BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi sistem menjelaskan pengimplementasian dari sistem yang telah dibuat. Implementasi yang dijelaskan berupa implementasi perangkat keras, implementasi perangkat lunak, implementasi basis data dan implementasi proses.

4.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras menjelaskan perangkat keras yang digunakan untuk mengimplementasikan sistem yang telah dibuat. Pada tabel 4.1 dapat dilihat implementasi perangkat keras yang digunakan pada sistem ini:

Tabel 4.1 Implementasi Perangkat Keras

Infrastruktur	Nama Perangkat Keras	Deskripsi
Server	CPU	Intel Dual Core i3
	RAM	2 Gb
	Monitor	LCD Monitor 19”
	Harddisk	250 Gb
Client	CPU	Intel Dual Core 4500
	RAM	1 Gb
	Monitor	LCD Monitor 16,5”
	Harddisk	160 GB

4.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan perangkat lunak yang digunakan untuk mengimplementasi sistem yang telah dibuat. Pada tabel 4.2 dapat dilihat implementasi perangkat lunak yang digunakan pada sistem ini:

Tabel 4.2 Implementasi Perangkat Lunak

Infrastruktur	Nama Perangkat Lunak	Deskripsi
Server	Sistem Operasi	Win Xp
	Code Editor	PHP Designer, Notepad++
	Bahasa Pemrograman	PHP
	Browser	Mozila, Opera, Chrome
	DBMS	MySql 5.5.8
Client	Sistem operasi	Win Xp
	Browser	Mozila, Opera, Chrome
Pendukung komunikasi data		Jaringan LAN (Local Area Network)

4.1.3 Implementasi Basis Data (*Database*)

Implementasi basis data menjelaskan implementasi dari basis data yang dibuat pada sistem ini, meliputi implementasi proses pembuatan *database* dan *table*. Pada tabel 4.3 dapat dilihat implementasi basis data pada sistem ini:

Tabel 0.3 Implementasi basis data

Pembuatan *database*	CREATE DATABASE db_mspare;
Pembuatan m_departemen	CREATE TABLE IF NOT EXISTS `m_departemen` (`kd_departemen` int(11) NOT NULL AUTO_INCREMENT, `nama_departemen` varchar(30) DEFAULT NULL,

	PRIMARY KEY (`kd_departemen`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;
Pembuatan Tabel m_grup	CREATE TABLE IF NOT EXISTS `m_grup` (`id_grup` int(11) NOT NULL AUTO_INCREMENT, `nama_grup` varchar(30) DEFAULT NULL, PRIMARY KEY (`id_grup`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
Pembuatan Tabel m_jnssparepart	CREATE TABLE IF NOT EXISTS `m_jnssparepart` (`kd_jns_sparepart` int(11) NOT NULL AUTO_INCREMENT, `nama_jns_sparepart` varchar(20) DEFAULT NULL, PRIMARY KEY (`kd_jns_sparepart`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
Pembuatan Tabel m_jnswarehouse	CREATE TABLE IF NOT EXISTS `m_jnswarehouse` (`kd_jnsWarehouse` int(11) NOT NULL AUTO_INCREMENT, `nama_jenis_warehouse` varchar(30) DEFAULT NULL, PRIMARY KEY (`kd_jnsWarehouse`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
Pembuatan Tabel m_sparepart	CREATE TABLE IF NOT EXISTS `m_sparepart` (`kd_sparepart` int(11) NOT NULL AUTO_INCREMENT, `no_sparepart` varchar(30) DEFAULT NULL, `nama_sparepart` varchar(30) DEFAULT NULL, `kd_jns_sparepart` int(11) DEFAULT NULL, `stock` int(11) DEFAULT NULL,

```
  `keluar` int(11) NOT NULL,
  `harga` double DEFAULT NULL,
  `kd_vendor` int(11) DEFAULT NULL,
  `kd_warehouse` int(11) DEFAULT NULL,
  `status` int(11) DEFAULT NULL,
  PRIMARY KEY (`kd_sparepart`),
  KEY `kd_jns_sparepart` (`kd_jns_sparepart`),
  KEY `kd_vendor` (`kd_vendor`),
  KEY `kd_warehouse` (`kd_warehouse`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=7 ;
--
-- RELATIONS FOR TABLE `m_sparepart`:
--   `kd_warehouse`
--       `m_warehouse` -> `kd_warehouse`
--   `kd_jns_sparepart`
--       `m_jnssparepart` -> `kd_jns_sparepart`
--   `kd_vendor`
--       `m_vendor` -> `kd_vendor`
--
-- Constraints for dumped tables
-- Constraints for table `m_sparepart`
--
ALTER TABLE `m_sparepart`
  ADD CONSTRAINT `m_sparepart_ibfk_3` FOREIGN
KEY (`kd_warehouse`) REFERENCES `m_warehouse`
(`kd_warehouse`),
  ADD CONSTRAINT `m_sparepart_ibfk_1` FOREIGN
KEY (`kd_jns_sparepart`) REFERENCES
`m_jnssparepart` (`kd_jns_sparepart`),
  ADD CONSTRAINT `m_sparepart_ibfk_2` FOREIGN
```

	KEY (`kd_vendor`) REFERENCES `m_vendor` (`kd_vendor`);
Pembuatan Tabel m_vendor	CREATE TABLE IF NOT EXISTS `m_vendor` (`kd_vendor` int(11) NOT NULL AUTO_INCREMENT, `nama_vendor` varchar(30) DEFAULT NULL, PRIMARY KEY (`kd_vendor`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
Pembuatan Tabel m_warehouse	CREATE TABLE IF NOT EXISTS `m_warehouse` (`kd_warehouse` int(11) NOT NULL AUTO_INCREMENT, `nama_warehouse` varchar(20) DEFAULT NULL, `kd_jnsWarehouse` int(20) DEFAULT NULL, PRIMARY KEY (`kd_warehouse`), KEY `kd_jnsWarehouse` (`kd_jnsWarehouse`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ; -- -- RELATIONS FOR TABLE `m_warehouse`: -- `kd_jnsWarehouse` -- `m_jnswarehouse` -> `kd_jnsWarehouse` -- -- Constraints for dumped tables -- Constraints for table `m_warehouse` -- ALTER TABLE `m_warehouse` ADD CONSTRAINT `m_warehouse_ibfk_1` FOREIGN KEY (`kd_jnsWarehouse`) REFERENCES `m_jnswarehouse` (`kd_jnsWarehouse`);
Pembuatan Tabel	CREATE TABLE IF NOT EXISTS `t_issue` (

t_issue	`kd_issue` int(11) NOT NULL AUTO_INCREMENT, `kd_req` int(11) NOT NULL, `no_issue` varchar(30) DEFAULT NULL, `tgl_issue` date NOT NULL, `id_user` int(11) DEFAULT NULL, `kd_departemen` int(11) DEFAULT NULL, `status_issue` varchar(13) NOT NULL DEFAULT 'Baru', PRIMARY KEY (`kd_issue`), KEY `kd_departemen` (`kd_departemen`), KEY `id_user` (`id_user`), KEY `kd_req` (`kd_req`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=136 ; -- -- RELATIONS FOR TABLE `t_issue`: -- `kd_departemen` -- `m_departemen` -> `kd_departemen` -- `kd_req` -- `t_request` -> `kd_req` -- `id_user` -- `user` -> `id_user` -- -- Constraints for dumped tables -- Constraints for table `t_issue` -- ALTER TABLE `t_issue` ADD CONSTRAINT `t_issue_ibfk_3` FOREIGN KEY (`kd_departemen`) REFERENCES `m_departemen` (`kd_departemen`), ADD CONSTRAINT `t_issue_ibfk_1` FOREIGN KEY (`kd_req`) REFERENCES `t_request` (`kd_req`),

	ADD CONSTRAINT `t_issue_ibfk_2` FOREIGN KEY (`id_user`) REFERENCES `user` (`id_user`);
Pembuatan Tabel t_issue_detail	CREATE TABLE IF NOT EXISTS `t_issue_detail` (`id` int(11) NOT NULL AUTO_INCREMENT, `kd_issue` int(11) DEFAULT NULL, `kd_sparepart` int(11) DEFAULT NULL, `qty_order` int(11) DEFAULT NULL, PRIMARY KEY (`id`), KEY `kd_issue` (`kd_issue`), KEY `kd_sparepart` (`kd_sparepart`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=136 ; -- -- RELATIONS FOR TABLE `t_issue_detail`: -- `kd_sparepart` -- `m_sparepart` -> `kd_sparepart` -- `kd_issue` -- `t_issue` -> `kd_issue` -- Constraints for dumped tables -- Constraints for table `t_issue_detail` -- ALTER TABLE `t_issue_detail` ADD CONSTRAINT `t_issue_detail_ibfk_2` FOREIGN KEY (`kd_sparepart`) REFERENCES `m_sparepart` (`kd_sparepart`), ADD CONSTRAINT `t_issue_detail_ibfk_1` FOREIGN KEY (`kd_issue`) REFERENCES `t_issue` (`kd_issue`);
Pembuatan Tabel t_po	CREATE TABLE IF NOT EXISTS `t_po` (`kd_po` int(11) NOT NULL AUTO_INCREMENT, `no_po` varchar(30) NOT NULL,

```
  `tgl_po` datetime DEFAULT NULL,
  `id_user` int(11) NOT NULL,
  `kd_departemen` int(11) DEFAULT NULL,
  `kd_pr` int(11) DEFAULT NULL,
  `kd_vendor` int(11) NOT NULL,
  `status_po` varchar(12) CHARACTER SET latin1
COLLATE latin1_general_ci NOT NULL DEFAULT 'PO
BARU',
  PRIMARY KEY (`kd_po`),
  KEY `kd_vendor` (`kd_vendor`),
  KEY `kd_pr` (`kd_pr`),
  KEY `kd_departemen` (`kd_departemen`),
  KEY `id_user` (`id_user`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=6 ;
--
-- RELATIONS FOR TABLE `t_po`:
--   `kd_vendor`
--      `m_vendor` -> `kd_vendor`
--   `id_user`
--      `user` -> `id_user`
--   `kd_departemen`
--      `m_departemen` -> `kd_departemen`
--   `kd_pr`
--      `t_pr` -> `kd_pr`
--
-- Constraints for dumped tables
-- Constraints for table `t_po`
--
ALTER TABLE `t_po`
  ADD CONSTRAINT `t_po_ibfk_4` FOREIGN KEY
```

	(`kd_vendor`) REFERENCES `m_vendor` (`kd_vendor`), ADD CONSTRAINT `t_po_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `user` (`id_user`), ADD CONSTRAINT `t_po_ibfk_2` FOREIGN KEY (`kd_departemen`) REFERENCES `m_departemen` (`kd_departemen`), ADD CONSTRAINT `t_po_ibfk_3` FOREIGN KEY (`kd_pr`) REFERENCES `t_pr` (`kd_pr`);
Pembuatan Tabel t_po_detail	CREATE TABLE IF NOT EXISTS `t_po_detail` (`id` int(11) NOT NULL AUTO_INCREMENT, `kd_po` int(11) DEFAULT NULL, `kd_sparepart` int(11) DEFAULT NULL, `qty` int(4) NOT NULL, `harga_po` int(11) NOT NULL, PRIMARY KEY (`id`), KEY `kd_po` (`kd_po`), KEY `kd_sparepart` (`kd_sparepart`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ; -- -- RELATIONS FOR TABLE `t_po_detail`: -- `kd_sparepart` -- `m_sparepart` -> `kd_sparepart` -- `kd_po` -- `t_po` -> `kd_po` -- -- Constraints for dumped tables -- -- Constraints for table `t_po_detail` --

	ALTER TABLE `t_po_detail` ADD CONSTRAINT `t_po_detail_ibfk_2` FOREIGN KEY (`kd_sparepart`) REFERENCES `m_sparepart` (`kd_sparepart`), ADD CONSTRAINT `t_po_detail_ibfk_1` FOREIGN KEY (`kd_po`) REFERENCES `t_po` (`kd_po`);
Pembuatan Tabel t_pr	CREATE TABLE IF NOT EXISTS `t_pr` (`kd_pr` int(11) NOT NULL AUTO_INCREMENT, `no_pr` varchar(12) NOT NULL, `id_user` int(3) NOT NULL, `kd_departemen` int(11) NOT NULL, `tgl_pr` datetime NOT NULL, `status_pr` varchar(30) CHARACTER SET latin1 COLLATE latin1_general_ci NOT NULL DEFAULT 'PR BARU', `keterangan` varchar(500) NOT NULL, PRIMARY KEY (`kd_pr`), KEY `id_user` (`id_user`), KEY `kd_departemen` (`kd_departemen`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ; -- -- RELATIONS FOR TABLE `t_pr`: -- `kd_departemen` -- `m_departemen` -> `kd_departemen` -- `id_user` -- `user` -> `id_user` -- Constraints for dumped tables -- Constraints for table `t_pr` --

	ALTER TABLE `t_pr` ADD CONSTRAINT `t_pr_ibfk_2` FOREIGN KEY (`kd_departemen`) REFERENCES `m_departemen` (`kd_departemen`), ADD CONSTRAINT `t_pr_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `user` (`id_user`);
Pembuatan Tabel t_pr_detail	CREATE TABLE IF NOT EXISTS `t_pr_detail` (`id` int(11) NOT NULL AUTO_INCREMENT, `kd_pr` int(11) NOT NULL, `kd_sparepart` int(11) NOT NULL, `qty` int(4) NOT NULL, PRIMARY KEY (`id`), KEY `kd_pr` (`kd_pr`), KEY `kd_sparepart` (`kd_sparepart`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ; -- -- RELATIONS FOR TABLE `t_pr_detail`: -- `kd_sparepart` -- `m_sparepart` -> `kd_sparepart` -- `kd_pr` -- `t_pr` -> `kd_pr` -- Constraints for dumped tables -- Constraints for table `t_pr_detail` -- ALTER TABLE `t_pr_detail` ADD CONSTRAINT `t_pr_detail_ibfk_2` FOREIGN KEY (`kd_sparepart`) REFERENCES `m_sparepart` (`kd_sparepart`),

	ADD CONSTRAINT `t_pr_detail_ibfk_1` FOREIGN KEY (`kd_pr`) REFERENCES `t_pr` (`kd_pr`);
Pembuatan Tabel t_receipt	CREATE TABLE IF NOT EXISTS `t_receipt` (`kd_receipt` int(11) NOT NULL AUTO_INCREMENT, `no_receipt` varchar(30) DEFAULT NULL, `tgl_receipt` datetime DEFAULT NULL, `id_user` int(11) DEFAULT NULL, `kd_departemen` int(11) DEFAULT NULL, `kd_po` int(11) DEFAULT NULL, `kd_vendor` int(11) NOT NULL, `status_receipt` varchar(30) NOT NULL DEFAULT 'NO RECEIVED', PRIMARY KEY (`kd_receipt`), KEY `kd_vendor` (`kd_vendor`), KEY `kd_po` (`kd_po`), KEY `kd_departemen` (`kd_departemen`), KEY `id_user` (`id_user`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ; -- -- RELATIONS FOR TABLE `t_receipt`: -- `kd_vendor` -- `m_vendor` -> `kd_vendor` -- `id_user` -- `user` -> `id_user` -- `kd_departemen` -- `m_departemen` -> `kd_departemen` -- `kd_po` -- `t_po` -> `kd_po` -- Constraints for dumped tables

	-- Constraints for table `t_receipt` -- ALTER TABLE `t_receipt` ADD CONSTRAINT `t_receipt_ibfk_4` FOREIGN KEY (`kd_vendor`) REFERENCES `m_vendor` (`kd_vendor`), ADD CONSTRAINT `t_receipt_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `user` (`id_user`), ADD CONSTRAINT `t_receipt_ibfk_2` FOREIGN KEY (`kd_departemen`) REFERENCES `m_departemen` (`kd_departemen`), ADD CONSTRAINT `t_receipt_ibfk_3` FOREIGN KEY (`kd_po`) REFERENCES `t_po` (`kd_po`);
Pembuatan Tabel t_receipt_detail	CREATE TABLE IF NOT EXISTS `t_receipt_detail` (`id` int(11) NOT NULL AUTO_INCREMENT, `kd_receipt` int(11) DEFAULT NULL, `kd_sparepart` int(11) DEFAULT NULL, `qty_receipt` int(11) DEFAULT NULL, `harga_receipt` int(11) NOT NULL, PRIMARY KEY (`id`), KEY `kd_receipt` (`kd_receipt`), KEY `kd_sparepart` (`kd_sparepart`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ; -- -- RELATIONS FOR TABLE `t_receipt_detail`: -- `kd_sparepart` -- `m_sparepart` -> `kd_sparepart` -- `kd_receipt` -- `t_receipt` -> `kd_receipt` -- Constraints for dumped tables

	-- Constraints for table `t_receipt_detail` -- ALTER TABLE `t_receipt_detail` ADD CONSTRAINT `t_receipt_detail_ibfk_2` FOREIGN KEY (`kd_sparepart`) REFERENCES `m_sparepart` (`kd_sparepart`), ADD CONSTRAINT `t_receipt_detail_ibfk_1` FOREIGN KEY (`kd_receipt`) REFERENCES `t_receipt` (`kd_receipt`);
Pembuatan Tabel t_request	CREATE TABLE IF NOT EXISTS `t_request` (`kd_req` int(11) NOT NULL AUTO_INCREMENT, `no_req` varchar(30) DEFAULT NULL, `tgl` datetime DEFAULT NULL, `id_user` int(11) DEFAULT NULL, `kd_departemen` int(11) DEFAULT NULL, `status` varchar(15) CHARACTER SET latin1 COLLATE latin1_general_ci NOT NULL DEFAULT 'Baru', PRIMARY KEY (`kd_req`), KEY `kd_departemen` (`kd_departemen`), KEY `id_user` (`id_user`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=136 ; -- -- RELATIONS FOR TABLE `t_request`: -- `kd_departemen` -- `m_departemen` -> `kd_departemen` -- `id_user` -- `user` -> `id_user`

	-- Constraints for dumped tables -- Constraints for table `t_request` -- ALTER TABLE `t_request` ADD CONSTRAINT `t_request_ibfk_2` FOREIGN KEY (`kd_departemen`) REFERENCES `m_departemen` (`kd_departemen`), ADD CONSTRAINT `t_request_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `user` (`id_user`);
Pembuatan Tabel t_request_detail	CREATE TABLE IF NOT EXISTS `t_request_detail` (`id` int(11) NOT NULL AUTO_INCREMENT, `kd_req` int(11) DEFAULT NULL, `kd_sparepart_out` int(11) DEFAULT NULL, `qty_order` int(11) DEFAULT NULL, PRIMARY KEY (`id`), KEY `kd_req` (`kd_req`), KEY `kd_sparepart_out` (`kd_sparepart_out`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=136 ; -- RELATIONS FOR TABLE `t_request_detail`: -- `kd_sparepart_out` -- `m_sparepart` -> `kd_sparepart` -- `kd_req` -- `t_request` -> `kd_req` -- Constraints for dumped tables -- Constraints for table `t_request_detail` -- ALTER TABLE `t_request_detail` ADD CONSTRAINT `t_request_detail_ibfk_2` FOREIGN KEY (`kd_sparepart_out`) REFERENCES

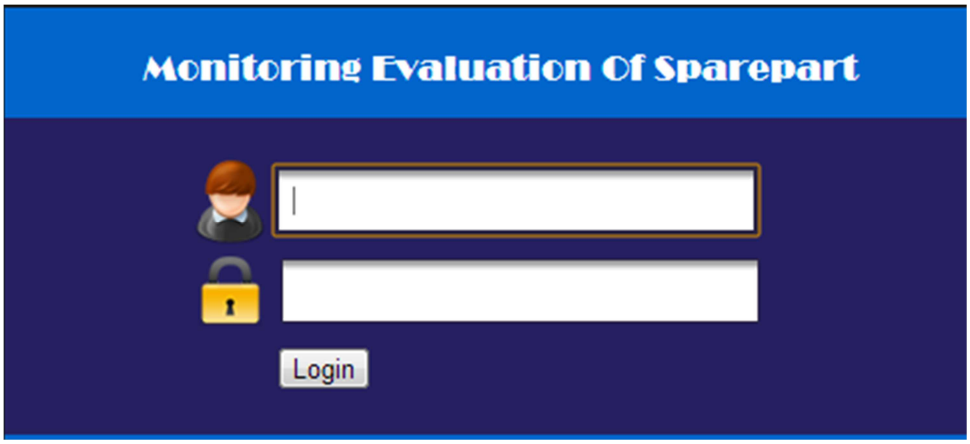
	`m_sparepart` (`kd_sparepart`), ADD CONSTRAINT `t_request_detail_ibfk_1` FOREIGN KEY (`kd_req`) REFERENCES `t_request` (`kd_req`);
Pembuatan Tabel user	CREATE TABLE IF NOT EXISTS `user` (`username` varchar(12) NOT NULL, `id_user` int(3) NOT NULL AUTO_INCREMENT, `password` varchar(32) NOT NULL, `nama` varchar(30) NOT NULL, `id_grup` int(11) NOT NULL, PRIMARY KEY (`id_user`), KEY `id_grup` (`id_grup`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ; -- -- RELATIONS FOR TABLE `user`: -- `id_grup` -- `m_grup` -> `id_grup` -- Constraints for dumped tables -- Constraints for table `user` -- ALTER TABLE `user` ADD CONSTRAINT `user_ibfk_1` FOREIGN KEY (`id_grup`) REFERENCES `m_grup` (`id_grup`);

4.1.4 Implementasi Proses

Implementasi proses menjelaskan dan menggambarkan implementasi dari setiap proses yang ada didalam sistem ini.

1. Tampilan Login

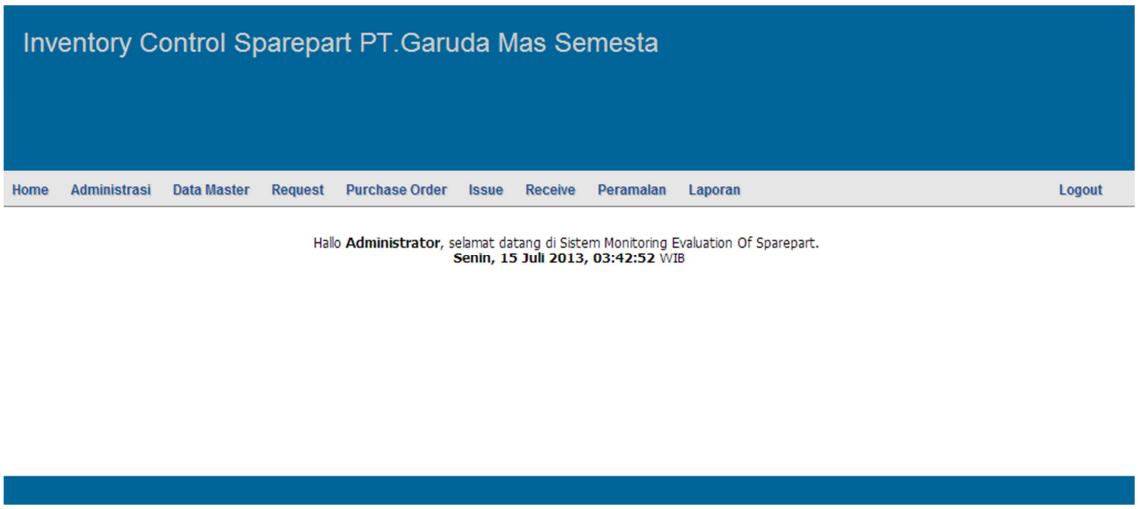
   Tampilan login merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan. Tampilan Login dapat dilihat pada Gambar 4.1.



Gambar 0.1 Tampilan Login

2. Tampilan Utama

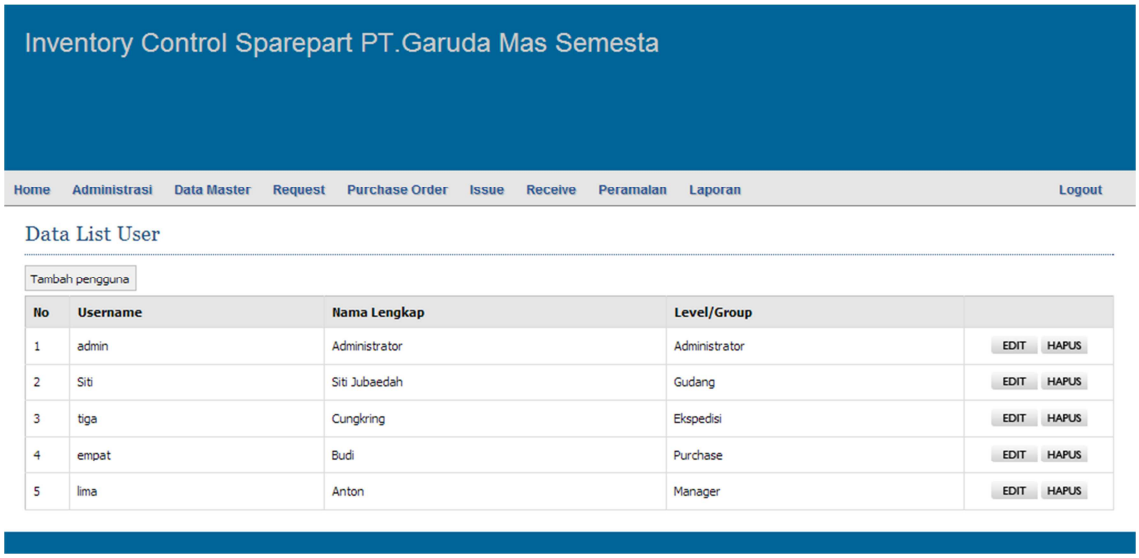
   Tampilan utama menampilkan menu-menu yang dapat diakses oleh user. Tampilan utama dapat dilihat pada Gambar 4.2.



Gambar 0.2 Tampilan Utama

3. Tampilan Data Pengguna

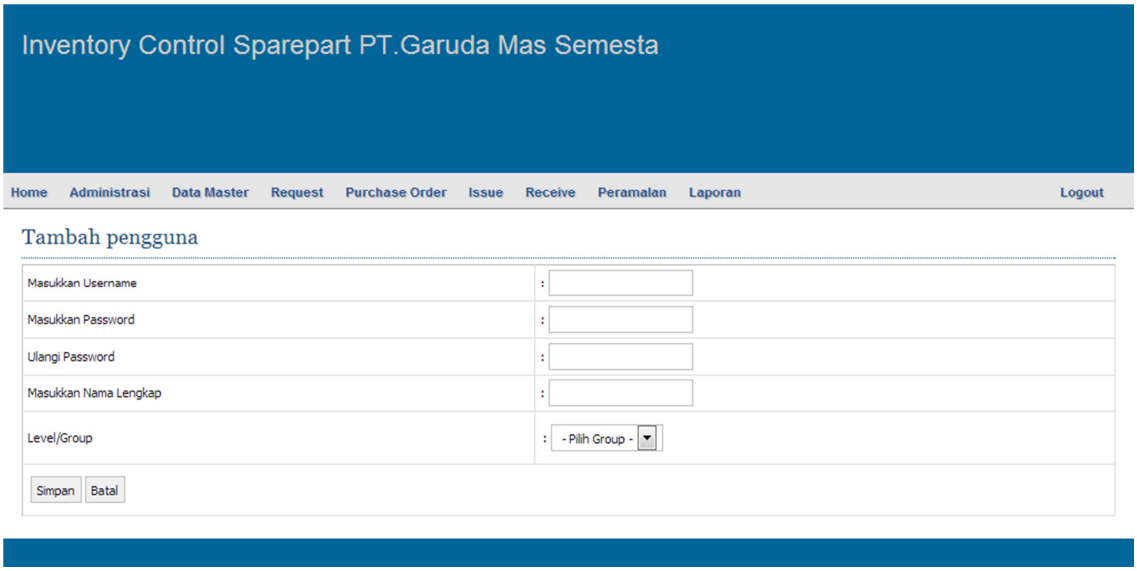
   Tampilan Data Pengguna menampilkan data siapa saja yang dapat mengkases aplikasi pengendalian *sparepart* yang tersimpan di *database*. Tampilan data pengguna dapat dilihat pada Gambar 4.3.



Gambar 0.3 Tampilan Data Pengguna

4. Tampilan Tambah Pengguna

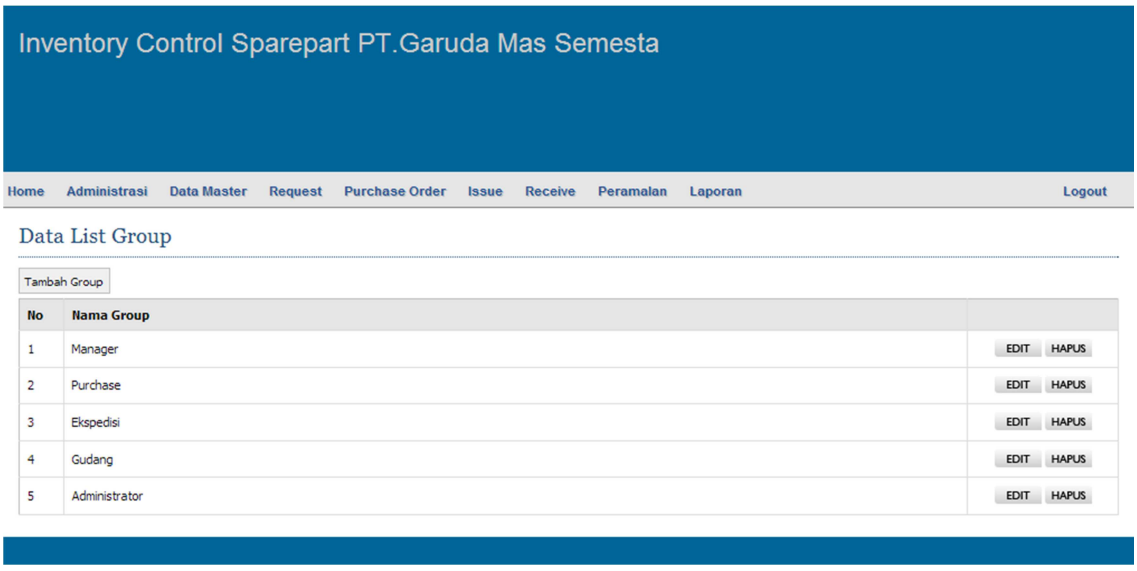
   Tampilan Tambah Pengguna menampilkan tampilan untuk menambah pengguna aplikasi yang baru. Tampilan Tambah Pengguna dapat dilihat pada Gambar 4.4.



Gambar 0.4 Tampilan Tambah Pengguna

5. Tampilan Data Group

Tampilan Data Group menampilkan data Group apa saja yang dapat mengkases aplikasi pengendalian *sparepart* yang tersimpan di *database*. Tampilan data group dapat dilihat pada Gambar 4.5



Gambar 0.5 Tampilan Data Group

6. Tampilan Tambah Group

jiTampilan Tambah Group menampilkan tampilan untuk menambah group aplikasi yang baru. Tampilan Tambah Pengguna dapat dilihat pada Gambar 4.6.



Gambar 0.6 Tampilan Tambah Group

7. Tampilan Jenis *Warehouse*

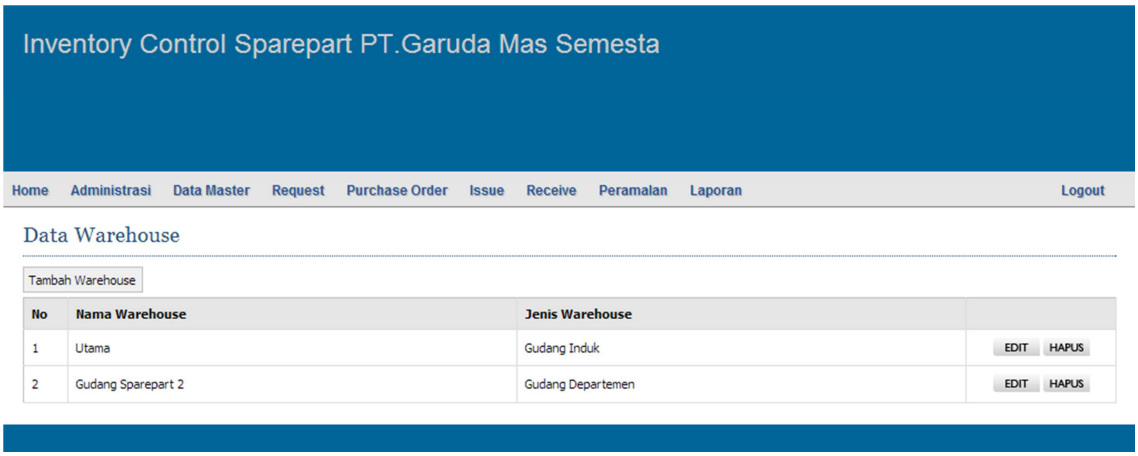
Tampilan jenis *warehouse* menampilkan tampilan jenis-jenis *warehaouse.* Tampilan Jenis *Warehouse* dapat dilihat pada Gambar 4.7.



Gambar 0.7 Tampilan *Report* Jenis *Warehouse*

8. Tampilan *Warehouse*

Tampilan *Warehouse* menampilkan tampilan *warehaouse* yang ada pada aplikasi. Tampilan *warehouse* dapat dilihat pada Gambar 4.8



Gambar 0.8 Tampilan *Warehouse*

9. Tampilan Jenis *Sparepart*

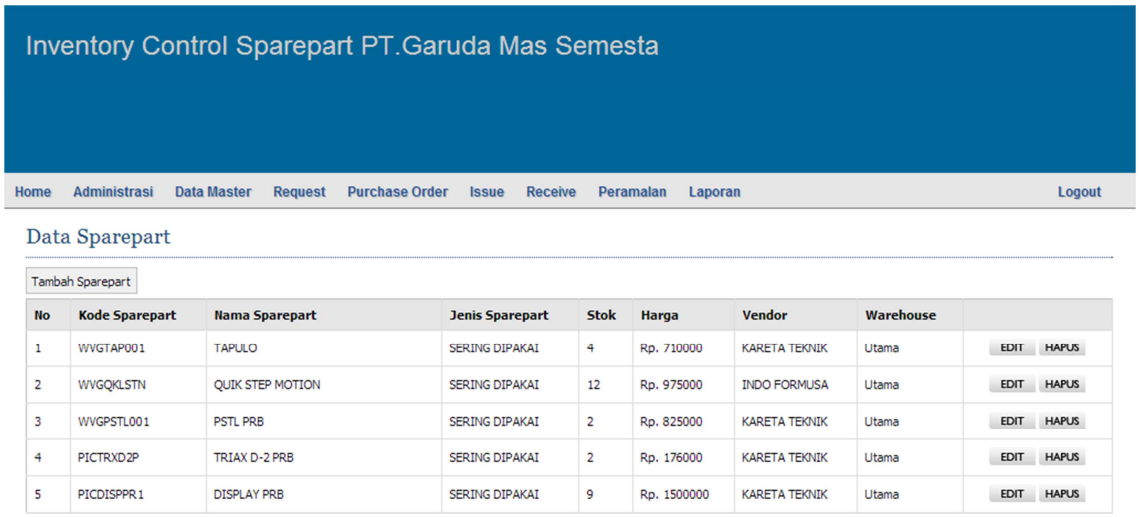
Tampilan jenis *sparepart* menampilkan tampilan jenis *sparepart* yang tersimpan di media penyimpanan. Tampilan jenis *sparepart* dapat dilihat pada Gambar 4.9



Gambar 0.9 Tampilan Jenis *Sparepart*

10. Tampilan *Sparepart*

Tampilan *Sparepart* menampilkan tampilan *Sparepart* yang ada di media penyimpanan. Tampilan *Sparepart* dapat dilihat pada Gambar 4.10



Gambar 0.10 Tampilan *Sparepart*

11. Tampilan Data *Vendor*

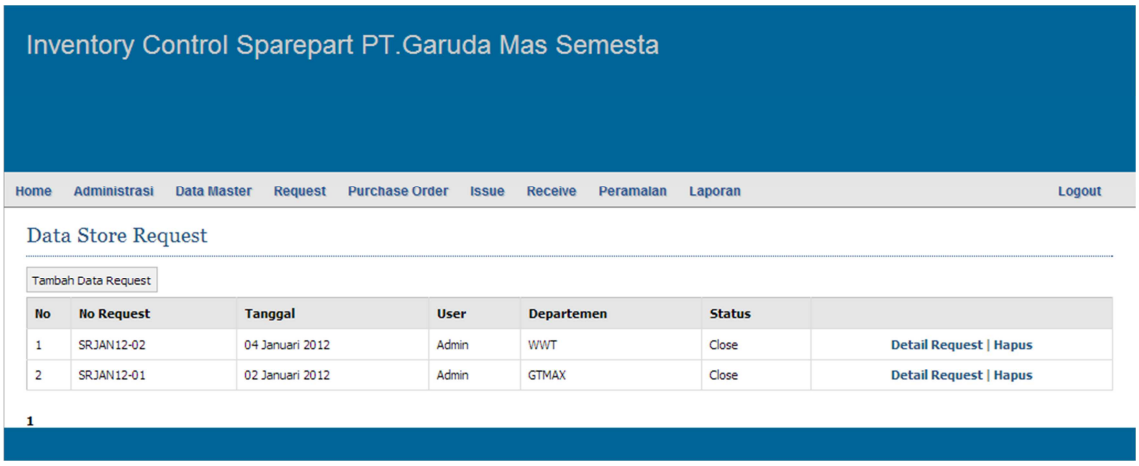
Tampilan Data *Vendor* menampilkan tampilan data *Vendor* yang ada di media penyimpanan. Tampilan Data *Vendor* dapat dilihat pada Gambar 4.11



Gambar 0.11 Tampilan Data *Vendor*

12. Tampilan *Store Request*

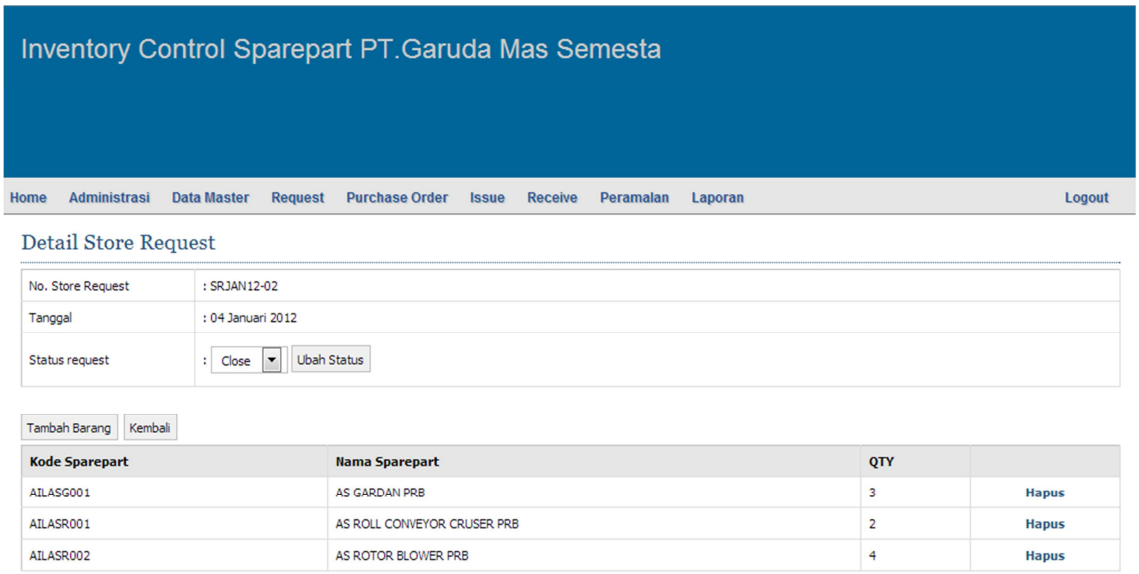
Tampilan *Store Request* menampilkan tampilan permintaan barang. Tampilan *Store Request* dapat dilihat pada Gambar 4.12



Gambar 0.12 Tampilan *Store Request*

13. Tampilan *Store Request Detail*

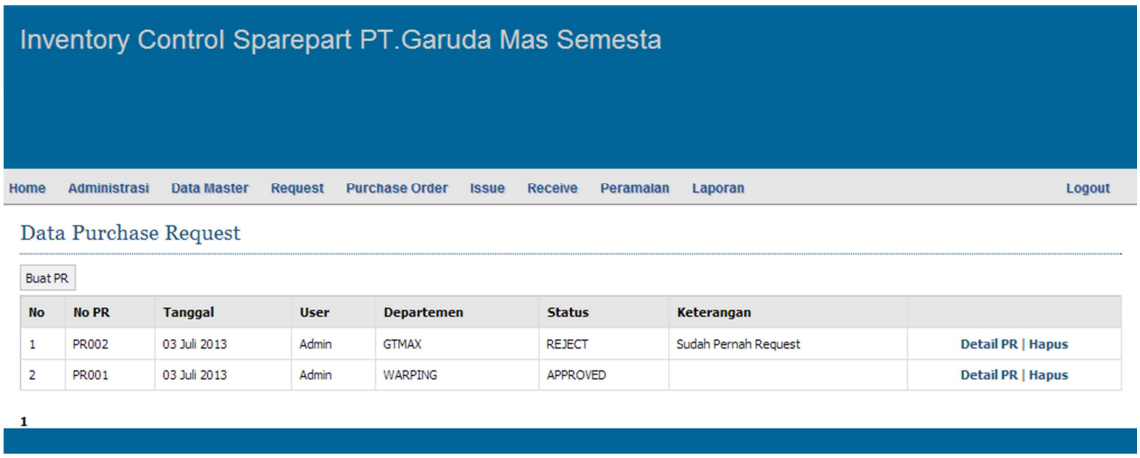
Tampilan *Store Request Detail* menampilkan tampilan daftar permintaan barang yang diminta ke bagian gudang . Tampilan *Store Request Detail* dapat dilihat pada Gambar 4.13



Gambar 0.13 Tampilan *Store Request Detail*

14. Tampilan *Purchase Request*

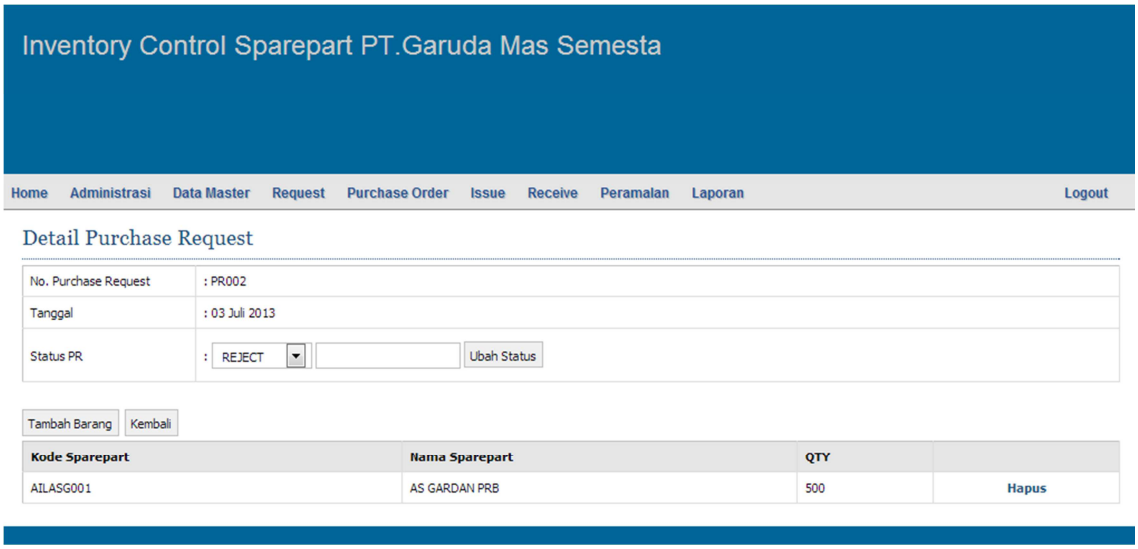
Tampilan *Purchase Request* menampilkan tampilan permintaan pembelian. Tampilan *Purchase Request* dapat dilihat pada Gambar 4.14



Gambar 0.14 Tampilan *Purchase Request*

15. Tampilan *Detail Purchase Request*

Tampilan *Detail Purchase Request* menampilkan tampilan daftar permintaan pembelian. Tampilan *Detail Purchase Request* dapat dilihat pada Gambar 4.15



Gambar 0.15 Tampilan *Detail Purchase Request*

16. Tampilan *Purchase Order*

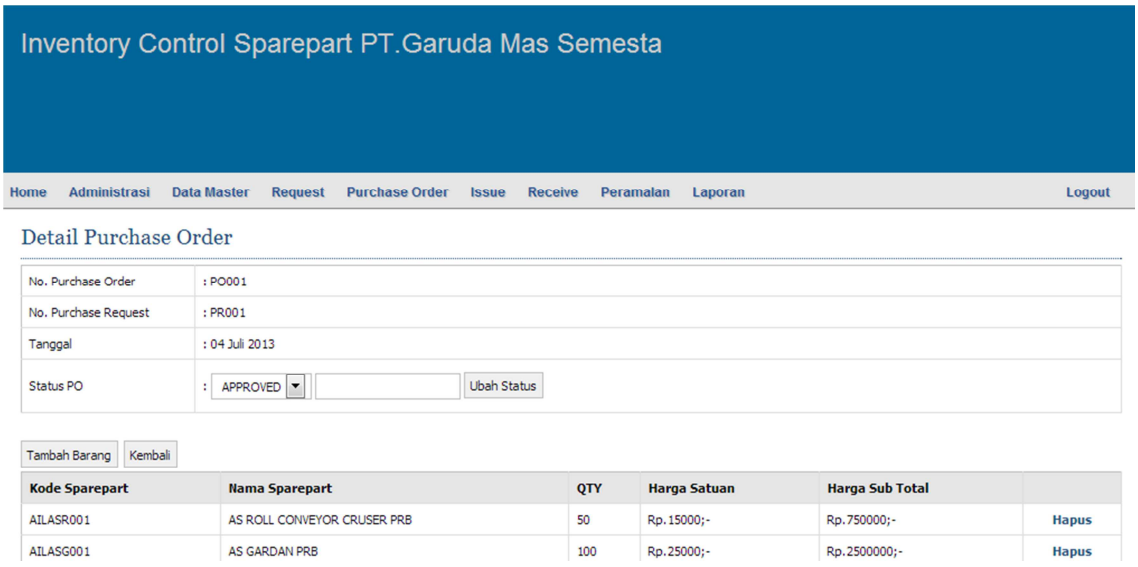
Tampilan *Purchase Order* menampilkan tampilan PO. Tampilan *Purchase Order* dapat dilihat pada Gambar 4.16



Gambar 0.16 Tampilan *Purchase Order*

17. Tampilan *Detail Purchase Order*

Tampilan *Detail Purchase Order* menampilkan tampilan untuk daftar *Purchase Order* yang tersimpan di media penyimpanan. Tampilan *Detail Purchase Order* dapat dilihat pada Gambar 4.17



Gambar 0.17 Tampilan *Detail Purchase Order*

4.2 Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan-kesalahan dan kekurangan-kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

4.3 Pengujian Alpha

Pengujian alpha merupakan pengujian yang menitikberatkan pada hasil *ouput* dari kendali input yang dimasukan pada tampilan kendali *input* form tampilan, dimana pengujian dikatakan berhasil apabila *output* sesuai dengan kendali *input* yang dimasukan pada tiap tampilan.

4.3.1 Rencana Pengujian

Rencana pengujian Sistem Informasi pengendalian persediaan *sparepart* menggunakan pengujian *black box.* Pengujian *black box* digunakan untuk menguji fungsional dari sistem yang dibuat. Rencana pengujian dapat dilihat pada tabel 4.4 yang akan dijalankan pada sistem ini:

Tabel 0.4 Rencana Pengujian Aplikasi

Pengujian	Butir Uji	Jenis Pengujian
Login	Proses login	*Black box*
Pengolahan Data Pengguna	1. Tambah Data Master 2. Hapus Data Master 3. Edit Data Master	*Black box*
Pengolahan Data Group	1. Tambah Data Group 2. Hapus Data Group 3. Edit Data Group	*Black box*

Pengolahan Data Jenis Warehouse	1. Tambah Data Jenis Warehouse 2. Hapus Data Jenis Warehouse 3. Edit Data Jenis Warehouse	*Black box*
Pengolahan Data Warehouse	1. Tambah Data Jenis Warehouse 2. Hapus Data Jenis Warehouse 3. Edit Data Jenis Warehouse	*Black box*
Pengolahan Data Jenis Sparepart	1. Tambah Data Jenis Sparepart 2. Hapus Data Jenis Sparepart 3. Edit Data Jenis Sparepart	*Black box*
Pengolahan Data Sparepart	1. Tambah Data Sparepart 2. Hapus Data Sparepart 3. Edit Data Sparepart	*Black box*
Pengolahan Data Vendor	1. Tambah Data Vendor 2. Hapus Data Vendor 3. Edit Data Vendor	*Black box*
Pengolahan Data Store Request	1. Tambah Data Store Request 2. Hapus Data Store Request	*Black box*
Pengolahan Data Detail Store Request	1. Tambah Data Detail Store Request 2. Hapus Data Detail Store Request 3. Ubah Status Data Store Request	*Black box*
Pengolahan Data Purchase Request	1. Tambah Data Purchase Request 2. Hapus Data Purchase Request	*Black box*
Pengolahan Data Detail Purchase Request	1. Tambah Data Detail Purchase Request 2. Hapus Data Detail Purchase Request 3. Ubah Status Data Purchase Request	*Black box*
Pengolahan Data Purchase Order	1. Tambah Data Purchase Order 2. Hapus Data Purchase Order	*Black box*
Pengolahan Data Detail Purchase Order	1. Tambah Data Detail Purchase Order 2. Hapus Data Detail Purchase Order 3. Ubah Status Data Purchase Order	*Black box*

Pengolahan Data Issue	1. Tambah Data Issue 2. Hapus Data Issue	*Black box*
Pengolahan Detail Data Issue	1. Tambah Data Detail Issue 2. Hapus Data Detail Issue 3. Ubah Status Data Issue	*Black box*
Pengolahan Data Receive	1. Tambah Data Receive 2. Hapus Data Receive	*Black box*
Pengolahan Detail Data Receive	1. Tambah Data Detail Receive 2. Hapus Detail Data Receive 3. Ubah Status Data Receive	*Black box*
Pengolahan Data Peramalan	1. Perhitungan Peramalan 2. Perhitungan Kesalahan	*Black box*
Cetak Laporan	1. Cetak Laporan Request 2. Cetak Laporan PO 3. Cetak Laporan Receive 4. Cetak Laporan Stock 5. Cetak Laporan Issue	*Black box*

4.3.2 Hasil Pengujian

Pengujian dilakukan sesuai dengan rencana pengujian dengan melakukan pengujian dengan data masukan yang benar dan salah.

4.3.2.1 Login

Pada Tabel 4.5 dapat dilihat butir pengujian pada proses Login adalah sebagai berikut:

Tabel 0.5 Pengujian data normal proses Login

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
1. IdPetugas:admin 2. Password:admin123	1. Nama Petugas tampil sesuai dengan IdPetugas 2. Autentikasi petugas berhasil Ketika tombol login di klik	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
1. IdPetugas: admin 2. Password: admin	1. Login ditolak 2. Autentikasi petugas tidak berhasil Ketika tombol login di klik 3. Message box muncul (pesan kesalahan)	Sesuai yang diharapkan	[X] diterima [ ] ditolak

4.3.2.2 Menu Data Administrasi

Pengolahan Data Administrasi

Pada tabel 4.6 dapat dilihat butir pengujian pada proses data Administrasi adalah sebagai berikut:

Tabel 0.6 Pengujian data Administrasi

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data Pengguna dan *Group*	Klik menu data Administrasi maka akan muncul sub menu yang berisi pengolahan data perngguna dan data *group*	Sesuai yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.7 dapat dilihat butir pengujian pada proses data Pengguna adalah sebagai berikut:

Tabel 0.7 Pengujian data Pengguna

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data Pengguna yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut data sparepart yang disediakan dengan data yang salah	Proses pemasukan data gagal, muncul pesan gagal.	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Ubah data			
Kasus dan Hasil Uji (Data normal)			
Mengubah atribut data pengguna yang sudah ada	Proses pemasukan berhasil, klik ubah	Sesuai dengan yang	[X] diterima [ ] ditolak

dalam *database* dengan data yang benar	dan data yang baru tersimpan di *database*	diharapkan	
Kasus dan Hasil Uji (Data salah)			
Mengubah atribut data pengguna yang sudah ada dalam *database* dengan data yang salah	Proses gagal	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.8 dapat dilihat butir pengujian pada proses data Group adalah sebagai berikut:

Tabel 0.8 Pengujian data Group

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data Group yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut data Group yang disediakan dengan data yang salah	Proses pemasukan data gagal, muncul pesan gagal.	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Ubah data			
Kasus dan Hasil Uji (Data normal)			

Mengubah atribut data Group yang sudah ada dalam *database* dengan data yang benar	Proses pemasukan berhasil, klik ubah dan data yang baru tersimpan di *database*	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengubah atribut data Group yang sudah ada dalam *database* dengan data yang salah	Proses gagal	Tidak sesuai dengan yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

4.3.2.3 *Menu Data Master*

Pengolahan Data Master

Pada tabel 4.9 dapat dilihat butir pengujian pada proses data Master adalah sebagai berikut:

Tabel 0.9 Pengujian data normal pada proses Data Master

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data *warehouse,* jenis *warehouse, sparepart,* Jenis *sparepart, vendor*	Klik menu data master maka akan muncul sub menu yang berisi pengolahan data *warehouse,* jenis *warehouse,*	Sesuai yang diharapkan	[X] diterima [ ] ditolak

	sparepart, jenis *sparepart*, *vendor*		

Pada tabel 4.10 dapat dilihat butir pengujian pada proses data *sparepart* adalah sebagai berikut:

Tabel 0.10 Pengujian Data *Sparepart*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data sparepart yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut data sparepart yang disediakan dengan data yang salah	Proses pemasukan data gagal	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Ubah data			
Kasus dan Hasil Uji (Data normal)			
Mengubah atribut data yang sudah ada dalam *database* dengan data yang benar	Proses pemasukan berhasil, klik ubah dan data yang baru tersimpan di *database*	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengubah atribut data yang sudah ada dalam *database* dengan data yang salah	Proses gagal	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.11 dapat dilihat butir pengujian pada proses data *sparepart* adalah sebagai berikut:

Tabel 0.11 Pengujian Data Jenis Sparepart

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data jenis sparepart yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut data jenis sparepart yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Ubah data			
Kasus dan Hasil Uji (Data normal)			
Mengubah atribut data yang sudah ada dalam *database* dengan data yang benar	Proses pemasukan berhasil, klik ubah dan data yang baru tersimpan di *database*	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			

Mengubah atribut data yang sudah ada dalam *database* dengan data yang salah	Proses gagal	Tidak sesuai dengan yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.12 dapat dilihat butir pengujian pada proses data *warehouse* adalah sebagai berikut:

Tabel 0.12 Pengujian Data *Warehouse*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *warehouse* yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut data *warehouse* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[X] diterima [ ] ditolak
Ubah data			
Kasus dan Hasil Uji (Data normal)			

Mengubah atribut data yang sudah ada dalam *database* dengan data yang benar	Proses pemasukan berhasil, klik ubah dan data yang baru tersimpan di *database*	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengubah atribut data yang sudah ada dalam *database* dengan data yang salah	Proses gagal	Tidak sesuai dengan yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.13 dapat dilihat butir pengujian pada proses data jenis *warehouse* adalah sebagai berikut:

Tabel 0.13 Pengujian data Jenis *Warehouse*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data jenis *warehouse* yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut jenis *warehouse* yang disediakan dengan data	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak

yang salah			
Ubah data			
Kasus dan Hasil Uji (Data normal)			
Mengubah atribut data yang sudah ada dalam *database* dengan data yang benar	Proses pemasukan berhasil, klik ubah dan data yang baru tersimpan di *database*	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengubah atribut data yang sudah ada dalam *database* dengan data yang salah	Proses gagal	Tidak sesuai dengan yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.14 dapat dilihat butir pengujian pada proses data *vendor* adalah sebagai berikut:

Tabel 0.14 Pengujian data *Vendor*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *vendor* yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak

Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *vendor* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Ubah data			
Kasus dan Hasil Uji (Data normal)			
Mengubah atribut data yang sudah ada dalam *database* dengan data yang benar	Proses pemasukan berhasil, klik ubah dan data yang baru tersimpan di *database*	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengubah atribut data yang sudah ada dalam *database* dengan data yang salah	Proses gagal	Tidak sesuai dengan yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

4.3.2.4 *Menu Data Transaksi*

Pada tabel 4.15 dapat dilihat butir pengujian pada proses data Transaksi adalah sebagai berikut:

Tabel 0.15 Pengujian data normal pada proses Data Transaksi

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data *store request,detail store request,purchase*	Klik menu data *store request,detail*	Sesuai yang diharapkan	[X] diterima [ ] ditolak

request ,detail purchase request, issue, detail issue,purchase order, detail purchase order, receive, detail receive	*store request,purchase request ,detail purchase request, issue, detail issue,purchase order, detail purchase order, receive, detail receive* menu yang berisikan - pengolahan data-data transaksi		

Pada tabel 4.16 dapat dilihat butir pengujian pada proses data *Store Request* adalah sebagai berikut:

Tabel 0.16 Pengujian data *Store Request*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *store request* yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *Store Request* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Hapus			

Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.17 dapat dilihat butir pengujian pada proses data *Detail Store Request* adalah sebagai berikut:

Tabel 0.17 Pengujian data *Detail Store Request*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *Detail Store Request* yang disediakan dengan data yang benar, ubah status menjadi close	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*, daftar *store request* masuk ke menu *issue*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *Detail Store Request* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.18 dapat dilihat butir pengujian pada proses data *Purchase Request* adalah sebagai berikut:

Tabel 0.18 Pengujian data *Purchase Request*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *Purchase Request* yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *Purchase Request* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.19 dapat dilihat butir pengujian pada proses data *Detail Purchase Request* adalah sebagai berikut:

Tabel 0.19 Pengujian data *Detail Purchase Request*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *Detail Purchase Request* yang disediakan dengan data yang benar, ubah status menjadi *close*	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*,	Sesuai yang diharapkan	[X] diterima [ ] ditolak

	daftar *Detail Purchase Request* masuk ke menu *Purchase Order*		
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *Detail Purchase Request* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.20 dapat dilihat butir pengujian pada proses data *Purchase Order* adalah sebagai berikut:

Tabel 0.20 Pengujian data *Purchase Order*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *Purchase Order* yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *Purchase Order* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Hapus			

Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.21 dapat dilihat butir pengujian pada proses data *Detail Purchase Order* adalah sebagai berikut:

Tabel 0.21 Pengujian data *Detail Purchase Order*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *Detail Purchase Order* yang disediakan dengan data yang benar, ubah status menjadi *approved*	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*, daftar *Detail Purchase Order* masuk ke menu *receive*	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *Detail Purchase Order* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.22 dapat dilihat butir pengujian pada proses data *Receive* adalah sebagai berikut:

Tabel 0.22 Pengujian data *Receive*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *Receive* yang disediakan dengan data yang benar	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke dalam *database*, stok bertambah	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *Receive* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

Pada tabel 4.23 dapat dilihat butir pengujian pada proses data *Detail Receive* adalah sebagai berikut:

Tabel 0.23 Pengujian data *Detail Receive*

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *Detail Purchase Order* yang disediakan dengan data yang benar, ubah	Proses pemasukan berhasil, klik simpan, data yang baru masuk ke	Sesuai yang diharapkan	[X] diterima [ ] ditolak

status menjadi approved	dalam *database*, jumlah stock bertambah		
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut *Detail Receive* yang disediakan dengan data yang salah	Proses pemasukan data gagal	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak
Hapus			
Memilih data yang akan dihapus	Klik tombol hapus, maka data pada *database* telah berhasil dihapus	Sesuai dengan yang diharapkan	[X] diterima [ ] ditolak

4.3.2.5 *Menu Peramalan*

Pada tabel 4.24 dapat dilihat butir pengujian pada proses data peramalan adalah sebagai berikut

Tabel 0.24 Pengujian data normal pada proses peramalan

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data peramalan (periode, nama *sparepart*)	Perhitungan hasil peramalan, perhitungan kesalah peramalan (galat)	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengisi atribut peramlan yang disediakan dengan data yang salah (kosong)	Proses peramalan tidak bisa dilanjutkan dan muncul pesan data tidak boleh kosong	Tidak sesuai yang diharapkan	[ ] diterima [X] ditolak

4.3.2.5 *Menu Laporan*

Proses Pelaporan

Pada Tabel 4.25 dapat dilihat butir pengujian pada proses tampilan:

Tabel 0.25 Pengujian Data Normal Pada Proses Tampilan Laporan

Kasus dan Hasil Uji (Data normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data Tanggal Awal dan Tanggal Akhir pelaporan	Report tampil sesuai dengan pemilihan tanggal dan jam	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data salah)			
Mengosongkan data Tanggal Awal dan Akhir pelaporan	Tidak ada laporan yang tampil dan muncul pesan “tidak ada data di periode ini”	Sesuai yang diharapkan	[X] diterima [ ] ditolak

4.3.3 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian dengan kasus sample uji menggunakan metode *black box* mendapatkan kesimpulan bahwa pada proses masih bisa terjadi beberapa kesalahan jika data yang dimasukan bukan data yang seharusnya, maka akan dilakukan perbaikan di bagian *coding* program dan melakukan pengujian alpha kembali untuk bagian yang tidak sesuai.

4.3.3.1 Recana Pengujian Alpha 2

Rencana pengujian Alpha 2 Sistem Informasi pengendalian persediaan *sparepart* menggunakan pengujian *black box.* Pengujian ini digunakan untuk menguji fungsional dari sistem yang masih belun sesuai dengan harapan. Rencana pengujian dapat dilihat pada tabel 4.26 yang akan dijalankan pada sistem ini :

Tabel 4.26 Rencana Pengujian Alpha 2

Pengujian	Butir Uji	Jenis Pengujian
Pengolahan Data Group	1. Tambah Data Group 2. Hapus Data Group 3. Edit Data Group	*Black box*
Pengolahan Data Jenis Warehouse	1. Tambah Data Jenis Warehouse 2. Hapus Data Jenis Warehouse 3. Edit Data Jenis Warehouse	*Black box*
Pengolahan Data Warehouse	1. Tambah Data Jenis Warehouse 2. Hapus Data Jenis Warehouse 3. Edit Data Jenis Warehouse	*Black box*
Pengolahan Data Jenis Sparepart	1. Tambah Data Jenis Sparepart 2. Hapus Data Jenis Sparepart 3. Edit Data Jenis Sparepart	*Black box*
Pengolahan Data Sparepart	1. Tambah Data Sparepart 2. Hapus Data Sparepart 3. Edit Data Sparepart	*Black box*
Pengolahan Data Vendor	1. Tambah Data Vendor 2. Hapus Data Vendor 3. Edit Data Vendor	*Black box*
Pengolahan Data Store Request	1. Tambah Data Store Request 2. Hapus Data Store Request	*Black box*
Pengolahan Data Detail Store Request	1. Tambah Data Detail Store Request 2. Hapus Data Detail Store Request 3. Ubah Status Data Store Request	*Black box*
Pengolahan Data Purchase Request	1. Tambah Data Purchase Request 2. Hapus Data Purchase Request	*Black box*
Pengolahan Data Detail Purchase Request	1. Tambah Data Detail Purchase Request 2. Hapus Data Detail Purchase Request	*Black box*

	3. Ubah Status Data Purchase Request	
Pengolahan Data Purchase Order	1. Tambah Data Purchase Order 2. Hapus Data Purchase Order	*Black box*
Pengolahan Data Detail Purchase Order	1. Tambah Data Detail Purchase Order 2. Hapus Data Detail Purchase Order 3. Ubah Status Data Purchase Order	*Black box*
Pengolahan Data Receive	1. Tambah Data Receive 2. Hapus Data Receive	*Black box*
Pengolahan Detail Data Receive	1. Tambah Data Detail Receive 2. Hapus Detail Data Receive 3. Ubah Status Data Receive	*Black box*

Tabel 4.27 Pengujian Alpha 2

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi atribut data *group* yang disediakan dengan data yang salah atau sudah ada	Proses pemasukan data ditolak dan muncul peringatan	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Mengisi atribut *sparepart* dan jenis *sparepart* yang disediakan dengan data yang salah atau sudah ada	Proses pemasukan data ditolak dan muncul peringatan	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Mengisi atribut *warehouse* dan jenis *warehouse* yang disediakan dengan data yang salah atau sudah ada	Proses pemasukan data ditolak dan muncul peringatan	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Mengisi atribut *Vendor* yang disediakan dengan	Proses pemasukan data ditolak dan	Sesuai yang diharapkan	[X] diterima [ ] ditolak

data yang salah atau sudah ada	muncul peringatan		
Mengisi atribut *Store request* dan *detail store request* yang disediakan dengan data yang salah atau sudah ada	Proses pemasukan data ditolak dan muncul peringatan	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Mengisi atribut *Purchase request* dan *detai purchase request* yang disediakan dengan data yang salah atau sudah ada	Proses pemasukan data ditolak dan muncul peringatan	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Mengisi atribut *Purchase Order* dan *detail purchase order* yang disediakan dengan data yang salah atau sudah ada	Proses pemasukan data ditolak dan muncul peringatan	Sesuai yang diharapkan	[X] diterima [ ] ditolak
Mengisi atribut *receive* dan *detail receive* yang disediakan dengan data yang salah atau sudah ada	Proses pemasukan data ditolak dan muncul peringatan	Sesuai yang diharapkan	[X] diterima [ ] ditolak

4.3.3.2 Kesimpulan Pengujian Alpha 2

Berdasarkan hasil pengujian Alpha 2 dengan kasus sample uji menggunakan metode *black box* mendapatkan kesimpulan bahwa sistem informasi pengendalian persediaan *sparepart* yang dibuat sudah sesuai dengan yang diharapkan.

4.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana diuji secara langsung di tempat penelitian dek ngan membuat kuisoner

mengenai kepuasan user dengan kandungan poin syarat *user friendly*. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulan terhadap penelitian penerapan sistem yang baru.

4.4.1 Wawancara Pengguna

Wawancara dilakukan dengan menggunakan teknik kualitatif melalui pertanyaan terbuka. Pengujian beta melalui wawancara ditunjukan untuk staff Gudang, staff Purchase, staff Ekspedisi, dan *Factory Manager*.

4.4.1.1 Wawancara Pengujian Beta terhadap Staff Gudang *Sparepart*

Wawancara pengujian beta terhadap staff gudang *sparepart* dilakukan untuk mengetahui pendapat serta pandangan pengguna mengenai sistem informasi pengendalian persediaan *sparepart*. Narasumber wawancara dapat dilihat pada tabel 4.28.

Tabel 4.28 Narasumber wawancara (*staff gudang sparepart*)

Nama	Jabatan	Waktu
Sahara Siti	Staff Gudang *Sparepart*	Rabu, 27 Agustus 2013

Pertanyaan dan jawaban dari hasil wawancara terhadap staff gudang *sparepart* dapat dilihat pada tabel 4.29.

Tabel 4.29 Hasil wawancara (Staff Gudang *Sparepart*)

No	Pertanyaan dan Jawaban
1	Pertanyaan : Apakah sistem informasi pengendalian persediaan *sparepart* ini memudahkan anda dalam mengelola data *sparepart*?
	Jawaban :
	Narasumber 1 :

	Ya, sistem ini memudahkan saya dalam proses pengelolaan data *sparepart*
	Narasumber 2:
	bila dibandingkan dengan sistem yang lama sistem yang baru ini lebih memudahkan saya dalam pengelolaan data *sparepart.*
2	Pertanyaan : Apakah anda mengalami kesulitan dalam menggunakan sistem informasi pengendalian persediaan *sparepart* ini?
	Jawaban :
	Narasumber 1 :
	Awal saya mencoba sistem ini saya mengalami kesulitan karena saya masih perlu penyesuaian, namun setelah saya mencoba beberapa kali akhirnya saya terbiasa dan tidak mengalami kendala lagi.
	Narasumber 2 :
	Tidak ada masalah
3	Pertanyaan : Apa pendapat anda tentang notifikasi/peringatan yang muncul jika anda memasukan data yang sama?
	Jawaban :
	Narasumber 1 :
	Bagus sekali, karena dengan adanya notifikasi tersebut membantu kami mengurangi terjadinya duplikasi data
	Narasumber 2 :
	Bagus sekali itu yang kami butuhkan

4.4.1.2 Wawancara Pengujian Beta terhadap Staff *Purchase*

Wawancara pengujian beta terhadap staff *purchase* dilakukan untuk mengetahui pendapat serta pandangan pengguna mengenai sistem informasi pengendalian persediaan *sparepart*. Narasumber wawancara dapat dilihat pada tabel 4.30.

Tabel 4.30 Narasumber wawancara (staff *purchase*)

Nama	Jabatan	Waktu
Retno	Staff *Purchase*	Rabu, 27 Agustus 2013

Pertanyaan dan jawaban dari hasil wawancara terhadap staff *purchase* dapat dilihat pada tabel 4.31.

Tabel 4.31 Hasil wawancara (Staff *Purchase*)

No	Pertanyaan dan Jawaban
1	Pertanyaan : Apakah sistem informasi pengendalian persediaan *sparepart* ini membantu anda dalam proses pembuatan PO?
	Jawaban : Ya, dengan adanya sistem informasi pengendalian *sparepart* ini saya tidak perlu lagi membuat PO secara manual dan saya tidak perlu lagi menyimpan banyak surat permintaan pembelian di meja saya.
2	Pertanyaan : Apakah dengan adanya sistem informasi pengendalian persediaan *sparepart* anda tidak lagi kesulitan dalam pencarian data pembelian terakir?
	Jawaban : Ya, saya sudah tidak mengalami kesulitan lagi
3	Pertanyaan : Bagaimana penyajian laporan-laporan yang disediakan oleh sistem informasi pengendalian persediaan *sparepart* ini?
	Jawaban : Sudah cukup lengkap
4	Pertanyaan : Apakah anda kesulitan meggunakan sistem informasi pengendalian persediaan *sparepart* ini?

	Jawaban : tidak, sampai saat ini saya belum menemukan kesulitan dalam menggunakan sistem ini.

4.4.2.3 Wawancara Pengujian Beta terhadap Staff Ekspedisi

Wawancara pengujian beta terhadap staff ekspedisi dilakukan untuk mengetahui pendapat serta pandangan pengguna mengenai sistem informasi pengendalian persediaan *sparepart*. Narasumber wawancara dapat dilihat pada tabel 4.32.

Tabel 4.32 Narasumber wawancara (staff ekspedisi)

Nama	Jabatan	Waktu
Nucky	Staff Ekspedisi	Rabu, 27 Agustus 2013

Pertanyaan dan jawaban dari hasil wawancara terhadap Staff Ekspedisi dapat dilihat pada tabel 4.33.

Tabel 4.33 Hasil Wawancara (Staff Ekspedisi)

No	Pertanyaan dan Jawaban
1	Pertanyaan : Apakah sistem informasi pengendalian persediaan *sparepart* ini membantu anda dalam proses pemeriksaan *sparepart* yang masuk?
	Jawaban : ya, dengan adanya sistem ini saya tidak perlu lagi mencari PO di tumpukan arsip.
2	Pertanyaan : Bagaimana penyajian laporan-laporan yang disediakan oleh sistem informasi pengendalian persediaan *sparepart* ini?
	Jawaban : sudah cukup

3	Pertanyaan : Apakah sistem informasi pengendalian persediaan *sparepart* ini menyulitkan anda?
	Jawaban : tidak sama sekali, malah membantu saya dalam pencarian PO

4.4.2.4 Wawancara Pengujian Beta terhadap *Factory Manager*

Wawancara pengujian beta terhadap *Factory Manager* dilakukan untuk mengetahui pendapat serta pandangan pengguna mengenai sistem informasi pengendalian persediaan *sparepart*. Narasumber wawancara dapat dilihat pada tabel 4.34.

Tabel 4.34 Narasumber wawancara (*Factory Manager*)

Nama	Jabatan	Waktu
Hendra	*Factory Manager*	Rabu, 27 Agustus 2013

Pertanyaan dan jawaban dari hasil wawancara terhadap *Factory Manager* dapat dilihat pada tabel 4.35.

Tabel 4.35 Hasil wawancara (*Factory Manager*)

No	Pertanyaan dan Jawaban
1	Pertanyaan : Setelah anda mencoba sistem informasi pengendalian *sparepart*, bagaimana pendapat anda tentang fitur peramalannya?
	Jawaban : Fitur peramalannya bagus, saya jadi punya gambaran lain akan jumlah kebutuhan *sparepart*.
2	Pertanyaan : Apakah hasil peramalannya membantu anda menentukan jumlah *sparepart* yang akan dibeli?

	Jawaban : ya, itu akan cukup membantu saya
3	Pertanyaan : Dengan adanya sistem peramalan ini apakah anda masih mengalami kesulitan dalam menentukan jumlah *sparepart* yang akan dibeli?
	Jawaban : tidak, setelah saya mendapatkan perkiraan jumlah *sparepart* saya tinggal memberikan toleransi terhadap hasil peramalannya.
4	Pertanyaan : Bagaimana pendapat anda tentang laporan-laporan yang disajikan oleh sistem informasi pengendalian persediaan *sparepart* ini?
	Jawaban : Saya rasa sudah cukup baik .

4.4.2 Kesimpulan Pengujian Beta

Berdasarkan hasil pengujian beta yang dilakukan maka penerapan sistem informasi pengendalian persediaan *sparepart* ini sudah sesuai dengan tujuan yaitu memberikan kemudahan kepada staff gudang *sparepart* dalam proses pemasukan data ke dalam sistem aplikasi, membantu staff *purchasing* dalam pembuatan *purchase order*, membantu *factory manager* dalam memutuskan atau menentukan jumlah *sparepart* yang akan di beli dengan fasilitas peramalan kebutuhan *sparepart*, dan memudahkan staff ekspedisi untuk mendapatkan data *purchase order*.