BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implentasi sistem adalah tahap mengaplikasikan yang telah dirancang pada tahap perancangan sistem. Implementasi sistem dilakukan sebagai penerapan terhadap hasil analisis dan perancangan sistem kedalam bahasa pemrograman. Hasil dari penerapan tersebut diharapkan dapat menghasilkan sistem yang dapat diuji dan digunakan

4.1.1 Implementasi Basis Data

Data yang dibutuhkan sistem dibuat menjadi tabel – tabel yang dimasukan dalam suatu database. *Database Management System* (DBMS) yang digunakan untuk mengolah tabel-tabel tersebut adalah MySQL. Implementasi basis data dijelaskan sebagai berikut ini.

Tabel 4.1 Implementasi Basis Data

No	Nama Tabel	Nama File
1	users	users.sql
2	bahanbaku	bahanbaku.sql
3	kategoribahanbaku	kategori.sql
4	jenisbahanbaku	jenisbahanbaku.sql
5	produk	produk.sql
6	jenisproduk	jenisproduk.sql
7	resep	resep.sql
8	detailresep	detailresep.sql
9	penjualanproduk	penjualanproduk.sql
10	detailpenjualanproduk	detailpenjualanproduk.sql
11	peramalan	peramalan.sql
12	detailperamalan	detailperamalan.sql
13	EOQ	EOQ.sql
14	detailEOQ	detailEOQ.sql
15	*supplier*	*supplier*.sql
16	pembelianbahanbaku	pembelianbahanbaku.sql
17	detailpembelianbahanbaku	detailpembelianbahanbaku.sql

1. Users

Berikut ini adalah query pembuatan tabel users.

SQL

```
CREATE TABLE IF NOT EXISTS `users` (
  `username` varchar(36) NOT NULL,
  `password` varchar(36) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `jabatan` enum('Kepala Administrasi','Staff
Distribusi','Staff Gudang','Staff PPIC','Staff Purchasing')
NOT NULL,
  `status` enum('aktif','delete') NOT NULL DEFAULT 'aktif',
  `inUse` enum('ya','tidak') NOT NULL DEFAULT 'tidak',
  PRIMARY KEY (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. Bahan Baku

Berikut ini adalah query pembuatan tabel bahan baku.

SQL

```
CREATE TABLE IF NOT EXISTS `bahanbaku` (
  `idBahanBaku` varchar(15) NOT NULL,
  `idJenisBahanBaku` varchar(7) NOT NULL,
  `nama` varchar(255) DEFAULT NULL,
  `jumlah` int(11) NOT NULL,
  `satuan` enum('kg','butir') NOT NULL,
  `status` enum('aktif','delete') NOT NULL DEFAULT 'aktif',
  PRIMARY KEY (`idBahanBaku`),
  ALTER TABLE `bahanbaku`
  ADD CONSTRAINT `bahanbaku_ibfk_1` FOREIGN KEY
(`idJenisBahanBaku`) REFERENCES `jenisbahanbaku`
(`idJenisBahanBaku`) ON UPDATE CASCADE;`
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

3. Kategori Bahan Baku

Berikut ini adalah query pembuatan tabel kategori bahan baku.

SQL

```
CREATE TABLE IF NOT EXISTS `kategoribahanbaku` (
  `idKategoriBahanBaku` varchar(5) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `status` enum('aktif','delete') NOT NULL DEFAULT 'aktif',
  PRIMARY KEY (`idKategoriBahanBaku`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4. Jenis Bahan Baku

Berikut ini adalah query pembuatan tabel jenis bahan baku.

SQL

```
CREATE TABLE IF NOT EXISTS `jenisbahanbaku` (
  `idJenisBahanBaku` varchar(7) NOT NULL,
  `idKategoriBahanBaku` varchar(5) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `status` enum('aktif','delete') NOT NULL DEFAULT 'aktif',
  PRIMARY KEY (`idJenisBahanBaku`),
  ALTER TABLE `jenisbahanbaku`
  ADD CONSTRAINT `jenisbahanbaku_ibfk_1` FOREIGN KEY
(`idKategoriBahanBaku`) REFERENCES `kategoribahanbaku`
(`idKategoriBahanBaku`) ON UPDATE CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5. Produk

Berikut ini adalah query pembuatan tabel produk.

SQL

```
CREATE TABLE IF NOT EXISTS `produk` (
  `idProduk` varchar(5) NOT NULL,
  `idJenisProduk` int(2) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `harga` int(11) NOT NULL,
  `status` enum('aktif','delete') NOT NULL DEFAULT 'aktif',
  PRIMARY KEY (`idProduk`),
ALTER TABLE `produk`
  ADD CONSTRAINT `produk_ibfk_1` FOREIGN KEY
(`idJenisProduk`) REFERENCES `jenisproduk`
(`idJenisProduk`) ON UPDATE CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

6. Jenis Produk

Berikut ini adalah query pembuatan tabel produk.

SQL

```
CREATE TABLE IF NOT EXISTS `jenisproduk` (
  `idJenisProduk` int(2) NOT NULL AUTO_INCREMENT,
  `nama` varchar(50) NOT NULL,
  `status` enum('aktif','delete') NOT NULL DEFAULT 'aktif',
  PRIMARY KEY (`idJenisProduk`),
  UNIQUE KEY `nama` (`nama`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

7. Tabel Resep

Berikut ini adalah query pembuatan tabel resep.

SQL

```
CREATE TABLE IF NOT EXISTS `resep` (
  `idResep` int(5) NOT NULL AUTO_INCREMENT,
  `idProduk` varchar(5) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `jumlah` double NOT NULL,
  PRIMARY KEY (`idResep`),
ALTER TABLE `resep`
  ADD CONSTRAINT `resep_ibfk_1` FOREIGN KEY (`idProduk`)
REFERENCES `produk` (`idProduk`) ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

8. Detail Resep

Berikut ini adalah query pembuatan tabel detail resep.

SQL

```
CREATE TABLE IF NOT EXISTS `detailresep` (
  `idDetailResep` int(11) NOT NULL AUTO_INCREMENT,
  `idResep` int(5) NOT NULL,
  `idBahanBaku` varchar(30) NOT NULL,
  `idDetailEOQ` int(11) NOT NULL,
  `jumlahBahan` double NOT NULL,
  PRIMARY KEY (`idDetailResep`),
  FOREIGN KEY (`idResep`)REFERENCES `idResep`,
  FOREIGN KEY (`idBahanBaku`)REFERENCES `idBahanBaku`,
  FOREIGN KEY (`idDetailEOQ`)REFERENCES `idDetailEOQ`
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

9. Penjualan Produk

Berikut ini adalah query pembuatan tabel penjualan produk.

SQL

```
CREATE TABLE IF NOT EXISTS `penjualanproduk` (
  `idPenjualanProduk` int(11) NOT NULL AUTO_INCREMENT,
  `idProduk` varchar(5) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `jumlah` double NOT NULL,
  `tanggal` date DEFAULT NULL,
  `total` double NOT NULL,
  `username` varchar(36) NOT NULL,
  PRIMARY KEY (`idPenjualanProduk`),

ALTER TABLE `penjualanproduk`
  ADD CONSTRAINT `penjualanproduk_ibfk_2` FOREIGN KEY
(`username`) REFERENCES `users` (`username`) ON UPDATE
CASCADE,
  ADD CONSTRAINT `penjualanproduk_ibfk_3` FOREIGN KEY
(`idProduk`) REFERENCES `produk` (`idProduk`) ON UPDATE
CASCADE;`
```

```
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

10. Detail Penjualan Produk

Berikut ini adalah query pembuatan tabel detail penjualan produk.

SQL

```
CREATE TABLE IF NOT EXISTS `detailpenjualanproduk` (
  `idDetailPenjualanProduk` int(11) NOT NULL
AUTO_INCREMENT,
  `idPenjualanProduk` int(11) NOT NULL,
  PRIMARY KEY (`idDetailPenjualanProduk`),
  FOREIGN KEY (`idPenjualanProduk`)REFERENCES
`idPenjualanProduk`
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

11. Peramalan

Berikut ini adalah query pembuatan tabel peramalan.

SQL

```
CREATE TABLE IF NOT EXISTS `peramalan` (
  `idPeramalan` int(11) NOT NULL AUTO_INCREMENT,
  `idPenjualanProduk` int(11) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `tanggal` date NOT NULL,
  PRIMARY KEY (`idPeramalan`),
  FOREIGN KEY (`idPenjualanProduk`)REFERENCES
`idPenjualanProduk`
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

12. Detail Peramalan

Berikut ini adalah query pembuatan tabel detail peramalan.

SQL

```
CREATE TABLE IF NOT EXISTS `detailperamalan` (
  `idDetailPeramalan` int(11) NOT NULL AUTO_INCREMENT,
  `idPeramalan` int(11) NOT NULL,
  `idDetailPenjualanProduk` int(11) NOT NULL,
  `jumlah` double NOT NULL,
  `bulanAkhir` date NOT NULL,
  `hasil` double NOT NULL,
  `error` int(11) NOT NULL,
  PRIMARY KEY (`idDetailPeramalan`),
  FOREIGN KEY (`idPeramalan`)REFERENCES `idPeramalan`,
  FOREIGN KEY (`idDetailPenjualanProduk`)REFERENCES
`idDetailPenjualanProduk`
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

13. EOQ

Berikut ini adalah query pembuatan tabel EOQ.

SQL

```
CREATE TABLE IF NOT EXISTS `eoq` (
  `idEOQ` int(11) NOT NULL AUTO_INCREMENT,
  `idProduk` varchar(5) NOT NULL,
  PRIMARY KEY (`idEOQ`),
  FOREIGN KEY (`idProduk`)REFERENCES `idProduk`
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

14. Detail EOQ

Berikut ini adalah query pembuatan tabel detail EOQ.

SQL

```
CREATE TABLE IF NOT EXISTS `detaileoq` (
  `idDetailEOQ` int(11) NOT NULL AUTO_INCREMENT,
  `idEOQ` int(11) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `minimalPembelian` double NOT NULL,
  `jumlahKebutuhan` double NOT NULL,
  `jumlahBiayaPesan` double NOT NULL,
  `jumlahBiayaSimpan` double NOT NULL,
  `totalSeluruhBiaya` double NOT NULL,
  `Q` double NOT NULL,
  `biayaPesan` double NOT NULL,
  `biayaSimpan` double NOT NULL,
  `periodePesan` double NOT NULL,
  `frekuensiPesan` double NOT NULL,
  `totalPersediaan` double NOT NULL,
  PRIMARY KEY (`idDetailEOQ`),
  FOREIGN KEY (`idEOQ`)REFERENCES `idEOQ`
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

15. *Supplier*

Berikut ini adalah query pembuatan tabel *supplier*.

SQL

```
CREATE TABLE IF NOT EXISTS `supplier` (
  `nama Supplier` varchar(255) NOT NULL,
  `alamat` varchar(255) NOT NULL,
  `status` enum('aktif','delete') NOT NULL DEFAULT 'aktif',
  PRIMARY KEY (`nama Supplier`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

16. Pembelian Bahan Baku

Berikut ini adalah query pembuatan tabel pembelian bahan baku.

SQL

```
CREATE TABLE IF NOT EXISTS `pembelianbahanbaku` (
  `idPembelianBahanBaku` int(11) NOT NULL AUTO_INCREMENT,
  `nama*Supplier*` varchar(255) NOT NULL,
  `username` varchar(36) NOT NULL,
  `idBahanBaku` varchar(10) NOT NULL,
  `jumlah` double NOT NULL,
  `harga` double NOT NULL,
  `totalHarga` double NOT NULL,
  `tanggal` date DEFAULT NULL,
  `minBeli` double NOT NULL,
  PRIMARY KEY (`idPembelianBahanBaku`),
  FOREIGN KEY (`username`)REFERENCES `username`,
  FOREIGN KEY (`nama*Supplier*`)REFERENCES `nama*Supplier*`
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

17. Detail Pembelian Bahan Baku

Berikut ini adalah query pembuatan tabel detail pembelian bahan baku.

SQL

```
CREATE TABLE IF NOT EXISTS `detailpembelianbahanbaku` (
  `idDetailPembelianBahanBaku` int(11) NOT NULL
AUTO_INCREMENT,
  `idPembelianBahanBaku` int(11) NOT NULL,
  `idBahanBaku` varchar(15) NOT NULL,
  `idDetailEOQ` int(11) NOT NULL,
  `detilJumlah` double NOT NULL,
  PRIMARY KEY (`idDetailPembelianBahanBaku`),
  FOREIGN KEY (`idPembelianBahanBaku`)REFERENCES
`idPembelianBahanBaku`,
  FOREIGN KEY (`idBahanBaku`)REFERENCES `idBahanBaku`,
  FOREIGN KEY (`idDetailEOQ`)REFERENCES `idDetailEOQ`
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

4.1.2 Implementasi Antarmuka

Implementasi antarmuka menggambarkan tampilan dari aplikasi yang dibangun yaitu implementasi antarmuka sistem informasi perencanaan dan pengendalian bahan baku PT Mustika Cita Rasa (Holland Bakery). Implementasi antarmuka dapat dilihat pada tabel berikut ini.

Tabel 4.2 Implementasi Antarmuka Kepala Administrasi

Antarmuka	Proses ke-	Nama File
Login	1	ULogin.pas
Data User	2	UUser.pas
Tambah Data User	2.1	UUserTambah.pas
Ubah Data User	2.2	UUserEdit.pas
Hapus Data User	2.3	UUser.pas
Cari Data User	2.4	UUser.pas

Tabel 4.3 Implementasi Antarmuka Departemen Gudang

Antarmuka	Proses ke-	Nama File
Login	1	ULogin.pas
Data Master	3	UMenu.pas
Data Bahan Baku	3.1	UBahanBaku.pas
Tambah Data Bahan Baku	3.1.1	UBahanBakuTambah.pas
Ubah Data Bahan Baku	3.1.2	UBahanBakuEdit.pas
Hapus Data Bahan Baku	3.1.3	UBahanBaku.pas
Cari Data Bahan Baku	3.1.4	UBahanBaku.pas
Cetak Data Bahan Baku	3.1.5	ULBahanBaku.pas
Data Kategori Bahan Baku	3.2	UKategoriBahanBaku.pas
Tambah Data Kategori Bahan Baku	3.2.1	UKategoriBahanBakuTambah.pas
Ubah Data Kategori Bahan Baku	3.2.2	UKategoriBahanBakuEdit.pas
Hapus Data Kategori Bahan Baku	3.2.3	UKategoriBahanBaku.pas
Cari Data Kategori Bahan Baku	3.2.4	UKategoriBahanBaku.pas
Data Jenis Bahan Baku	3.3	UBahanBaku.pas
Tambah Data Jenis Bahan Baku	3.3.1	UJenisBahanBakuTambah.pas
Ubah Data Jenis Bahan Baku	3.3.2	UJenisBahanBakuEdit.pas
Hapus Data Jenis Bahan Baku	3.3.3	UJenisBahanBaku.pas
Cari Data Jenis Bahan Baku	3.3.4	UJenisBahanBaku.pas

Tabel 4.4 Implementasi Antarmuka Departemen Distribusi

Antarmuka	Proses ke-	Nama File
Login	1	ULogin.pas
Data Master	3	UMenu.pas
Data Produk	3.4	UProduk.pas
Tambah Data Produk	3.4.1	UProdukTambah.pas
Ubah Data Produk	3.4.2	UProdukEdit.pas
Hapus Data Produk	3.4.3	UProduk.pas
Cari Data Produk	3.4.4	UProduk.pas
Data Produk	3.5	UJenisProduk.pas
Tambah Data Jenis Produk	3.5.1	UJenisProdukTambah.pas
Ubah Data Jenis Produk	3.5.2	UJenisProdukEdit.pas
Hapus Data Jenis Produk	3.5.3	UJenisProduk.pas

Antarmuka	Proses ke-	Nama File
Cari Data Jenis Produk	3.5.4	UJenisProduk.pas
Data Transaksi	4	UMenu.pas
Data Penjualan Produk	4.3	UPenjualanProduk.pas
Tambah Data Penjualan Produk	4.3.1	UPenjualanProdukTambah.pas
Ubah Data Penjualan Produk	4.3.2	UPenjualanProdukUbah.pas
Hapus Data Penjualan Produk	4.3.3	UPenjualanProduk.pas
Cari Data Penjaualn Produk	4.3.4	UPenjualanProduk.pas
Data Peramalan	4.4	UPeramalan.pas
Hitung Data Peramalan	4.4.1	UPeramalanTambah.pas
Ubah Data Peramalan	4.4.2	UPeramalanUbah.pas
Hapus Data Peramalan	4.4.3	UPeramalan.pas
Cari Data Peramalan	4.4.4	UPeramalan.pas

Tabel 4.5 Implementasi Antarmuka Departemen PPIC

Antarmuka	Proses ke-	Nama File
Login	1	ULogin.pas
Data Master	3	UMenu.pas
Data Resep	3.6	UResep.pas
Tambah Data Resep	3.6.1	UResepTambah.pas
Ubah Data Resep	3.6.2	UResepUbah.pas
Hapus Data Resep	3.6.3	UResep.pas
Cari Data Resep	3.6.4	UResep.pas
Data Detail Resep	3.7	UDetailResep.pas
Memilih Resep	3.7.1	UDetailResepTambah.pas
Memilih Komposisi Resep	3.7.2	UDetailResepTambah.pas
Hitung Pemakaian Bahan Baku	3.7.3	UDetailResepTambah.pas
Hitung EOQ	3.7.4	UDetailResepTambah.pas
Cari Data Resep	3.7.5	UDetailResep.pas
Ubah Data Resep	3.7.6	UDetailResepUbah.pas

Tabel 4.6 Implementasi Antarmuka Departemen *Purchasing*

Antarmuka	Proses ke-	Nama File
Login	1	ULogin.pas
Data Transaksi	4	UMenu.pas
Data Pembelian Bahan Baku	4.1	UPembelianBahanBaku.pas
Tambah Data Pembelian Bahan Baku	4.1.1	UPembelianBahanBakuTambah.pas
Ubah Data Pembelian Bahan Baku	4.1.2	UPembelianBahanBakuUbah.pas
Hapus Data Pembelian Bahan Baku	4.1.3	UPembelianBahanBaku.pas
Cari Data Pembelian Bahan Baku	4.1.4	UPembelianBahanBaku.pas
Data *Supplier*	4.2	U*Supplier*.pas

Antarmuka	Proses ke-	Nama File
Tambah Data *Supplier*	4.2.1	U*Supplier*Tambah.pas
Ubah Data *Supplier*	4.2.2	U*Supplier*Ubah.pas
Hapus Data *Supplier*	4.2.3	U*Supplier*.pas
Cari Data *Supplier*	4.2.4	U*Supplier*.pas

4.2 Pengujian Sistem

Pengujian sistem merupakan tahapan untuk menelusuri dan mempelajari sistem dengan cara melakukan serangkaian tes untuk mencoba sistem yang telah dibangun guna mengetahui bagaimana kualitas dari sistem tersebut.

4.2.1 Rencana Pengujian

Pengujian yang akan dilakukan adalah dengan menggunakan metode *black box* dan pengujian *beta* dengan melakukan wawancara kepada karyawan Holland Bakery. Rencana pengujian metode *black box* dapat dilihat pada table-tabel berikut ini.

Tabel 4.7 Rencana Pengujian Kepala Administrasi

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Username	*Black Box*
	Password	*Black Box*
Data User	Username	*Black Box*
	Password	*Black Box*
	Nama	*Black Box*
	Jabatan	*Black Box*

Tabel 4.8 Rencana Pengujian Departemen Gudang

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Username	*Black Box*
	Password	*Black Box*
Data Bahan Baku	Kategori	*Black Box*
	Jenis	*Black Box*
	Nama	*Black Box*
	Jumlah	*Black Box*
	Satuan	*Black Box*
Data Kategori Bahan Baku	Nama	*Black Box*
Data Jenis Bahan Baku	Nama	*Black Box*

Tabel 4.9 Rencana Pengujian Departemen Distribusi

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Username	*Black Box*
	Password	*Black Box*
Data Produk	Jenis Produk	*Black Box*
	Nama	*Black Box*
	Harga	*Black Box*
Data Jenis Produk	Nama	*Black Box*
Data Penjualan Produk	Tanggal	*Black Box*
	Nama Produk	*Black Box*
	Jumlah	*Black Box*
Data Peramalan	Nama Produk	*Black Box*
	Tanggal	*Black Box*
	Periode Awal	*Black Box*
	Periode Akhir	*Black Box*

Tabel 4.10 Rencana Pengujian Departemen PPIC

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Username	*Black Box*
	Password	*Black Box*
Data Resep	Tanggal	*Black Box*
	Nama	*Black Box*
	Jumlah	*Black Box*
Data Detail Resep	Nama Resep	*Black Box*
	Bahan Baku	*Black Box*
	Jumlah Bahan	*Black Box*

Tabel 4.11 Rencana Pengujian Departemen *Purchasing*

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Username	*Black Box*
	Password	*Black Box*
Data Pembelian Bahan Baku	Tanggal Pembelian	*Black Box*
	Nama *Supplier*	*Black Box*
	Minimal Pembelian	*Black Box*
	Harga Satuan	*Black Box*
Data *Supplier*	Nama *Supplier*	*Black Box*
	Alamat	*Black Box*

4.2.2 Pengujian *Black Box*

Pengujian *black box* dilakukan untuk menguji fungsi-fungsi dari perangkat lunak agar mengetahui apakah data yang dimasukan sudah benar atau belum benar serta apakah hasil keluarannya sudah sesuai dengan yang diinginkan atau tidak. Teknik pengujian *black box* yang digunakan adalah *Equivalence Partitioning* dan *Cause-Effect Relationship*.

4.2.2.1 *Equivalence Partitioning Testing*

Teknik pengujian *Equivalence Partitioning* adalah pengujian yang dilakukan dengan mencoba beberapa kemungkinan nilai masukan sebagai kelas-kelas uji untuk mengetahui kesalahan apa saja yang terdapat dalam sistem. Untuk pengujian *Equivalence Partitioning* dapat dilihat pada tabel berikut ini.

1. *Equivalence Partitioning Testing* Kepala Administrasi

a. *Login*

Tabel 4.12 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian *login* Kepala Administrasi.

Tabel 4.12 *Equivalence Partitioning Testing*-Login Kepala Administrasi

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	admin	Dapat memasukkan username pada *field* username	Dapat memasukkan username pada *field* username	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	admin	Dapat memasukkan password pada *field* password	Dapat memasukkan password pada *field* password	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Username tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong."	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Password tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong"	[✓] Diterima [ ] Ditolak

a. Data Users

Tabel 4.13 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian data user Kepala Administrasi.

Tabel 4.13 *Equivalence Partitioning Testing*-Data Users

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Pricilia	Dapat memasukkan nama pada *field* nama	Dapat memasukkan nama pada *field* nama	[✓] Diterima [ ] Ditolak
Jabatan	{text} [a-z, A-Z, 0-9, simbol]	-	Kepala Administrasi	Dapat memilih jabatan pada *combo box* jabatan	Dapat memilih jabatan pada *combo box* jabatan	[✓] Diterima [ ] Ditolak
Username	{text} [a-z, A-Z, 0-9, simbol]	-	admin	Dapat memasukkan username pada *field* username	Dapat memasukkan username pada *field* username	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	admin	Dapat memasukkan password pada *field* password	Dapat memasukkan password pada *field* password	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan “Nama tidak boleh kosong”	Dapat menampilkan pesan “Nama tidak boleh kosong”	[✓] Diterima [ ] Ditolak
Jabatan	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan “pilih jabatan”	Dapat menampilkan pesan “Silahkan pilih jabatan”	[✓] Diterima [ ] Ditolak
Username	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan “Username tidak boleh kosong”	Dapat menampilkan pesan “Username tidak boleh kosong.”	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan “Password tidak boleh kosong”	Dapat menampilkan pesan “Username tidak boleh kosong”	[✓] Diterima [ ] Ditolak

2. *Equivalence Partitioning Testing* Departemen Gudang

a. *Login*

Tabel 4.14 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian *login* Departemen Gudang.

Tabel 4.14 *Equivalence Partitioning Testing-Login* Departemen Gudang

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	gudang	Dapat memasukkan username pada *field* username	Dapat memasukkan username pada *field* username	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	gudang	Dapat memasukkan password pada *field* password	Dapat memasukkan password pada *field* password	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Username tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong."	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Password tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong"	[✓] Diterima [ ] Ditolak

b. Data Bahan Baku

Tabel 4.15 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian data bahan baku Departemen Gudang.

Tabel 4.15 *Equivalence Partitioning Testing*-Data Bahan Baku

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kategori	{text} [a-z, A-Z, 0-9, simbol]	-	Tepung	Dapat memilih kategori pada *combo box* kategori	Dapat memilih kategori pada *combo box* kategori	[✓] Diterima [ ] Ditolak
Jenis	{text} [a-z, A-Z, 0-9, simbol]	-	Cakra	Dapat memilih jenis pada *combo box* jenis	Dapat memilih jenis pada *combo box* jenis	[✓] Diterima [ ] Ditolak
Jumlah	{numeric} [0-9]	-	125000	Dapat memasukkan jumlah pada *field* jumlah	Dapat memasukkan jumlah pada *field* jumlah	[✓] Diterima [ ] Ditolak
Satuan	{text} [a-z, A-Z, 0-9, simbol]	-	Kg	Dapat memilih satuan pada *combo box* satuan	Dapat memilih satuan pada *combo box* satuan	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						

Butir Uji	Masukan yang Valid	Masukan yang Tidak Valid	Data yang Dimasukkan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Kategori	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan “pilih kategori”	Dapat menampilkan pesan “Silahkan pilih kategori”	[✓] Diterima [ ] Ditolak
Jenis	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan “pilih jenis”	Dapat menampilkan pesan “Silahkan pilih jenis”	[✓] Diterima [ ] Ditolak
Jumlah	{numeric} [0-9]	-	Null	Dapat menampilkan pesan “Jumlah tidak boleh kosong”	Dapat menampilkan pesan “Jumlah tidak boleh kosong”	[✓] Diterima [ ] Ditolak
Satuan	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan “Silahkan pilih satuan”	Dapat menampilkan pesan “Silahkan pilih satuan”	[✓] Diterima [ ] Ditolak

c. Data Kategori Bahan Baku

Tabel 4.16 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian kategori bahan baku Departemen Gudang.

Tabel 4.16 *Equivalence Partitioning Testing*-Data Kategori Bahan Baku

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Tepung	Dapat memasukkan nama pada *field* nama	Dapat memasukkan nama pada *field* nama	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan “nama tidak boleh kosong”	Dapat menampilkan pesan “nama tidak boleh kosong”	[✓] Diterima [ ] Ditolak

d. Data Jenis Bahan Baku

Tabel 4.17 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian jenis bahan baku Departemen Gudang.

Tabel 4.17 *Equivalence Partitioning Testing*-Data Jenis Bahan Baku

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kategori	{text} [a-z, A-Z, 0-9, simbol]	-	**Tepung**	Dapat memilih kategori pada *combo box* kategori	Dapat memilih kategori pada *combo box* kategori	[✓] Diterima [ ] Ditolak
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Cakra	Dapat memasukkan nama pada *field* nama	Dapat memasukkan nama pada *field* nama	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kategori	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih kategori"	Dapat menampilkan pesan "Silahkan pilih kategori"	[✓] Diterima [ ] Ditolak
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "nama tidak boleh kosong"	Dapat menampilkan pesan "nama tidak boleh kosong"	[✓] Diterima [ ] Ditolak

3. *Equivalence Partitioning Testing* Departemen Distribusi

a. *Login*

Tabel 4.18 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian *login* Departemen Distribusi.

Tabel 4.18 *Equivalence Partitioning Testing-Login* Departemen Distribusi

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	distribusi	Dapat memasukkan username pada *field* username	Dapat memasukkan username pada *field* username	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	distribusi	Dapat memasukkan password pada *field* password	Dapat memasukkan password pada *field* password	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						

Butir Uji	Masukan yang Valid	Masukan yang Tidak Valid	Data yang Dimasukkan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Username	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Username tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong."	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Password tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong"	[✓] Diterima [ ] Ditolak

b. Data Produk

Tabel 4.19 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian produk Departemen Distribusi.

Tabel 4.19 *Equivalence Partitioning Testing*-Data Produk

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Jenis	{text} [a-z, A-Z, 0-9, simbol]	-	Roti	Dapat memilih jenis pada *combo box* jenis	Dapat memilih jenis pada *combo box* jenis	[✓] Diterima [ ] Ditolak
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Coklat	Dapat memasukkan nama pada *field* nama	Dapat memasukkan nama pada *field* nama	[✓] Diterima [ ] Ditolak
Harga	{numeric} [0-9]	-	6500	Dapat memasukkan harga pada *field* harga	Dapat memasukkan harga pada *field* harga	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Jenis	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat memilih jenis pada *combo box* jenis	Dapat memilih jenis pada *combo box* jenis	[✓] Diterima [ ] Ditolak
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "nama tidak boleh kosong"	Dapat menampilkan pesan "nama tidak boleh kosong"	[✓] Diterima [ ] Ditolak
Harga	{numeric} [0-9]	-	Null	Dapat menampilkan pesan "harga tidak boleh kosong"	Dapat menampilkan pesan "harga tidak boleh kosong"	[✓] Diterima [ ] Ditolak

c. Data Jenis Produk

Tabel 4.20 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian jenis produk Departemen Distribusi.

Tabel 4.20 *Equivalence Partitioning Testing*-Data Jenis Produk

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Jenis	{text} [a-z, A-Z, 0-9, simbol]	-	Roti	Dapat memasukkan nama pada *field* nama	Dapat memasukkan nama pada *field* nama	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Jenis	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "nama tidak boleh kosong"	Dapat menampilkan pesan "nama tidak boleh kosong"	[✓] Diterima [ ] Ditolak

d. Data Penjualan Produk

Tabel 4.21 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian penjualan produk Departemen Distribusi.

Tabel 4.21 *Equivalence Partitioning Testing*-Data Penjualan Produk

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tanggal	{text} [a-z, A-Z, 0-9, simbol]	-	1 feb 2013	Dapat memasukkan tanggal pada *field* tanggal	Dapat memasukkan tanggal pada *field* tanggal	[✓] Diterima [ ] Ditolak
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Roti Coklat	Dapat memilih produk pada *combo box* produk	Dapat memilih produk pada *combo box* produk	[✓] Diterima [ ] Ditolak
Jumlah	{numeric} [0-9]	-	228	Dapat memasukkan jumlah pada *field* jumlah	Dapat memasukkan jumlah pada *field* jumlah	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						

Butir Uji	Masukan yang Valid	Masukan yang Tidak Valid	Data yang Dimasukkan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Tanggal	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "tanggal tidak boleh kosong"	Dapat menampilkan pesan "tanggal tidak boleh kosong"	[✓] Diterima [ ] Ditolak
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih produk"	Dapat menampilkan pesan "pilih produk"	[✓] Diterima [ ] Ditolak
Jumlah	{numeric} [0-9]	-	Null	Dapat menampilkan pesan "jumlah tidak boleh kosong"	Dapat menampilkan pesan "jumlah tidak boleh kosong"	[✓] Diterima [ ] Ditolak

e. Data Peramalan

Tabel 4.22 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian peramalan Departemen Distribusi.

Tabel 4.22 *Equivalence Partitioning Testing*-Data Peramalan

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
idProduk	{text} [a-z, A-Z, 0-9, simbol]	-	1001	Dapat memasukkan idproduk pada *field* idproduk	Dapat memasukkan idproduk pada *field* idproduk	[✓] Diterima [ ] Ditolak
Tanggal	{text} [a-z, A-Z, 0-9, simbol]	-	1 mar 2013	Dapat memilih tanggal pada *combo box* tanggal	Dapat memilih tanggal pada *combo box* tanggal	[✓] Diterima [ ] Ditolak
Jumlah	{numeric} [0-9]		912	Dapat memasukkan jumlah pada *field* jumlah	Dapat memasukkan jumlah pada *field* jumlah	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
idProduk	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "idproduk tidak boleh kosong"	Dapat menampilkan pesan "idproduk tidak boleh kosong"	[✓] Diterima [ ] Ditolak
Tanggal	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih tanggal"	Dapat menampilkan pesan "pilih tanggal"	[✓] Diterima [ ] Ditolak
Jumlah	{numeric} [0-9]		Null	Dapat menampilkan pesan "jumlah tidak boleh kosong"	Dapat menampilkan pesan "jumlah tidak boleh kosong"	[✓] Diterima [ ] Ditolak

4. *Equivalence Partitioning Testing* Departemen PPIC

a. *Login*

Tabel 4.23 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian *login* Departemen PPIC.

Tabel 4.23 *Equivalence Partitioning Testing-Login* Departemen PPIC

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	ppic	Dapat memasukkan username pada *field* username	Dapat memasukkan username pada *field* username	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	ppic	Dapat memasukkan password pada *field* password	Dapat memasukkan password pada *field* password	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Username tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong."	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Password tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong"	[✓] Diterima [ ] Ditolak

b. Data Resep

Tabel 4.24 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian resep Departemen PPIC.

Tabel 4.24 *Equivalence Partitioning Testing*-Data Resep

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Roti Coklat	Dapat memilih produk pada *combo box* produk	Dapat memilih produk pada *combo box* produk	[✓] Diterima [ ] Ditolak
Jumlah	{numeric} [0-9]	-	40	Dapat memasukkan jumlah pada *field* jumlah	Dapat memasukkan jumlah pada *field* jumlah	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						

Butir Uji	Masukan yang Valid	Masukan yang Tidak Valid	Data yang Dimasukkan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih produk"	Dapat menampilkan pesan "pilih produk"	[✓] Diterima [ ] Ditolak
Jumlah	{numeric} [0-9]	-	Null	Dapat menampilkan pesan "jumlah tidak boleh kosong"	Dapat menampilkan pesan "jumlah tidak boleh kosong"	[✓] Diterima [ ] Ditolak

c. Data Detail Resep

Tabel 4.25 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian detail resep Departemen PPIC.

Tabel 4.25 *Equivalence Partitioning Testing*-Data Detail Resep

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama Resep	{text} [a-z, A-Z, 0-9, simbol]	-	Roti Coklat	Dapat memilih resep pada *combo box* resep	Dapat memilih resep pada *combo box* resep	[✓] Diterima [ ] Ditolak
Nama Bahan	{text} [a-z, A-Z, 0-9, simbol]	-	Tepung Cakra	Dapat memilih bahan baku pada *combo box* bahan baku	Dapat memilih bahan baku pada *combo box* bahan baku	[✓] Diterima [ ] Ditolak
Jumlah Bahan	{numeric} [0-9]	-	1	Dapat memasukkan jumlah pada *field* jumlah	Dapat memasukkan jumlah pada *field* jumlah	[✓] Diterima [ ] Ditolak
Satuan	{text} [a-z, A-Z, 0-9, simbol]	-	Kg	Dapat memilih satuan pada *combo box* satuan	Dapat memilih satuan pada *combo box* satuan	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama Resep	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih resep"	Dapat menampilkan pesan "pilih resep"	[✓] Diterima [ ] Ditolak
Nama Bahan	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih bahan baku"	Dapat menampilkan pesan "pilih bahan baku"	[✓] Diterima [ ] Ditolak
Jumlah Bahan	{numeric} [0-9]	-	Null	Dapat menampilkan pesan "jumlah bahan tidak boleh kosong"	Dapat menampilkan pesan "jumlah bahan tidak boleh kosong"	[✓] Diterima [ ] Ditolak
Satuan	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih satuan"	Dapat menampilkan pesan "pilih satuan"	[✓] Diterima [ ] Ditolak

5. *Equivalence Partitioning Testing* Departemen *Purchasing*

a. *Login*

Tabel 4.26 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian *login* Departemen *Purchasing*.

Tabel 4.26 *Equivalence Partitioning Testing*-*Login* Departemen *Purchasing*

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	*purchasing*	Dapat memasukkan username pada *field* username	Dapat memasukkan username pada *field* username	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	*purchasing*	Dapat memasukkan password pada *field* password	Dapat memasukkan password pada *field* password	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Username tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong."	[✓] Diterima [ ] Ditolak
Password	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "Password tidak boleh kosong"	Dapat menampilkan pesan "Username tidak boleh kosong"	[✓] Diterima [ ] Ditolak

b. Data Supplier

Tabel 4.27 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian suplier Departemen *Purchasing*.

Tabel 4.27 *Equivalence Partitioning Testing*-Data *Supplier*

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	PT Mekar Kencana	Dapat memasukkan nama pada *field* nama	Dapat memasukkan nama pada *field* nama	[✓] Diterima [ ] Ditolak
Alamat	{text} [a-z, A-Z, 0-9, simbol]	-	Jl. Pegadaian 2 Blok A/39	Dapat memasukkan alamat pada *field* alamat	Dapat memasukkan alamat pada *field* alamat	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						

Butir Uji	Masukan yang Valid	Masukan yang Tidak Valid	Data yang Dimasukkan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Nama	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "nama tidak boleh kosong"	Dapat menampilkan pesan "nama tidak boleh kosong"	[✓] Diterima [ ] Ditolak
Alamat	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "alamat tidak boleh kosong"	Dapat menampilkan pesan "alamat tidak boleh kosong"	[✓] Diterima [ ] Ditolak

c. Data Pembelian Bahan Baku

Tabel 4.28 merupakan skenario pengujian menggunakan teknik *Equivalence Partitioning Testing* pada bagian pembelian bahan baku Departemen Distribusi.

Tabel 4.28 *Equivalence Partitioning Testing*-Data Pembelian Bahan Baku

Kasus dan Hasil Uji (Data Valid)						
Butir Uji	**Masukan yang Valid**	**Masukan yang Tidak Valid**	**Data yang Dimasukkan**	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tanggal	{text} [a-z, A-Z, 0-9, simbol]	-	2 feb 2013	Dapat memasukkan tanggal pada *field* tanggal	Dapat memasukkan tanggal pada *field* tanggal	[✓] Diterima [ ] Ditolak
Nama *Supplier*	{text} [a-z, A-Z, 0-9, simbol]	-	PT Mekar Kencana	Dapat memilih *supplier* pada *combo box supplier*	Dapat memilih *supplier* pada *combo box supplier*	[✓] Diterima [ ] Ditolak
Nama Bahan Baku	{text} [a-z, A-Z, 0-9, simbol]	-	Tepung Cakra	Dapat memilih bahan baku pada *combo box* bahan baku	Dapat memilih bahan baku pada *combo box* bahan baku	[✓] Diterima [ ] Ditolak
Minimal Pembelian	{numeric} [0-9]	-	125000000	Dapat memasukkan jumlah minimal pada *field* jumlah minimal	Dapat memasukkan jumlah minimal pada *field* jumlah minimal	[✓] Diterima [ ] Ditolak
Satuan	{text} [a-z, A-Z, 0-9, simbol]	-	Kg	Dapat memilih satuan pada *combo box* satuan	Dapat memilih satuan pada *combo box* satuan	[✓] Diterima [ ] Ditolak
Harga Satuan	{numeric} [0-9]	-	160000	Dapat memasukkan harga pada *field* harga	Dapat memasukkan harga pada *field* harga	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Valid)						

Butir Uji	Masukan yang Valid	Masukan yang Tidak Valid	Data yang Dimasukkan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Tanggal	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "tangal tidak boleh kosong"	Dapat menampilkan pesan "tangal tidak boleh kosong"	[✓] Diterima [ ] Ditolak
Nama *Supplier*	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih *supplier*"	Dapat menampilkan pesan "pilih *supplier*"	[✓] Diterima [ ] Ditolak
Nama Bahan Baku	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih bahan baku"	Dapat menampilkan pesan "pilih bahan baku"	[✓] Diterima [ ] Ditolak
Minimal Pembelian	{numeric} [0-9]	-	Null	Dapat menampilkan pesan "minimal pembelian tidak boleh kosong"	Dapat menampilkan pesan "minimal pembelian tidak boleh kosong"	[✓] Diterima [ ] Ditolak
Satuan	{text} [a-z, A-Z, 0-9, simbol]	-	Null	Dapat menampilkan pesan "pilih satuan"	Dapat menampilkan pesan "pilih satuan"	[✓] Diterima [ ] Ditolak
Harga Satuan	{numeric} [0-9]	-	Null	Dapat menampilkan pesan "harga tidak boleh kosong"	Dapat menampilkan pesan "harga tidak boleh kosong"	[✓] Diterima [ ] Ditolak

4.2.2.2 *Cause-Effect Relationalship Testing*

Teknik pengujian *Cause-Effect Relationalship* adalah pengujian yang dilakukan dengan mengkombinasikan beberapa masukan yang dijadikan sebagai kasus uji dan digambarkan dalam bentuk tabel keputusan. Untuk pengujian *Cause-Effect Relationalship* dapat dilihat pada tabel berikut ini.

1. *Cause-Effect Relationalship Testing* Kepala Administrasi

a. Login

Input States.

1. *Username* memenuhi kondisi yang ditentukan.
2. *Password* memenuhi kondisi yang ditentukan.

Output States.

A. Berhasil melakukan *login* dan masuk ke *form* Kepala Administrasi.

B. Menampilkan kesalahan dari *username* yang dimasukan.

C. Menampilkan kesalahan dari *password* yang dimasukan.

Tabel 4.29 *Cause-Effect Relationalship Testing Login* Kepala Administrasi

Test Case / ***Cause Effect***	1	2	3
1	1	1	0
2	1	0	1
A	1	0	0
B	0	1	0
C	0	0	1

b. Data Users

Input States.

1. Nama memenuhi kondisi yang ditentukan.
2. Jabatan memenuhi kondisi yang ditentukan.
3. *Username* memenuhi kondisi yang ditentukan.
4. *Password* memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari nama yang dimasukan.

C. Menampilkan kesalahan dari jabatan yang dimasukan.

D. Menampilkan kesalahan dari *username* yang dimasukan.

E. Menampilkan kesalahan dari *password* yang dimasukan.

Tabel 4.30 *Cause-Effect Relationalship Testing* Data Users

Test Case / *Cause Effect*	1	2	3	4	5
1	1	1	1	1	0
2	1	1	1	0	1
3	1	1	0	1	1
4	1	0	1	1	1
A	1	0	0	0	0
B	0	0	0	0	1
C	0	0	0	1	0
D	0	0	1	0	0
E	0	1	0	0	0

2. *Cause – Effect Relationalship Testing* Departemen Gudang

a. Login

Input States.

1. *Username* memenuhi kondisi yang ditentukan.

2. *Password* memenuhi kondisi yang ditentukan.

Output States.

A. Berhasil melakukan *login* dan masuk ke form Departemen Gudang.

B. Menampilkan kesalahan dari *username* yang dimasukan.

C. Menampilkan kesalahan dari *password* yang dimasukan.

Tabel 4.31 *Cause-Effect Relationalship Testing Login* Departemen Gudang

Test Case / *Cause Effect*	1	2	3
1	1	1	0
2	1	0	1
A	1	0	0
B	0	1	0
C	0	0	1

b. Data Bahan Baku

Input States.

1. Kategori memenuhi kondisi yang ditentukan.
2. Jenis memenuhi kondisi yang ditentukan.
3. Jumlah memenuhi kondisi yang ditentukan.
4. Satuan memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari kategori yang dimasukan.

C. Menampilkan kesalahan dari jenis yang dimasukan.

D. Menampilkan kesalahan dari jumlah yang dimasukan.

E. Menampilkan kesalahan dari satuan yang dimasukan.

Tabel 4.32 *Cause-Effect Relationalship Testing* Data Bahan Baku

Test Case / ***Cause Effect***	1	2	3	4	5
1	1	1	1	1	0
2	1	1	1	0	1
3	1	1	0	1	1
4	1	0	1	1	1
A	1	0	0	0	0
B	0	0	0	0	1
C	0	0	0	1	0
D	0	0	1	0	0
E	0	1	0	0	0

c. Data Kategori Bahan Baku

Input States.

1. Nama memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari nama yang dimasukan.

Tabel 4.33 *Cause-Effect Relationalship Testing* Kategori Bahan Baku Departemen Gudang

Test Case / *Cause Effect*	1	2
1	1	0
A	1	0
B	0	1

d. Data Jenis Bahan Baku

Input States.

1. Kategori memenuhi kondisi yang ditentukan.

2. Nama memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam database.

B. Menampilkan kesalahan dari kategori yang dimasukan.

C. Menampilkan kesalahan dari nama yang dimasukan.

Tabel 4.34 *Cause-Effect Relationalship Testing* Data Jenis Bahan Baku

Test Case / *Cause Effect*	1	2	3
1	1	1	0
2	1	0	1
A	1	0	0
B	0	0	1
C	0	1	0

3. *Cause-Effect Relationalship Testing* Departemen Distribusi

a. Login

Input States.

1. *Username* memenuhi kondisi yang ditentukan.

2. *Password* memenuhi kondisi yang ditentukan.

Output States.

A. Berhasil melakukan login dan masuk ke form Departemen Distribusi.

B. Menampilkan kesalahan dari *username* yang dimasukan.

C. Menampilkan kesalahan dari *password* yang dimasukan.

Tabel 4.35 *Cause-Effect Relationalship Testing* Login Departemen Distribusi

Test Case / ***Cause Effect***	1	2	3
1	1	1	0
2	1	0	1
A	1	0	0
B	0	1	0
C	0	0	1

b. Data Produk

Input States.

1. Jenis memenuhi kondisi yang ditentukan.

2. Nama memenuhi kondisi yang ditentukan.

3. Harga memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari jenis yang dimasukan.

C. Menampilkan kesalahan dari nama yang dimasukan.

D. Menampilkan kesalahan dari harga yang dimasukan.

Tabel 4.36 *Cause-Effect Relationalship Testing* Data Produk

Test Case / *Cause Effect*	1	2	3	4
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	0	1	1
A	1	0	0	0
B	0	0	0	1
C	0	0	1	0
D	0	1	0	0

c. Data Jenis Produk

Input States.

1. Nama memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari nama yang dimasukan.

Tabel 4.37 *Cause-Effect Relationalship Testing* Jenis Produk

Test Case / *Cause Effect*	1	2
1	1	0
A	1	0
B	0	1

d. Data Penjualan Produk

Input States.

1. Tanggal memenuhi kondisi yang ditentukan.

2. Nama memenuhi kondisi yang ditentukan.

3. Jumlah memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari tanggal yang dimasukan.

C. Menampilkan kesalahan dari nama yang dimasukan.

D. Menampilkan kesalahan dari jumlah yang dimasukan.

Tabel 4.38 *Cause-Effect Relationalship Testing* Data Penjualan Produk

Test Case / *Cause Effect*	1	2	3	4
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	0	1	1
A	1	0	0	0
B	0	0	0	1
C	0	0	1	0
D	0	1	0	0

e. Data Peramalan

Input States.

1. idProduk memenuhi kondisi yang ditentukan.

2. Tanggal memenuhi kondisi yang ditentukan.

3. Jumlah memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari idProduk yang dimasukan.

C. Menampilkan kesalahan dari tanggal yang dimasukan.

D. Menampilkan kesalahan dari jumlah yang dimasukan.

Tabel 4.39 *Cause-Effect Relationalship Testing* Data Peramalan

Test Case / *Cause Effect*	1	2	3	4
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	0	1	1
A	1	0	0	0
B	0	0	0	1
C	0	0	1	0
D	0	1	0	0

4. *Cause-Effect Relationalship Testing* Departemen PPIC

a. Login

Input States.

1. *Username* memenuhi kondisi yang ditentukan.

2. *Password* memenuhi kondisi yang ditentukan.

Output States.

A. Berhasil melakukan *login* dan masuk ke form Departemen PPIC.

B. Menampilkan kesalahan dari *username* yang dimasukan.

C. Menampilkan kesalahan dari *password* yang dimasukan.

Tabel 4.40 *Cause-Effect Relationalship Testing Login* Departemen PPIC

Test Case / ***Cause Effect***	1	2	3
1	1	1	0
2	1	0	1
A	1	0	0
B	0	1	0
C	0	0	1

b. Data Resep

Input States.

1. Nama memenuhi kondisi yang ditentukan.

2. Jumlah memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari nama yang dimasukan.

C. Menampilkan kesalahan dari jumlah yang dimasukan.

Tabel 4.41 *Cause-Effect Relationalship Testing* Data Resep

Test Case / *Cause Effect*	1	2	3
1	1	1	0
2	1	0	1
A	1	0	0
B	0	0	1
C	0	1	0

c. Data Detail Peramalan

Input States.

1. Nama resep memenuhi kondisi yang ditentukan.
2. Nama bahan memenuhi kondisi yang ditentukan.
3. Jumlah bahan memenuhi kondisi yang ditentukan.
4. Satuan memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari nama resep yang dimasukan.

C. Menampilkan kesalahan dari nama bahan yang dimasukan

D. Menampilkan kesalahan dari jumlah bahan yang dimasukan.

E. Menampilkan kesalahan dari satuan yang dimasukan.

Tabel 4.42 *Cause-Effect Relationalship Testing* Data Detail Peramalan

Test Case / *Cause Effect*	1	2	3	4	5
1	1	1	1	1	0
2	1	1	1	0	1
3	1	1	0	1	1
4	1	0	1	1	1
A	1	0	0	0	0
B	0	0	0	0	1
C	0	0	0	1	0
D	0	0	1	0	0
E	0	1	0	0	0

5. *Cause-Effect Relationalship Testing* Departemen *Purchasing*

a. Login

Input States.

1. *Username* memenuhi kondisi yang ditentukan.

2. *Password* memenuhi kondisi yang ditentukan.

Output States.

A. Berhasil melakukan *login* dan masuk ke form Departemen *Purchasing*.

B. Menampilkan kesalahan dari *username* yang dimasukan.

C. Menampilkan kesalahan dari *password* yang dimasukan.

Tabel 4.43 *Cause-Effect Relationalship Testing Login* Departemen *Purchasing*

Test Case / ***Cause Effect***	1	2	3
1	1	1	0
2	1	0	1
A	1	0	0
B	0	1	0
C	0	0	1

b. Data Supplier

Input States.

1. Nama memenuhi kondisi yang ditentukan.

2. Alamat memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari nama yang dimasukan.

C. Menampilkan kesalahan dari alamat yang dimasukan.

Tabel 4.44 *Cause-Effect Relationalship Testing* Data Supplier

Test Case / *Cause Effect*	1	2	3
1	1	1	0
2	1	0	1
A	1	0	0
B	0	0	1
C	0	1	0

c. Data Pembelian Bahan Baku

Input States.

1. Tanggal memenuhi kondisi yang ditentukan.
2. Nama *supplier* memenuhi kondisi yang ditentukan.
3. Nama bahan memenuhi kondisi yang ditentukan.
4. Minimal pembelian memenuhi kondisi yang ditentukan.
5. Harga satuan memenuhi kondisi yang ditentukan.
6. Satuan memenuhi kondisi yang ditentukan.

Output States.

A. Data tersimpan dalam *database*.

B. Menampilkan kesalahan dari tanggal yang dimasukan.

C. Menampilkan kesalahan dari nama *supplier* yang dimasukan

D. Menampilkan kesalahan dari nama bahan yang dimasukan.

E. Menampilkan kesalahan dari minimal pembelian yang dimasukan.

F. Menampilkan kesalahan dari harga satuan yang dimasukan.

G. Menampilkan kesalahan dari satuan yang dimasukan.

Tabel 4.45 *Cause-Effect Relationalship Testing* Data Pembelian Bahan Baku

Test Case / *Cause Effect*	1	2	3	4	5	6	7
1	1	1	1	1	1	1	0
2	1	1	1	1	1	0	1
3	1	1	1	1	0	1	1
4	1	1	1	0	1	1	1
5	1	1	0	1	1	1	1
6	1	0	1	1	1	1	1
A	1	0	0	0	0	0	0
B	0	0	0	0	0	0	1
C	0	0	0	0	0	1	0
D	0	0	0	0	1	0	0
E	0	0	0	1	0	0	0
F	0	0	1	0	0	0	0
G	0	1	0	0	0	0	0

4.2.3 Pengujian *Beta*

Pengujian *beta* dilakukan untuk mengetahui sampai sejauh mana kualitas dari sistem yang sudah dibangun apakah sudah sesuai yang diharapkan atau tidak. Pengujian *beta* dilakukan dengan melakukan wawancara terhadap karyawan Holland Bakery yang terdiri dari Kepala Administrasi, Staff Departemen Gudang, Staff Deparetmen Distribusi, Staff Departemen PPIC, Staff Departemen *Purchasing*.

Wawancara dilakukan berdasarkan pertanyaan yang diajukan adalah sebagai berikut.

1. Apakah aplikasi ini telah membantu mempermudah dalam melakukan proses memperkirakan jumlah produk yang akan diproduksi?
2. Apakah aplikasi ini telah membantu mempermudah dalam pengolahan data bahan baku?
3. Apakah aplikasi ini telah membantu mempermudah dalam melakukan perhitungan jumlah bahan baku yang akan digunakan untuk melakukan produksi suatu produk?
4. Apakah aplikasi ini secara garis besar mudah digunakan?

Adapun jawaban dari masing-masing karyawan sesuai jabatan dapat dilihat pada tabel berikut.

Tabel 4.46 Hasil Wawancara Karyawan Holland Bakery

Pengguna	Jawaban Pertanyaan
Kepala Administrasi	1. Beliau berkata bahwa aplikasi ini mudah digunakan.
Staff Departemen Gudang	1. Beliau berkata bahwa aplikasi sudah membantu dalam memperoleh informasi data bahan baku. 2. Beliau berkata bahwa aplikasi ini cukup mudah untuk digunakan.
Staff Deparetmen Distribusi	1. Beliau berkata ini aplikasi sudah cukup membantu dalam melakukan perhitungan memperkirakan jumlah produk yang akan diproduksi. 2. Beliau berkata bahwa aplikasi ini sudah cukup membantu untuk mengetahui informasi data penjualan produk. 3. Beliau berkata bahwa setelah dilihat dari aplikasinya, aplikasi ini mudah digunakan dan jelas penggunaannya.
Staff Departemen PPIC	1. Beliau berkata bahwa aplikasi ini sudah cukup Membantu dalam melakukan perhitungan bahan baku yang dibutuhkan . 2. Beliau berkata bahwa aplikasi ini mudah digunakan.
Staff Departemen *Purchasing*	1. Beliau berkata bahwa aplikasi ini sudah membantu dan mudah digunakan.

4.2.4 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian *black box* dan *beta* yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut.

1. Sistem informasi yang dibangun membantu mempermudah dalam melakukan perhitungan memperkirakan jumlah produk yang akan diproduksi.
2. Sistem informasi yang dibangun membantu mempermudah dalam pengolahan data bahan baku.
3. Sistem informasi yang dibangun membantu mempermudah dalam melakukan perhitungan bahan baku yang akan digunakan untuk memproduksi suatu produk.
4. Sistem informasi yang dibangun mudah digunakan oleh pengguna.