunakan.

Implementasi bertujuan untuk menguji apakah sistem telah sesuai yang diharapkan oleh pengguna, sehingga akan terlihat kekuranga-kekurangan pada sistem yang akan dibangun untuk selanjutnya dapat dilkaukan pengembangan dan perbaikan pada sistem.

4.1.1 Implementasi Perangkat keras

Perangkat keras yang diperlukan untuk mengimplementasikan perangkat keras dari Sistem informasi Inventori Suku cadang Lokomotif berbasis web di pt.kereta api (persero) daop 2 bd unit dipo lokomotif bandung antara lain:

Tabel 4.1 Spesifikasi Perangkat Keras

Spesifikasi Perangkat Keras yang dibutuhkan	
Processor	Intel Pentium IV
Memori	1 Gb
Harddisk	80 Gb
Lan card	10/100 Mbps
Monitor	14”

4.1.2 Implementasi Perangkat Lunak

Adapun spesifikasi perangakat lunak yang dibutuhkan antara lain:

Tabel 4.2 Spesifikasi Perangkat Lunak

Spesifikasi Perangkat Lunak yang dibutuhkan	
Sistem Operasi	Windows XP/ Vista/7
Software Pendukung	Microsoft Office 2010 Browser Firefox, Internet Explorer, Google Chrome

4.1.3 Implementasi Basis Data

Pembuatan database dilakukan dengan menggunakan aplikasi DBMS MySQL. Implementasi dalam bahasa SQL adalah sebagai berikut:

1. Pembuatan Database

   CREATE DATABASE”sisinfoka”;

2. Struktur Data Tabel Pegawai

```
CREATE TABLE IF NOT EXISTS `pegawai` (
  `kode_pegawai` varchar(5) NOT NULL,
  `nama_pegawai` varchar(50) NOT NULL,
  `password` varchar(15) NOT NULL,
  `kode_jabatan` varchar(10) NOT NULL,
  PRIMARY KEY (`kode_pegawai`),
  KEY `kode_jabatan` (`kode_jabatan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `pegawai`
  ADD CONSTRAINT `pegawai_ibfk_1` FOREIGN KEY
(`kode_jabatan`) REFERENCES `jabatan` (`kode_jabatan`) ON
UPDATE CASCADE;
```

3. Struktur Data Tabel Jabatan

```
  CREATE TABLE IF NOT EXISTS `jabatan` (
  `kode_jabatan` varchar(10) NOT NULL,
  `nama_jabatan` varchar(25) NOT NULL,
  `hak_akses` varchar(35) NOT NULL,
  PRIMARY KEY (`kode_jabatan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4. Struktur Data Tabel Barang

```
CREATE TABLE IF NOT EXISTS `barang` (
  `kode_barang` varchar(10) NOT NULL,
  `nama_barang` varchar(50) NOT NULL,
  `stok_barang` bigint(4) NOT NULL,
  `kode_satuan` varchar(5) NOT NULL,
  `harga_satuan` int(11) NOT NULL,
```

```
  PRIMARY KEY (`kode_barang`),
  KEY `kode_satuan` (`kode_satuan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `barang`
  ADD CONSTRAINT `barang_ibfk_1` FOREIGN KEY
(`kode_satuan`) REFERENCES `satuan` (`kode_satuan`) ON
UPDATE CASCADE;
```

5. Struktur Data Tabel Satuan

```
CREATE TABLE IF NOT EXISTS `satuan` (
  `kode_satuan` varchar(5) NOT NULL,
  `nama_satuan` varchar(15) NOT NULL,
  PRIMARY KEY (`kode_satuan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6. Struktur Data Tabel Supplier

```
CREATE TABLE IF NOT EXISTS `supplier` (
  `kode_supplier` varchar(15) NOT NULL,
  `nama_supplier` varchar(50) NOT NULL,
  `alamat` varchar(100) NOT NULL,
  `no_telepon` varchar(15) NOT NULL,
  `email` varchar(100) NOT NULL,
  PRIMARY KEY (`kode_supplier`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

7. Struktur Data Tabel EOQ

```
CREATE TABLE IF NOT EXISTS `eoq` (
  `kode_eoq` int(4) NOT NULL AUTO_INCREMENT,
  `tahun_pemakaian` int(4) NOT NULL,
```

```
  `biaya_simpan` int(10) NOT NULL,
  `biaya_pesan` int(10) NOT NULL,
  `kode_barang` varchar(10) NOT NULL,
  `jumlah_pesan` float NOT NULL,
  `frekuensi_pesan` float NOT NULL,
  `periode_pesan` float NOT NULL,
  `waktu_tunggu` int(10) NOT NULL,
  `penggunaan` int(10) NOT NULL,
  `rop` int(10) NOT NULL,
  `keterangan` varchar(50) NOT NULL,
  PRIMARY KEY (`kode_eoq`),
  KEY `kode_barang` (`kode_barang`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8
AUTO_INCREMENT=9 ;
ALTER TABLE `eoq`
  ADD CONSTRAINT `eoq_ibfk_1` FOREIGN KEY
(`kode_barang`) REFERENCES `barang` (`kode_barang`) ON
UPDATE CASCADE;
```

8. Struktur Data Tabel Permintaan

```
CREATE TABLE IF NOT EXISTS `permintaan` (
  `no_permintaan` varchar(20) NOT NULL,
  `tgl_permintaan` datetime NOT NULL,
  `kode_pegawai` varchar(5) NOT NULL,
  `status_permintaan` int(2) NOT NULL DEFAULT '0',
  PRIMARY KEY (`no_permintaan`),
  KEY `kode_pegawai` (`kode_pegawai`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `permintaan`
```

```
  ADD CONSTRAINT `permintaan_ibfk_1` FOREIGN KEY
(`kode_pegawai`) REFERENCES `pegawai` (`kode_pegawai`) ON
UPDATE CASCADE;
```

9. Struktur Data Tabel Detil Permintaan

```
CREATE TABLE IF NOT EXISTS `detil_permintaan` (
  `no_permintaan` varchar(20) NOT NULL,
  `kode_barang` varchar(10) NOT NULL,
  `jumlah` int(4) NOT NULL,
  KEY `kode_barang` (`kode_barang`),
  KEY `no_permintaan` (`no_permintaan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `detil_permintaan`
  ADD CONSTRAINT `detil_permintaan_ibfk_2` FOREIGN KEY
(`kode_barang`) REFERENCES `barang` (`kode_barang`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `detil_permintaan_ibfk_1` FOREIGN KEY
(`no_permintaan`) REFERENCES `permintaan` (`no_permintaan`)
ON UPDATE CASCADE;
```

10. Struktur Data Tabel Pemakaian

```
CREATE TABLE IF NOT EXISTS `pemakaian` (
  `no_pemakaian` varchar(20) NOT NULL,
  `tgl_pemakaian` datetime NOT NULL,
  `no_permintaan` varchar(20) NOT NULL,
  `kode_pegawai` varchar(5) NOT NULL,
  PRIMARY KEY (`no_pemakaian`),
  KEY `no_permintaan` (`no_permintaan`),
  KEY `kode_pegawai` (`kode_pegawai`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `pemakaian`
  ADD CONSTRAINT `pemakaian_ibfk_2` FOREIGN KEY
(`kode_pegawai`) REFERENCES `pegawai` (`kode_pegawai`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `pemakaian_ibfk_1` FOREIGN KEY
(`no_permintaan`) REFERENCES `permintaan` (`no_permintaan`)
ON UPDATE CASCADE;
```

11. Struktur Data Tabel Detil Pemakaian

```
CREATE TABLE IF NOT EXISTS `detil_pemakaian` (
  `no_pemakaian` varchar(20) NOT NULL,
  `kode_barang` varchar(10) NOT NULL,
  `jumlah` int(4) NOT NULL,
  `harga_satuan` int(11) NOT NULL,
  KEY `kode_barang` (`kode_barang`),
  KEY `no_pemakaian` (`no_pemakaian`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `detil_pemakaian`
  ADD CONSTRAINT `detil_pemakaian_ibfk_1` FOREIGN KEY
(`no_pemakaian`) REFERENCES `pemakaian` (`no_pemakaian`)
ON UPDATE CASCADE,
  ADD CONSTRAINT `detil_pemakaian_ibfk_2` FOREIGN KEY
(`kode_barang`) REFERENCES `barang` (`kode_barang`) ON
UPDATE CASCADE;
```

12. Struktur Data Tabel PO

```
CREATE TABLE IF NOT EXISTS `po` (
  `kode_PO` varchar(20) NOT NULL,
```

```
  `tgl_PO` datetime NOT NULL,
  `kode_supplier` varchar(15) NOT NULL,
  `kode_pegawai` varchar(5) NOT NULL,
  `tgl_rencana_kirim` datetime NOT NULL,
  `status_po` int(2) NOT NULL DEFAULT '0',
  PRIMARY KEY (`kode_PO`),
  KEY `kode_supplier` (`kode_supplier`),
  KEY `kode_pegawai` (`kode_pegawai`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `po`
  ADD CONSTRAINT `po_ibfk_1` FOREIGN KEY
(`kode_supplier`) REFERENCES `supplier` (`kode_supplier`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `po_ibfk_2` FOREIGN KEY
(`kode_pegawai`) REFERENCES `pegawai` (`kode_pegawai`) ON
UPDATE CASCADE;
```

13. Struktur Data Tabel Detil PO

```
CREATE TABLE IF NOT EXISTS `detil_po` (
  `kode_PO` varchar(20) NOT NULL,
  `kode_barang` varchar(10) NOT NULL,
  `jumlah` int(4) NOT NULL,
  `harga` int(11) NOT NULL,
  `kode_eoq` int(4) NOT NULL,
  `jumlah_kekurangan` int(11) NOT NULL,
  KEY `kode_barang` (`kode_barang`),
  KEY `kode_PO` (`kode_PO`),
  KEY `kode_eoq` (`kode_eoq`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
 ALTER TABLE `detil_po`
 ADD CONSTRAINT `detil_po_ibfk_3` FOREIGN KEY
(`kode_eoq`) REFERENCES `eoq` (`kode_eoq`) ON UPDATE
CASCADE,
 ADD CONSTRAINT `detil_po_ibfk_1` FOREIGN KEY
(`kode_PO`) REFERENCES `po` (`kode_PO`) ON UPDATE
CASCADE,
 ADD CONSTRAINT `detil_po_ibfk_2` FOREIGN KEY
(`kode_barang`) REFERENCES `barang` (`kode_barang`) ON
UPDATE CASCADE;
```

14. Struktur Data Tabel Penerimaan

```
 CREATE TABLE IF NOT EXISTS `penerimaan` (
  `no_penerimaan` varchar(20) NOT NULL,
  `tgl_penerimaan` datetime NOT NULL,
  `kode_pegawai` varchar(5) NOT NULL,
  `kode_PO` varchar(20) NOT NULL,
  PRIMARY KEY (`no_penerimaan`),
  KEY `kode_pegawai` (`kode_pegawai`),
  KEY `kode_PO` (`kode_PO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `penerimaan`
 ADD CONSTRAINT `penerimaan_ibfk_2` FOREIGN KEY
(`kode_PO`) REFERENCES `po` (`kode_PO`) ON UPDATE
CASCADE,
 ADD CONSTRAINT `penerimaan_ibfk_1` FOREIGN KEY
(`kode_pegawai`) REFERENCES `pegawai` (`kode_pegawai`) ON
UPDATE CASCADE;
```

15. Struktur Data Tabel Detil Penerimaan

```
CREATE TABLE IF NOT EXISTS `detil_penerimaan` (
  `no_penerimaan` varchar(20) NOT NULL,
  `kode_barang` varchar(10) NOT NULL,
  `harga_satuan` int(11) NOT NULL,
  `jumlah` int(4) NOT NULL,
  KEY `kode_barang` (`kode_barang`),
  KEY `no_penerimaan` (`no_penerimaan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `detil_penerimaan`
  ADD CONSTRAINT `detil_penerimaan_ibfk_1` FOREIGN KEY
(`no_penerimaan`) REFERENCES `penerimaan` (`no_penerimaan`)
ON UPDATE CASCADE,
  ADD CONSTRAINT `detil_penerimaan_ibfk_2` FOREIGN KEY
(`kode_barang`) REFERENCES `barang` (`kode_barang`) ON
UPDATE CASCADE
```

16. Struktur Data Tabel Retur Penerimaan

```
CREATE TABLE IF NOT EXISTS `retur_penerimaan` (
  `no_retur` varchar(20) NOT NULL,
  `no_penerimaan` varchar(20) NOT NULL,
  `tgl_retur` datetime NOT NULL,
  `kode_pegawai` varchar(5) NOT NULL,
  PRIMARY KEY (`no_retur`),
  KEY `no_penerimaan` (`no_penerimaan`),
  KEY `kode_pegawai` (`kode_pegawai`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `retur_penerimaan`
```

```
  ADD CONSTRAINT `retur_penerimaan_ibfk_2` FOREIGN KEY
(`kode_pegawai`) REFERENCES `pegawai` (`kode_pegawai`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `retur_penerimaan_ibfk_1` FOREIGN KEY
(`no_penerimaan`) REFERENCES `penerimaan` (`no_penerimaan`)
ON UPDATE CASCADE;
```

17. Struktur Data Tabel Detil Retur Penerimaan

```
CREATE TABLE IF NOT EXISTS `detil_retur_penerimaan` (
  `no_retur` varchar(20) NOT NULL,
  `kode_barang` varchar(10) NOT NULL,
  `jumlah` int(4) NOT NULL,
  `harga_satuan` int(11) NOT NULL,
  `keterangan` varchar(50) NOT NULL,
  KEY `kode_barang` (`kode_barang`),
  KEY `no_retur` (`no_retur`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `detil_retur_penerimaan`
  ADD CONSTRAINT `detil_retur_penerimaan_ibfk_2` FOREIGN
KEY (`kode_barang`) REFERENCES `barang` (`kode_barang`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `detil_retur_penerimaan_ibfk_1` FOREIGN
KEY (`no_retur`) REFERENCES `retur_penerimaan` (`no_retur`)
ON UPDATE CASCADE;
```

18. Struktur Data Tabel Pengiriman

```
CREATE TABLE IF NOT EXISTS `pengiriman` (
  `no_pengiriman` varchar(20) NOT NULL,
  `tgl_pengiriman` datetime NOT NULL,
```

```
  `kode_PO` varchar(20) NOT NULL,
  `kode_supplier` varchar(15) NOT NULL,
  PRIMARY KEY (`no_pengiriman`),
  KEY `kode_supplier` (`kode_supplier`),
  KEY `kode_PO` (`kode_PO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `pengiriman`
  ADD CONSTRAINT `pengiriman_ibfk_1` FOREIGN KEY
(`kode_PO`) REFERENCES `po` (`kode_PO`) ON UPDATE
CASCADE,
  ADD CONSTRAINT `pengiriman_ibfk_2` FOREIGN KEY
(`kode_supplier`) REFERENCES `supplier` (`kode_supplier`) ON
UPDATE CASCADE;
```

19. Struktur Data Tabel Mutasi

```
CREATE TABLE IF NOT EXISTS `mutasi` (
  `kode_barang` varchar(10) NOT NULL,
  `tgl_transaksi` datetime NOT NULL,
  `saldo_awal` bigint(20) DEFAULT NULL,
  `terima` bigint(20) DEFAULT NULL,
  `pakai` bigint(20) DEFAULT NULL,
  `saldo_akhir` bigint(20) DEFAULT NULL,
  KEY `kode_barang` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.1.4 Implementasi Antar muka

Implementasi antarmuka dilakukan dengan setiap halaman program yang dibuat dan pengkodeannya dalam bentuk file program. Berikut ini adalah implementasi antarmuka yang dibuat:

Tabel 4.3 Implementasi Antarmuka

Menu	**Deskripsi**	**Nama File**
Login	Digunakan untuk halaman login	Login.php
Home	Digunakan sebagai halaman utama untuk akses ke semua file	Index. php
Barang	Digunakan untuk menambah, merubah, menghapus, dan mencari data barang	Barang.php
Satuan	Digunakan untuk menambah, merubah, menghapus, dan mencari data satuan	Satuan.php
Jabatan	Digunakan untuk menambah, menghapus, dan mencari data jabatan	Jabatan.php
Pegawai	Digunakan untuk menambah, menghapus, dan mencari data pegawai	Pegawai.php
Supplier	Digunakan untuk menambah, merubah, menghapus, dan mencari data supplier	Supplier.php
EOQ	Digunakan untuk menambah, menghapus, dan mencari data EOQ	Eoq_form.php
PO	Digunakan untuk menambah dan mencari data PO	Po_form.php
Penerimaan	Digunakan untuk menambah dan mencari data Penerimaan	Pemakaian.php
Permintaan	Digunakan untuk menambah dan mencari data Permintaan	Permintaan.php
Pemakaian	Digunakan untuk menambah dan mencari data Pemakaian	Pemakaian.php
Retur	Digunakan untuk menambah dan mencari data Retur	Retur.php

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk menguji system dimana pada tahap ini diketahui kesalahan-kesalahan sehingga dapat dilakukan perbaikan dan pengembangan sistem sehingga dapat diketahui apakah alpikasi telah sesuai dengan kebutuhannya.

Pengujian yang digunakan untuk menguji sistem yang baru adalah dengan mengggunakan metode pengujian black box. Pada pengujian black box lebih fokus pada persyaratan fungsional perangkat lunak dan pegujian ini berusaha menemukan kesalahan- kesalahan dalam kategori sebagai berikut:

1.Kesalahan Interface

2.Fungsi-fungsi yang tidak sesuai dengan prosedur

3.Kesalahan dalam struktur data

4.Kesalahan kinerja system

4.2.1 Rencana Pengujian Alpha

Pengujian Alpha dilakukan dengan menggunakan metode blak box

Pengujian alpha dilakukan pada pengguna sistem

Tabel 4.4 Rencana pengujian

Item Uji	**Detail Pengujian**	**JenisPengujian**
Login	Verifikasi Login	*Black-Box*
Pengolahan master data barang	Tambah, Ubah, Hapus	*Black-Box*
Pengolahan master data satuan	Tambah, Ubah, Hapus	*Black-Box*
Pengolahan master data jabatan	Tambah, Ubah, Hapus	*Black-Box*
Pengolahan master data pegawai	Tambah, Ubah, Hapus	*Black-Box*
Pengolahan master data supplier	Tambah, Ubah, Hapus	*Black-Box*
Pengolahan transaksi eoq	Tambah, Hapus	*Black-Box*
Pengolahan transaksi po	Tambah, Cari	*Black-Box*

Pengolahan transaksi penerimaan	Tambah, Cari	*Black-Box*
Pengolahan transaksi permintaan	Tambah, Cari	*Black-Box*
Pengolahan transaksi pemakaian	Tambah, Cari	*Black-Box*
Pengolahan transaksi retur	Tambah, Cari	*Black-Box*
Pengujian input diluar range	Tambah	*Black-Box*

4.2.2 Kasus dan Hasil Pengujian Alpha

Berdasarkan rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut:

4.2.2.1 Pengujian Login

Tabel 4.5 Pengujian Login

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukkan username dan password yang benar	Mengisikan data username pada field username dan password	Dapat mengisikan username pada field username dan password	[√] Diterima [ ] Ditolak
Klik tombol login untuk masuk pada menu utama masing-masing bagian	Data username dan password benar sehingga dapat masuk pada menu utama masing-masing bagian	Data username dan password benar sehingga dapat masuk pada menu utama masing-masing bagian	[√] Diterima [ ] Ditolak

Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Username: xxx Passord : xxx	Muncul pada halaman login pesan kesalahan gagal login	Muncul pada halaman login pesan kesalahan gagal login	[√] Diterima [ ] Ditolak
Username: xxx Passord : Kosong	Muncul pada halaman login pesan kesalahan gagal login	Muncul pada halaman login pesan kesalahan gagal login	[√] Diterima [ ] Ditolak
Username:Kosong Passord : xxx	Muncul pada halaman login pesan kesalahan gagal login	Muncul pada halaman login pesan kesalahan gagal login	[√] Diterima [ ] Ditolak
Username:Kosong Passord : Kosong	Masuk pada halaman gagal login	Masuk pada halaman gagal login	[√] Diterima [ ] Ditolak

4.2.2.2 Pengujian kelola barang

1. Tambah data barang

Tabel 4.6 Tambah data barang

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

Kode barang Nama barang Stok Harga satuan Kode satuan	Data tersimpan pada database dan data bertambah, muncul pesan:"satu data barang berhasil ditambah".	Data tersimpan pada database dan data bertambah, muncul pesan:"satu data barang berhasil ditambah".	[√]Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode barang contoh :dikosongkan	Menampilkan pesan kesalahan "field kode barang harus diisi"	Menampilkan pesan kesalahan "field kode barang harus diisi"	[√]Diterima [ ] Ditolak

2. Ubah data barang

Tabel 4.7 Ubah data barang

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode barang contoh :T358AA Nama barang Contoh:Bidur abar	Data tersimpan pada database dan data berubah, muncul pesan:" satu data barang berhasil diubah".	Data tersimpan pada database dan data berubah, muncul pesan:" satu data barang berhasil diubah".	[√]Diterima [ ]Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

Kode barang contoh:dikosongkan	Menampilkan pesan kesalahan"field kode barang harus diisi"	Menampilkan pesan kesalahan"field kode barang harus diisi"	[√]Diterima [ ]Ditolak

3. Hapus data barang

Tabel 4.8 Hapus data barang

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode barang contoh :T358 Nama barang Contoh:Bidur abar	Data di database berhasil di hapus dan muncul pesan "Apakah anda yakin akan menghapus data ini?"	Muncul pesan "Apakah anda yakin akan menghapus data ini?" sesuai yang diharapakan	[√]Diterima [ ]Ditolak

4.2.2.3 Pengujian kelola satuan

1. Tambah data satuan

Tabel 4.9 Tambah data satuan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

Kode satuan contoh :Lt Nama satuan Contoh:Liter	Data tersimpan pada database dan data bertambah, muncul pesan:"satu data satuan berhasil ditambah".	Data tersimpan pada database dan data bertambah, muncul pesan:"satu data satuan berhasil ditambah".	[√]Diterima [ ]Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode satuan Contoh:dikosongkan Nama satuan Contoh:Liter	Menampilkan pesan kesalahan "field kode satuan harus diisi"	Menampilkan pesan kesalahan "field kode satuan harus diisi"	[√]Diterima [ ]Ditolak

2.Ubah data satuan

Tabel 4.10 Ubah data satuan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode satuan contoh :Ltrs Nama satuan Contoh:Liters	Data tersimpan pada database dan data berubah, muncul pesan:" satu data satuan berhasil diubah".	Data tersimpan pada database dan data berubah, muncul pesan:" satu data satuan berhasil diubah".	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			

Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode satuan contoh:Dikosongkan Nama satuan Contoh:Dikosongkan	Menampilkan pesan kesalahan "field kode satuan harus diisi"	Menampilkan pesan kesalahan "field kode satuan harus diisi"	[√] Diterima [ ] Ditolak

3. Hapus data satuan

Tabel 4.11 Hapus data satuan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode satuan contoh :Lt Nama satuan Contoh:Liter	Muncul pesan "Apakah anda yakin akan menghapus data ini?"	Muncul pesan "Apakah anda yakin akan menghapus data ini?" sesuai yang diharapkan	[√] Diterima [ ] Ditolak

4.2.2.4 Pengujian kelola jabatan

1. Tambah data jabatan

Tabel 4.12 Tambah data jabatan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

Kode jabatan contoh :KDT Nama jabatan Contoh:Kepala Dipo Traksi	Data tersimpan pada database dan data bertambah, muncul pesan:”satu data jabatan berhasil ditambah”.	Data tersimpan pada database dan data bertambah, muncul pesan:”satu data jabatan berhasil ditambah”.	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode jabatan Contoh:dikosongkan Nama jabatan Contoh:dikosongkan	Menampilkan pesan kesalahan “field kode jabatan harus diisi”	Menampilkan pesan kesalahan “field kode jabatan harus diisi”	[√] Diterima [ ] Ditolak

2.Ubah data jabatan

Tabel 4.13 Ubah data jabatan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode jabatan contoh :KDTT Nama jabatan Contoh:Kepala Dipo Traksi	Data tidak tersimpan pada database dan data tidak berubah, muncul pesan :”kode jabatan	Data tidak tersimpan pada database dan data tidak berubah, muncul pesan :” kode jabatan	[√] Diterima [ ] Ditolak

	tersebut sudah tersimpan pada database".	tersebut sudah tersimpan pada database"	
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode jabatan contoh:Dikosongkan	Menampilkan pesan kesalahan "field kode jabatan harus diisi"	Menampilkan pesan kesalahan "field kode jabatan harus diisi"	[√] Diterima [ ] Ditolak

3. Hapus data jabatan

Tabel 4.14 Hapus data jabatan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode jabatan contoh :KDT Nama jabatan Contoh:Kepala Dipo Traksi	Muncul pesan "Apakah anda yakin akan menghapus data ini?"	Muncul pesan "Apakah anda yakin akan menghapus data ini?" sesuai yang diharapkan	[√] Diterima [ ] Ditolak

4.2.2.5 Pengujian kelola pegawai

1. Tambah data pegawai

Tabel 4.15 Tambah data pegawai

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode pegawai contoh :46035 Nama pegawai Contoh:Nurasa	Data tersimpan pada database dan data bertambah, muncul pesan:”satu data pegawai berhasil ditambah”.	Data tersimpan pada database dan data bertambah, muncul pesan:”satu data pegawai berhasil ditambah”.	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode pegawai Contoh:dikosongkan Nama pegawai Contoh:dikosongkan	Menampilkan pesan kesalahan “field kode pegawai harus diisi”	Menampilkan pesan kesalahan “field kode pegawai harus diisi”	[√] Diterima [ ] Ditolak

2.Ubah data pegawai

Tabel 4.16 Ubah data pegawai

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode pegawai contoh :37681 Nama pegawai Contoh:Mudiono	muncul pesan :"Pegawai dengan kode ini sudah masuk dalam database".	muncul pesan :"Pegawai dengan kode ini sudah masuk dalam database".	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode pegawai contoh:Dikosongkan	Menampilkan pesan kesalahan "field kode pegawai harus diisi"	Menampilkan pesan kesalahan "field kode pegawai harus diisi"	[√] Diterima [ ] Ditolak

3. Hapus data pegawai

Tabel 4.17 Hapus data pegawai

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

Kode pegawai contoh :37682 Nama pegawai Contoh:Mudiono	Muncul pesan “Apakah anda yakin akan menghapus data ini?”	Muncul pesan “Apakah anda yakin akan menghapus data ini?”	[√] Diterima [ ] Ditolak

4.2.2.6 Pengujian kelola Supplier

1. Tambah data Supplier

Tabel 4.18 Tambah data supplier

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode supplier contoh :spl4 Nama supplier Contoh:CV. Sanmorin	Data tersimpan pada database dan data bertambah, muncul pesan:”satu data supplier berhasil ditambah”	Data tersimpan pada database dan data bertambah, muncul pesan:”satu data supplier berhasil ditambah”	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode supplier Contoh:dikosongkan	Menampilkan pesan kesalahan : “field kode supplier harus diisi”.	Menampilkan pesan kesalahan : “field kode supplier harus diisi”.	[√] Diterima [ ] Ditolak

2.Ubah data supplier

Tabel 4.19 Ubah data supplier

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode supplier contoh :spl4a Nama supplier Contoh: CV. Sanmorin	Data tersimpan pada database dan data berubah, muncul pesan:”satu data supplier berhasil diubah”.	Data tersimpan pada database dan data berubah, muncul pesan:”satu data supplier berhasil diubah”.	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode supplier contoh:Dikosongkan	Menampilkan pesan kesalahan : “field kode supplier harus diisi”.	Menampilkan pesan kesalahan : “field kode supplier harus diisi”.	[√] Diterima [ ] Ditolak

3. Hapus data supplier

Tabel 4.20 Hapus data supplier

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode supplier contoh :spl 4 Nama supplier Contoh:CV. Sanmorin	Muncul pesan "Apakah anda yakin akan menghapus data ini?" Data di database berhasil dihapus	Muncul pesan "Apakah anda yakin akan menghapus data ini?" Data di database berhasil dihapus	[√] Diterima [ ] Ditolak

4.2.2.7 Pengujian kelola EOQ

1. Tambah data EOQ

Tabel 4.21 Tambah data EOQ

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

kode barang contoh :1211BF Nama barang Contoh:Blower rectifier Tahun pemakaian Biaya simpan Biaya pesan keterangan	Data tersimpan pada database dan data bertambah, muncul pesan:"satu data eoq berhasil ditambah"	Data tersimpan pada database dan data bertambah, muncul pesan:"satu data eoq berhasil ditambah"	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Biaya simpan Contoh:dikosongkan	Menampilkan pesan kesalahan : "field biaya simpan harus diisi".	Menampilkan pesan kesalahan : "field biaya simpan harus diisi".	[√] Diterima [ ] Ditolak

2. Hapus data eoq

Tabel 4.22 Hapus data eoq

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode eoq Contoh: 22	Muncul pesan "Apakah anda yakin akan menghapus data ini?" Data di database berhasil dihapus	Muncul pesan "Apakah anda yakin akan menghapus data ini?" Data di database berhasil dihapus	[√] Diterima [ ] Ditolak

4.2.2.8 Pengujian kelola Transaksi PO

1. Tambah data Po

Tabel 4.23 Tambah data po

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode barang contoh :1211BF Nama barang Contoh:Blower Rectifier	Muncul pesan "Pembelian akan disimpan?" OK data po tersimpan pada database dan data bertambah	Muncul pesan "Pembelian akan disimpan?" OK data po tersimpan pada database dan data bertambah	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode barang contoh :1211BF Nama barang Contoh:Blower Rectifier	Muncul pesan "Pembelian akan disimpan?" CANCEL data di database data po tidak bertambah	Muncul pesan "Pembelian akan disimpan?" CANCEL data di database data po tidak bertambah	[√] Diterima [ ] Ditolak

2.a.Cari data po

Tabel 4.24 Cari data po

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode po contoh :1/PO/KDT/7/3013	Muncul data dengan kode “no.po, tgl po, kirim, pembuat, status, total”	Muncul data dengan kode “no.po, tgl po, kirim, pembuat, status, total” sesuai yang diharapkan	[√] Diterima [ ] Ditolak

2.b Cari po terlambat

Tabel 4.24 Cari data po terlambat

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode po contoh :1/PO/KDT/7/3013	Muncul data – data po yang tgl rencana kirim sudah lewat, klik “kirm” email terkirim ke supplier	Muncul data – data po yang tgl rencana kirim sudah lewat, klik “kirm” email terkirim ke supplier	[√] Diterima [ ] Ditolak

4.2.2.9 Pengujian Pengelolaan Transaksi Penerimaan

1. Tambah data penerimaan

Tabel 4.26 Tambah data Penerimaan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode po contoh :1/PO/KDT/7/2013	Data barang di database bertambah	Data barang di database bertambah	[√] Diterima [ ] Ditolak

2.Cari data penerimaan

Tabel 4.27 Cari Data Penerimaan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode penerimaan contoh :1/A14A/7/2013	Muncul Informasi: kode po, tgl, supplier, pembuat, tgl rencana kirim, tgl terima, penerima, total	Muncul Informasi: kode po, tgl, supplier, pembuat, tgl rencana kirim, tgl terima, penerima, tota	[√] Diterima [ ] Ditolak

4.2.2.10 Pengujian Pengelolaan Permintaan

1. Tambah data permintaan

Tabel 4.28 Tambah data Permintaan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode permintaan contoh :AF890 Nama barang Contoh: air filter	Data permintaan di database bertambah	Data permintaan di database bertambah	[√] Diterima [ ] Ditolak

2.Cari permintaan

Tabel 4.29 Cari Permintaan

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode permintaan contoh :1/A14A/07/2013	Muncul Informasi: kode permintaan, tgl minta, status, kode pemakaian, tgl pakai, peminta	Muncul Informasi: kode permintaan, tgl minta, status, kode pemakaian, tgl pakai, peminta	[√] Diterima [ ] Ditolak

4.2.2.11 Pengujian Pengelolaan Pemakaian

1. Tambah pemakaian

Tabel 4.30 Tambah Data Pemakaian

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode permintaan contoh :1/A14A/KDT/7/2013	Data barang di database berkurang	Data barang di database berkurang	[√] Diterima [ ] Ditolak

2.Cari pemakaian

Tabel 4.31 Cari Data Pemakaian

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kode pemakaian contoh :1/A14B/7/2013	Muncul Informasi: kode permintaan, tgl minta, peminta, kode pemakaian, tgl pakai, total, pemberi	Muncul Informasi: kode permintaan, tgl minta, peminta, kode pemakaian, tgl pakai, total, pemberi	[√] Diterima [ ] Ditolak

4.2.2.12 Pengujian input di luar range

Tabel 4.32 Pengujian input diluar Range

Kasus dan hasil uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama barang contoh :Fly wheel Harga barang contoh: 2.147.483.648	Harga barang di data base menjadi 2.147.483.647	Harga barang di data base menjadi 2.147.483.647	[√] Diterima [ ] Ditolak

4.2.3 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian Alpha (Fungsional) dengan kasus uji diatas dapat ditarik kesimpulan sebagai berikut:

1. Jika data yang dimasukkan benar, maka sistem secara fungsional akan mengeluarkan keluaran sesuai dengan harapan.
2. Jika data yang dimasukkan salah, maka sistem akan mengeluarkan pesan kesalahan dan data keluaran tidak akan sesuai harapan.

Dengan kata lain bahwa pembangunan system ini bebas dari kesalahan dan secara fungsional mengeluarkan hasil sesuai yang diharapkan.

4.2.4 Pengujian Beta

Pengujian beta dilakukan untuk mengetahui sejauhmana kualitas dari perangkat lunak yang dibangun apakah sudah sesuai dengan harapan apa belum. Untuk itu dalam pengujian beta dilakukan penelitian dengan cara memberikan kuesioner pada calon pengguna perangkat lunak yang dibangun.

Kuesioner ini terdiri dari 7 pertanyaan dengan menggunakan skala likert. Skala Likert adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survey. Nama skala ini diambil dari Rensis Likert, suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu pilihan yang tersedia, (kuesioner dapat dilihat pada lampiran).

Berdasarkan data hasil kuesioner diperoleh prosentasi masing-masing jawaban sebagai berikut :

1. Perangkat lunak ini sangat membantu dalam mengontrol stok barang

No	Keterangan	Responden
1	Sangat setuju	3
2	Setuju	2

3	Ragu-ragu	2
4	Tidak setuju	2
5	Sangat tidak setuju	1

Mengukur skor jawaban tersebut dengan cara:

Jumlah skor untuk 3 orang menjawab SS = 3x5=15

Jumlah skor untuk 2 orang menjawab ST = 2x4= 8

Jumlah skor untuk 2 orang menjawab RG = 2x3= 6

Jumlah skor untuk 2 orang menjawab TS = 2x2= 4

Jumlah skor untuk 1 orang menjawab STS = 1x1= 1

Total 34

Jumlah skor ideal untuk item no.1(skor tertinggi)=5x10=50 SS

Jumlah skor ideal untuk item no.5(skor terendah)=1x10=10 STS

Keterangan: Kriteria Interpretasi skor

Angka 0% - 20% STS

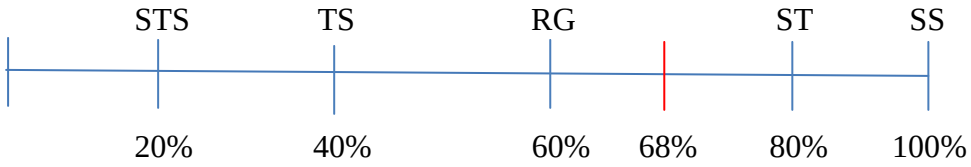
Angka 21% - 40% TS

Angka 41% - 60% RG

Angka 61% - 80% ST

Angka 81% - 100% SS

Jadi berdasarkan data yang diperoleh dari 10 responden , maka kecenderungan jawabannya adalah (34:50)x100%=68%. Hasil 68% ini jika kita analisis termasuk kategori SETUJU



2. Apakah perangkat lunak ini membantu dalam menentukan jumlah barang yang harus diorder ?

No	Keterangan	Responden
1	Sangat setuju	2
2	Setuju	3
3	Ragu-ragu	2
4	Tidak setuju	2
5	Sangat tidak setuju	1

Mengukur skor jawaban tersebut dengan cara:

Jumlah skor untuk 2 orang menjawab SS = 2x5=10

Jumlah skor untuk 3 orang menjawab ST = 3x4=12

Jumlah skor untuk 2 orang menjawab RG = 2x3= 6

Jumlah skor untuk 2 orang menjawab TS = 2x2= 4

Jumlah skor untuk 1 orang menjawab STS = 1x1= 1

Total 33

Jumlah skor ideal untuk item no.1(skor tertinggi)=5x10=50 SS

Jumlah skor ideal untuk item no.5(skor terendah)=1x10=10 STS

Keterangan: Kriteria Interpretasi skor

Angka 0% - 20% STS

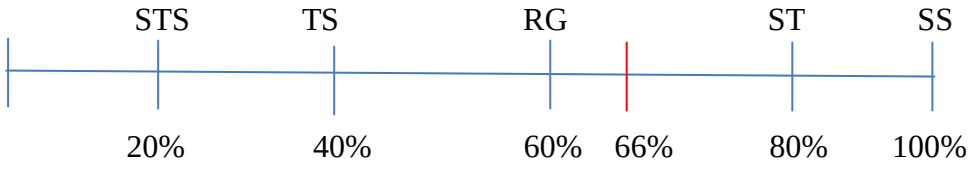
Angka 21% - 40% TS

Angka 41% - 60% RG

Angka 61% - 80% ST

Angka 81% - 100% SS

Jadi berdasarkan data yang diperoleh dari 10 responden , maka kecenderungan jawabannya adalah (33:50)x100%=66%. Hasil 66% ini jika kita analisis termasuk kategori SETUJU



3. Apakah anda setuju sistem informasi ini mempermudah menentukan waktu yang tepat dalam pengorderan barang ?

No	Keterangan	Responden
1	Sangat setuju	4
2	Setuju	3
3	Ragu-ragu	1
4	Tidak setuju	1
5	Sangat tidak setuju	1

Mengukur skor jawaban tersebut dengan cara:

Jumlah skor untuk 4 orang menjawab SS = 4x5=20

Jumlah skor untuk 3 orang menjawab ST = 3x4=12

Jumlah skor untuk 1 orang menjawab RG = 1x3= 3

Jumlah skor untuk 1 orang menjawab TS = 1x2= 2

Jumlah skor untuk 1 orang menjawab STS = 1x1= 1

Total 38

Jumlah skor ideal untuk item no.1(skor tertinggi)=5x10=50 SS

Jumlah skor ideal untuk item no.5(skor terendah)=1x10=10 STS

Keterangan: Kriteria Interpretasi skor

Angka 0% - 20% STS

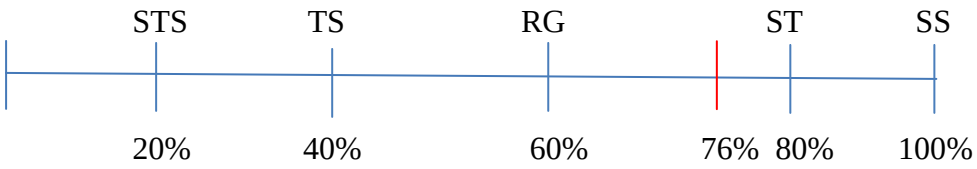
Angka 21% - 40% TS

Angka 41% - 60% RG

Angka 61% - 80% ST

Angka 81% - 100% SS

Jadi berdasarkan data yang diperoleh dari 10 responden , maka kecenderungan jawabannya adalah (38:50)x100%=76%. Hasil 76% ini jika kita analisis termasuk kategori SETUJU



4. Apakah anda setuju sistem informasi ini membantu kebutuhan LOSD selalu tersedia?

No	Keterangan	Responden
1	Sangat setuju	3
2	Setuju	3
3	Ragu-ragu	2
4	Tidak setuju	2
5	Sangat tidak setuju	0

Mengukur skor jawaban tersebut dengan cara:

Jumlah skor untuk 3 orang menjawab SS = 3x5=15

Jumlah skor untuk 3 orang menjawab ST = 3x4=12

Jumlah skor untuk 2 orang menjawab RG = 2x3= 6

Jumlah skor untuk 2 orang menjawab TS = 2x2= 4

Jumlah skor untuk 0 orang menjawab STS = 0x1= 0

Total 37

Jumlah skor ideal untuk item no.1(skor tertinggi)=5x10=50 SS

Jumlah skor ideal untuk item no.5(skor terendah)=1x10=10 STS

Keterangan: Kriteria Interpretasi skor

Angka 0% - 20% STS

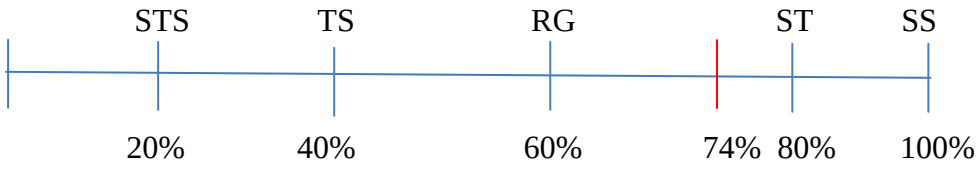
Angka 21% - 40% TS

Angka 41% - 60% RG

Angka 61% - 80% ST

Angka 81% - 100% SS

Jadi berdasarkan data yang diperoleh dari 10 responden , maka kecenderungan jawabannya adalah (37:50)x100%=74%. Hasil 74% ini jika kita analisis termasuk kategori SETUJU



5. Apakah anda setuju sistem informasi ini membantu memperlancar pergerakan barang ?

No	Keterangan	Responden
1	Sangat setuju	3
2	Setuju	1

3	Ragu-ragu	4
4	Tidak setuju	1
5	Sangat tidak setuju	1

Mengukur skor jawaban tersebut dengan cara:

Jumlah skor untuk 3 orang menjawab SS = 3x5=15

Jumlah skor untuk 1 orang menjawab ST = 1x4= 4

Jumlah skor untuk 4 orang menjawab RG = 4x3=12

Jumlah skor untuk 1 orang menjawab TS = 1x2= 2

Jumlah skor untuk 1 orang menjawab STS = 1x1= 1

Total 34

Jumlah skor ideal untuk item no.1(skor tertinggi)=5x10=50 SS

Jumlah skor ideal untuk item no.5(skor terendah)=1x10=10 STS

Keterangan: Kriteria Interpretasi skor

Angka 0% - 20% STS

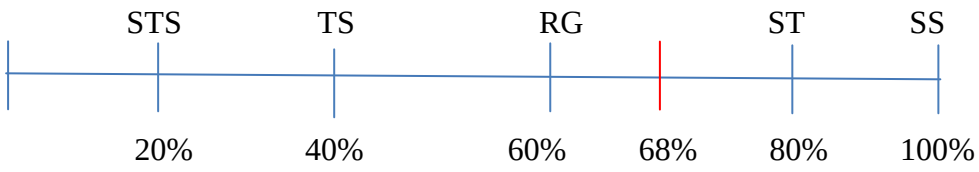
Angka 21% - 40% TS

Angka 41% - 60% RG

Angka 61% - 80% ST

Angka 81% - 100% SS

Jadi berdasarkan data yang diperoleh dari 10 responden , maka kecenderungan jawabannya adalah (34:50)x100%=68%. Hasil 68% ini jika kita analisis termasuk kategori SETUJU



6. Informasi/laporan yang dihasilkan perangkat lunak sangat akurat

No	Keterangan	Responden
1	Sangat setuju	2
2	Setuju	4
3	Ragu-ragu	3
4	Tidak setuju	0
5	Sangat tidak setuju	1

Mengukur skor jawaban tersebut dengan cara:

Jumlah skor untuk 2 orang menjawab SS = 2x5=10

Jumlah skor untuk 4 orang menjawab ST = 4x4=16

Jumlah skor untuk 3 orang menjawab RG = 3x3= 9

Jumlah skor untuk 0 orang menjawab TS = 0x2= 0

Jumlah skor untuk 1 orang menjawab STS = 1x1= 1

Total 36

Jumlah skor ideal untuk item no.1(skor tertinggi)=5x10=50 SS

Jumlah skor ideal untuk item no.5(skor terendah)=1x10=10 STS

Keterangan: Kriteria Interpretasi skor

Angka 0% - 20% STS

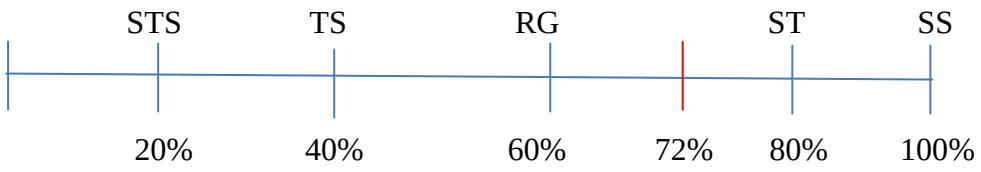
Angka 21% - 40% TS

Angka 41% - 60% RG

Angka 61% - 80% ST

Angka 81% - 100% SS

Jadi berdasarkan data yang diperoleh dari 10 responden , maka kecenderungan jawabannya adalah (36:50)x100%=72%. Hasil 72% ini jika kita analisis termasuk kategori SETUJU



7. Apakah sistem informasi ini membantu dalam hal stok opname?

No	Keterangan	Responden
1	Sangat setuju	2
2	Setuju	3
3	Ragu-ragu	3
4	Tidak setuju	1
5	Sangat tidak setuju	1

Mengukur skor jawaban tersebut dengan cara:

Jumlah skor untuk 2 orang menjawab SS = 2x5=10

Jumlah skor untuk 3 orang menjawab ST = 3x4=12

Jumlah skor untuk 3 orang menjawab CS = 3x3= 9

Jumlah skor untuk 1 orang menjawab KS = 1x2= 2

Jumlah skor untuk 1 orang menjawab TS = 1x1= 1

Total 34

Keterangan: Kriteria Interpretasi skor

Angka 0% - 20% STS

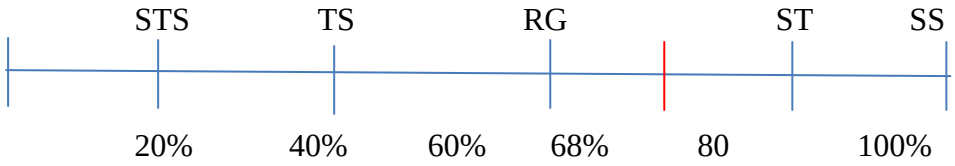
Angka 21% - 40% TS

Angka 41% - 60% RG

Angka 61% - 80% ST

Angka 81% - 100% SS

Jadi berdasarkan data yang diperoleh dari 10 responden , maka kecenderungan jawabannya adalah (34:50)x100%=68%. Hasil 68% ini jika kita analisis termasuk kategori SETUJU



4.2.5 Kesimpulan Pengujian Beta

Berdasarkan skor diatas yang didapat dari pengujian beta pengguna, maka penerapan system informasi ini sudah sesuai dengan tujuan yaitu:

1. Sistem informasi ini memberikan kemudahan kepada para pengguna dalam pengontrolan stok barang.
2. Sistem informasi dengan metode EOQ ini cukup membantu dalam menentukan jumlah barang dan waktu yang tepat untuk pemesanan barang.
3. Sistem informasi ini mempermudah bagian LOSD dalam proses permintaan barang.
4. Sistem informasi ini mempermudah bagian gudang dalam stok opname barang.