BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi sistem merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin, serta penerapan perangkat lunak pada keadaan yang sesungguhnya. Implementasi diterapkan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman yang akan digunakan.

4.2 Perangkat Lunak Pembangun

Perangkat lunak yang digunakan untuk mengimplementasikan sistem *data warehouse* adalah sebagai berikut :

1. Sistem Operasi *Windows XP Profesional SP 2*.
2. *WampServer* untuk penyimpanan *database*.
3. *Borland Delphi 7* untuk implementasi program yang akan dibangun.

4.2.1 Perangkat Keras Pembangun

Kebutuhan minimum perangkat keras (*hardware*) yang diperlukan untuk mengimplementasikan program aplikasi yang dibuat adalah perangkat keras komputer PC dengan spesifikasi yang disebutkan dibawah ini. Semakin tinggi

spesifikasi komputer yang digunakan untuk menjalankan aplikasi, akan semakin baik. Kebutuhan minimumnya yaitu :

1. Processor : CPU 1.7 GHz
2. Memory : RAM 128 MB
3. Hardisk : 20 GB (Free Space)
4. VGA : 32 MB

4.2.2 Implementasi *Database*

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

Tabel kategori :

```
CREATE TABLE `kategori` (
  `id_kategori`  int(10)  unsigned  NOT  NULL
AUTO_INCREMENT,
  `kode_kategori` varchar(3) DEFAULT NULL,
  `nama_kategori` varchar(20) NOT NULL,
  PRIMARY KEY (`id_kategori`),
  UNIQUE KEY `un_kategori` (`kode_kategori`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel warna :

```
CREATE TABLE `warna` (
  `id_warna`   int(11)   unsigned   NOT   NULL
AUTO_INCREMENT,
  `kode_warna` varchar(2) NOT NULL,
  `nama_warna` varchar(20) NOT NULL,
  PRIMARY KEY (`id_warna`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel desain :

```
CREATE TABLE `desain` (
  `id_desain`   int(11)   unsigned   NOT   NULL
AUTO_INCREMENT,
  `id_grup_musik` int(11) unsigned NOT NULL,
  `nama_desain` varchar(50) NOT NULL,
  `gambar_desain` varchar(50) NOT NULL,
  PRIMARY KEY (`id_desain`),
  KEY `FK_desain_grup_musik` (`id_grup_musik`),
```

```
  CONSTRAINT `FK_desain_grup_musik` FOREIGN KEY
(`id_grup_musik`)     REFERENCES     `grup_musik`
(`id_grup_musik`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel grup_musik :

```
CREATE TABLE `grup_musik` (
  `id_grup_musik`  int(11)  unsigned  NOT  NULL
AUTO_INCREMENT,
  `kode_grup_musik` varchar(3) NOT NULL,
  `nama_grup_musik` varchar(20) NOT NULL,
  PRIMARY KEY (`id_grup_musik`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel barang :

```
CREATE TABLE `barang` (
  `id_barang`   int(11)   unsigned   NOT   NULL
AUTO_INCREMENT,
  `id_kategori` int(11) unsigned NOT NULL,
  `id_warna` int(11) unsigned NOT NULL,
  `id_desain` int(11) unsigned NOT NULL,
  `kode_barang` varchar(9) DEFAULT NULL,
  `ukuran` varchar(2) NOT NULL,
  `gambar_barang` varchar(100) NOT NULL,
  `harga` int(11) NOT NULL,
  PRIMARY KEY (`id_barang`),
  UNIQUE KEY `U_kode_barang` (`kode_barang`),
  KEY `FK_barang_kategori` (`id_kategori`),
  KEY `FK_barang_warna` (`id_warna`),
  KEY `FK_barang_desain` (`id_desain`),
  CONSTRAINT   `FK_barang_desain`   FOREIGN   KEY
(`id_desain`) REFERENCES `desain` (`id_desain`)
ON UPDATE CASCADE,
  CONSTRAINT  `FK_barang_kategori`  FOREIGN  KEY
(`id_kategori`)       REFERENCES       `kategori`
(`id_kategori`) ON UPDATE CASCADE,
  CONSTRAINT   `FK_barang_warna`   FOREIGN   KEY
(`id_warna`) REFERENCES `warna` (`id_warna`) ON
UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel tanggal :

```
CREATE TABLE `tanggal` (
  `id_tanggal`   int(11)   unsigned   NOT   NULL
AUTO_INCREMENT,
  `tanggal` int(2) unsigned NOT NULL,
  `bulan`
enum('Januari','Pebruari','Maret','April','Mei',
'Juni','Juli','Agustus','September','Oktober','N
ovember','Desember') NOT NULL,
  `tahun` int(4) unsigned NOT NULL,
```

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

Tabel propinsi :

```
CREATE TABLE `propinsi` (
  `id_propinsi`  int(11)  unsigned  NOT  NULL
AUTO_INCREMENT,
  `nama_propinsi` varchar(50) NOT NULL,
  PRIMARY KEY (`id_propinsi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel kota :

```
CREATE TABLE `kota` (
  `id_kota`   int(11)   unsigned   NOT   NULL
AUTO_INCREMENT,
  `id_propinsi` int(11) unsigned NOT NULL,
  `nama_kota` varchar(30) NOT NULL,
  PRIMARY KEY (`id_kota`),
  KEY `FK_kota_propinsi` (`id_propinsi`),
  CONSTRAINT  `FK_kota_propinsi`  FOREIGN  KEY
(`id_propinsi`)      REFERENCES      `propinsi`
(`id_propinsi`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel toko :

```
CREATE TABLE `toko` (
  `id_toko`   int(11)   unsigned   NOT   NULL
AUTO_INCREMENT,
  `id_kota` int(11) unsigned NOT NULL,
  `nama_toko` varchar(50) NOT NULL,
  `alamat_toko` varchar(50) NOT NULL,
  PRIMARY KEY (`id_toko`),
  KEY `FK_toko_kota` (`id_kota`),
  CONSTRAINT   `FK_toko_kota`   FOREIGN   KEY
(`id_kota`)  REFERENCES  `kota`  (`id_kota`)  ON
UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel penjualan :

```
CREATE TABLE `penjualan` (
  `id_penjualan`  int(11)  unsigned  NOT  NULL
AUTO_INCREMENT,
  `id_barang` int(11) unsigned NOT NULL,
  `id_toko` int(11) unsigned NOT NULL,
  `id_tanggal` int(11) unsigned NOT NULL,
  `jumlah` int(11) unsigned NOT NULL,
  `total` bigint(20) unsigned NOT NULL,
  PRIMARY KEY (`id_penjualan`),
  KEY `FK_penjualan_barang` (`id_barang`),
  KEY `FK_penjualan_toko` (`id_toko`),
```

```
  KEY `FK_penjualan_tanggal` (`id_tanggal`),
  CONSTRAINT `FK_penjualan_barang` FOREIGN KEY
(`id_barang`) REFERENCES `barang` (`id_barang`)
ON UPDATE CASCADE,
  CONSTRAINT `FK_penjualan_tanggal` FOREIGN KEY
(`id_tanggal`)       REFERENCES       `tanggal`
(`id_tanggal`) ON UPDATE CASCADE,
  CONSTRAINT `FK_penjualan_toko` FOREIGN KEY
(`id_toko`) REFERENCES `toko` (`id_toko`) ON
UPDATE CASCADE
)  ENGINE=InnoDB  AUTO_INCREMENT=226  DEFAULT
CHARSET=latin1;
```

Tabel penerimaan :

```
CREATE TABLE `penerimaan` (
  `id_penerimaan` int(11) unsigned NOT NULL
AUTO_INCREMENT,
  `id_barang` int(11) unsigned NOT NULL,
  `id_toko` int(11) unsigned NOT NULL,
  `id_tanggal` int(11) unsigned NOT NULL,
  `jumlah_penerimaan` int(11) unsigned NOT NULL,
  PRIMARY KEY (`id_penerimaan`),
  KEY `FK_terima_barang` (`id_barang`),
  KEY `FK_penerimaan_toko` (`id_toko`),
  KEY `FK_penerimaan_tanggal` (`id_tanggal`),
  CONSTRAINT `FK_penerimaan_tanggal` FOREIGN KEY
(`id_tanggal`)       REFERENCES       `tanggal`
(`id_tanggal`) ON UPDATE CASCADE,
  CONSTRAINT `FK_penerimaan_toko` FOREIGN KEY
(`id_toko`) REFERENCES `toko` (`id_toko`) ON
UPDATE CASCADE,
  CONSTRAINT `FK_terima_barang` FOREIGN KEY
(`id_barang`) REFERENCES `barang` (`id_barang`)
ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabel pengembalian :

```
CREATE TABLE `pengembalian` (
  `id_pengembalian` int(11) unsigned NOT NULL
AUTO_INCREMENT,
  `id_barang` int(11) unsigned NOT NULL,
  `id_toko` int(11) unsigned NOT NULL,
  `id_tanggal` int(11) unsigned NOT NULL,
  `jumlah_pengembalian` int(11) unsigned NOT
NULL,
  PRIMARY KEY (`id_pengembalian`),
  KEY `FK_terima_barang` (`id_barang`),
  KEY `FK_pengembalian_toko` (`id_toko`),
  KEY `FK_pengembalian_tanggal` (`id_tanggal`),
  CONSTRAINT `FK_pengembalian_barang` FOREIGN
KEY    (`id_barang`)    REFERENCES    `barang`
(`id_barang`) ON UPDATE CASCADE,
```

```
  CONSTRAINT `FK_pengembalian_tanggal` FOREIGN
KEY (`id_tanggal`) REFERENCES `tanggal`
(`id_tanggal`) ON UPDATE CASCADE,
  CONSTRAINT `FK_pengembalian_toko` FOREIGN KEY
(`id_toko`) REFERENCES `toko` (`id_toko`) ON
UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.2.3 Implementasi Perangkat Lunak

Pembuatan perangkat lunak dilakukan dengan menggunakan *Borland Delphi 7*.

Tabel 4.1 Tabel Implementasi Perangkat Lunak

No. Proses	Nama Proses	Nama File
1	*ETL*	UFUtama.pas
2.1	Pemilihan Filter	UFUtama.pas
2.2	Analisis Data	UFUtama.pas
2.3	Hasil Analisis Data	UFHasilAnalisis.pas
2.4	Cetak Laporan	UFLaporan.pas
3	Pengaturan	UFUtama.pas

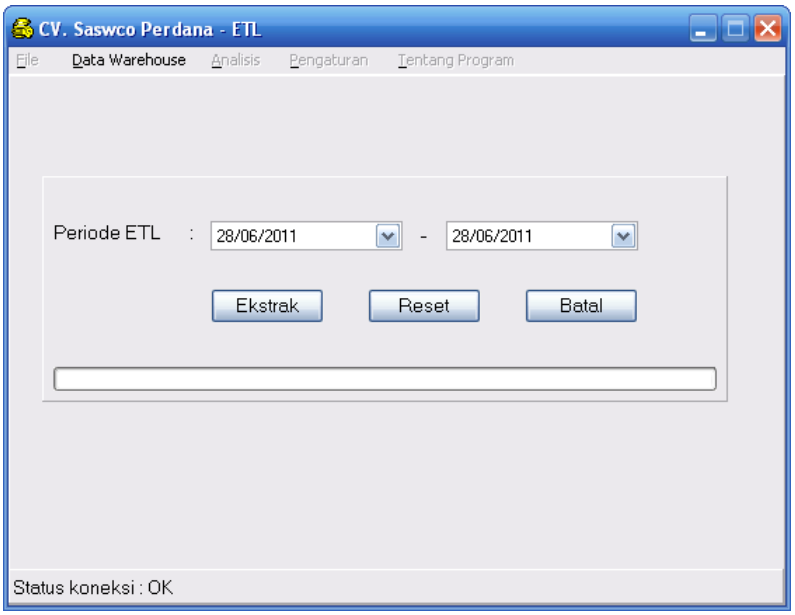
4.3 Penggunaan Aplikasi

1. Pertama jalankan *Warehouse.exe*. Kemudian akan muncul tampilan halaman utama seperti pada gambar 4.1.



Gambar 4.1 Form Halaman Utama

2. Untuk melakukan proses *ETL* pilih menu *Data Warehouse* kemudia pilih *ETL*. Setelah itu, tampilan halaman utama akan berubah menjadi seperti gambar 4.2.

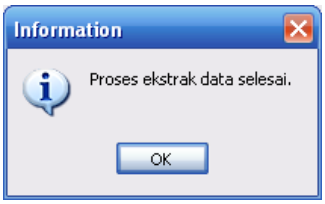


Gambar 4.2 Form *ETL*

3. Setelah form *ETL* muncul, masukkan periode *ETL* kemudian tekan tombol *Ekstrak*. Tombol *Reset* berfungsi untuk mereset inputan

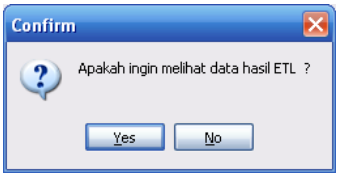
periode *ETL*, sedangkan tombol kembali berfungsi untuk kembali ke halaman utama.

4. Jika anda melakukan proses *ETL*, dan proses telah selesai, maka akan muncul pesan seperti gambar 4.3.



Gambar 4.3 Pesan *ETL* Telah selesai

5. Untuk melanjutkan tekan *OK*. Setelah itu muncul pesan yang akan menanyakan apakah user ingin melihat data hasil ekstraksi seperti pada gambar 4.4. Jika user memilih *No*, maka form halaman utama akan muncul. Jika user memilih *Yes*, maka muncul form seperti pada gambar 4.5.

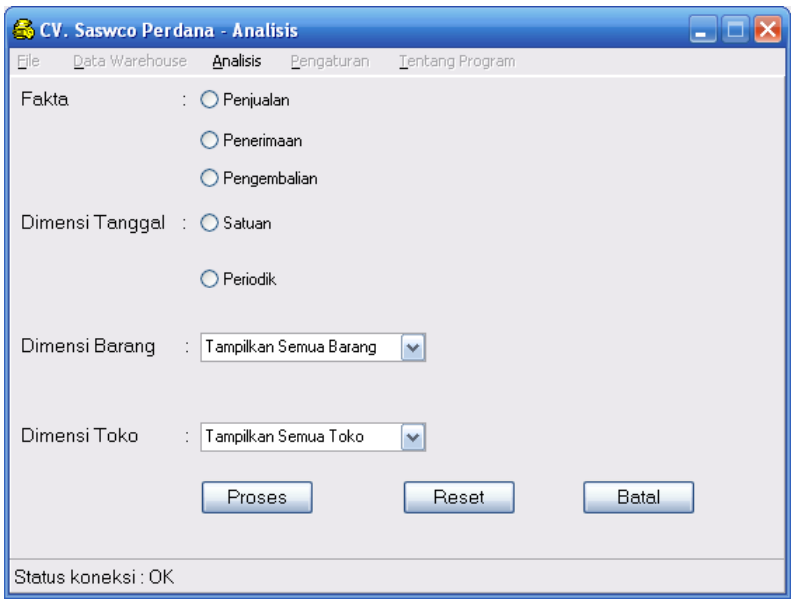


Gambar 4.4 Konfirmasi ingin melihat Hasil *ETL* atau tidak



Gambar 4.5 Form hasil *ETL*

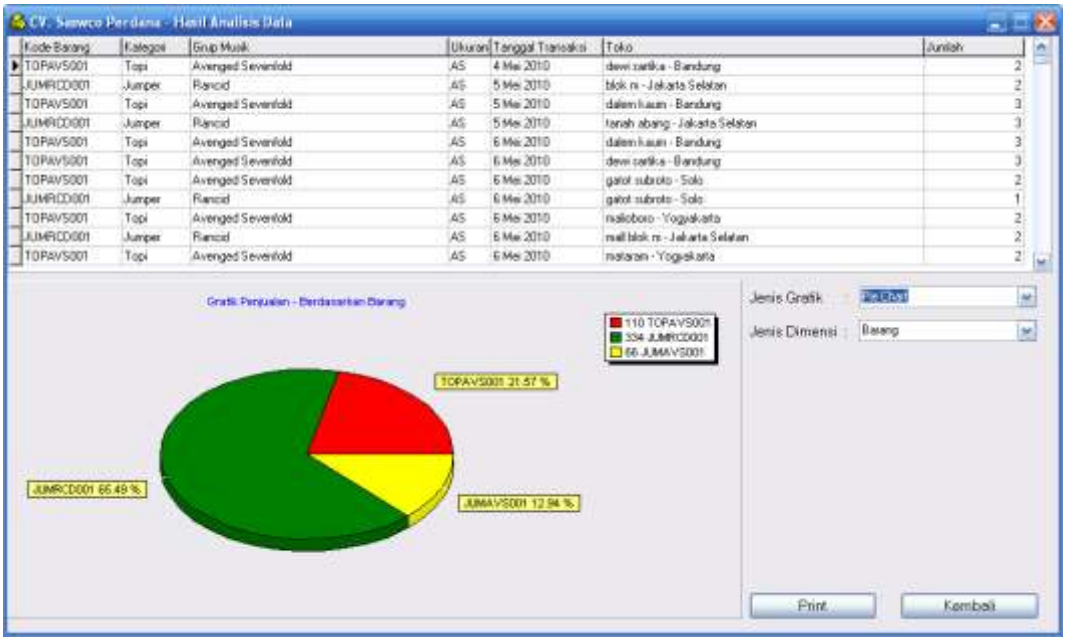
6. Jika user ingin melakukan proses analisis, pada form halaman utama, pilih menu *Analisis*. Jika memilih menu *Analisis*, maka akan muncul form seperti pada gambar 4.6.



Gambar 4.6 Form Analisis

7. Bagian fakta adalah pilihan tabel fakta yang ingin dianalisis. Bagian dimensi berguna sebagai filter data yang akan ditampilkan. Tombol

Proses melakukan proses analisis dan menampilkan hasil analisis tersebut. Jika proses analisis telah selesai, maka akan muncul form seperti gambar 4.7. Tombol *Reset* berguna untuk menghapus semua filter yang telah dimasukkan, sedangkan tombol batal berguna untuk kembali ke form halaman utama.



Gambar 4.7 Form hasil analisis

8. Pada form analisis akan muncul data hasil analisis beserta grafik berdasarkan data yang diperoleh. Tampilan grafik yang bisa dipilih ada tiga jenis, pie chart, grafik batang, dan grafik garis. Grafik juga bisa ditampilkan berdasarkan dimensi barang, tanggal, maupun toko. Tombol *Print* digunakan untuk mencetak data hasil analisis. Sedangkan tombol kembali untuk kembali ke form analisis.

4.4 Pengujian

Teknik pengujian yang digunakan kali ini adalah pengujian *Alpha* dan *Beta*. Pengujian *Alpha* akan dilakukan dengan metode *Black Box*, sedangkan pengujian *Beta* dilakukan dengan wawancara.

4.4.1 Pengujian *Alpha*

Pengujian *Alpha* digunakan untuk menguji apakah perangkat lunak sudah bebas dari kesalahan pemrograman. Berikut adalah rencana pengujian *alpha* :

Tabel 4.2 Tabel rencana pengujian

Kelas Uji	Butir Uji	Jenis Pengujian
ETL	Inputan periode *ETL*	*Black box*
Analisis	Pemilihan fakta dan dimensi	*Black box*
Pengaturan	Pengisian data setting	*Black box*

4.4.1.1 Pengujian *ETL*

Pegujian proses *ETL* dilakukan untuk melihat apakah proses dapat berjalan sesuai dengan periode masukan dari user.

Tabel 4.3 Pengujian *ETL*

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Periode ETL	Dapat melakukan proses ETL sesuai dengan batasan periode yang dimasukkan.	Hasil ETL sesuai dengan periode yang dimasukkan user	[√] diterima [ ] ditolak

4.4.1.2 Pengujian Analisis

Pengujian bagian analisis dilakukan dengan menguji fakta dan dimensi yang akan dijadikan parameter analisis.

Tabel 4.4 Tabel Pengujian Analisis

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Fakta : penjualan Dimensi : -	Menampilkan semua data penjualan	Tampil semua data penjualan	[√] diterima [ ] ditolak
Fakta : penjualan Dimensi : tanggal	Menampilkan data penjualan yang difilter berdasarkan tanggal	Tampil data penjualan yang sudah difilter berdasarkan tanggal	[√] diterima [ ] ditolak
Grafik: pie chart Dimensi:barang	Menampilkan grafik barang berjenis pie chart berdasarkan data hasil analisis	Tampil grafik barang berjenis pie chart sesuai dengan data hasil analisis	[√] diterima [ ] ditolak
Grafik: batang Dimensi:barang	Menampilkan grafik barang berjenis batang berdasarkan data hasil analisis	Tampil barang berjenis grafik batang sesuai dengan data hasil analisis	[√] diterima [ ] ditolak
Grafik: garis Dimensi:barang	Menampilkan grafik barang berjenis garis berdasarkan data hasil analisis	Tampil barang berjenis grafik garis sesuai dengan data hasil analisis	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Fakta: -	Proses analisis	Proses analisis	[√] diterima

Dimensi: -	dibatalkan dan muncul pesan “Pilih fakta”.	dibatalkan kemudian muncul pesan “Pilih fakta”	[ ] ditolak

4.4.1.3 Pengujian Pengaturan

Pengujian bagian pengaturan dilakukan dengan menguji inputan user.

Tabel 4.5 Tabel pengujian pengaturan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Host : 127.0.0.1 Port : 3306 User : root Password : -	Menampilkan status bahwa koneksi ke database berhasil	Menampilkan pesan di status koneksi : “OK“	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Host : - Port : - User : - Password : -	Menampilkan pesan untuk meminta user mengisi seluruh inputan	Menampilkan pesan untuk melengkapi inputan.	[√] diterima [ ] ditolak
Host : 127.0.0.1 Port : 3306 User : test Password : -	Menampilkan pesan bahwa koneksi ke database gagal	Menampilkan pesan bahwa koneksi ke database gagal dan meminta user untuk memeriksa pengaturan	[√] diterima [ ] ditolak

4.4.2 Kesimpulan Pengujian *Alpha*

Berdasarkan hasil pengujian *Alpha* yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi sudah berjalan cukup maksimal, tetapi tidak menutup kemungkinan dapat terjadi kesalahan suatu saat, pada saat aplikasi digunakan, sehingga membutuhkan proses *maintenance* untuk lebih mengetahui kekurangan dari aplikasi.

4.4.3 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara langsung pada perusahaan yaitu untuk mengetahui kualitas dari *software* yang telah dibuat, apakah sudah sesuai dengan yang diharapkan atau belum, pengujian dilakukan dengan melakukan wawancara kepada pihak manager.

4.4.4 Wawancara

Wawancara dilakukan dengan mengajukan empat pertanyaan utama. Berikut adalah kelima pertanyaan utama yang diajukan pada saat wawancara :

1. Apakah *data warehouse* ini memudahkan dalam proses analisis data ?
2. Apakah *data warehouse* ini sudah menyajikan informasi yang diperlukan oleh pihak manager ?
3. Apakah informasi yang diperoleh sudah dapat membantu dalam menentukan strategi produksi dan distribusi ?
4. Bagaimana pendapat pengguna tentang tampilan antarmuka pada aplikasi ini ?

4.4.5 Hasil Wawancara

Berikut adalah jawaban dari Ir. Dedi Supriyadi, direktur CV. Saswco Perdana, terhadap keempat pertanyaan utama diatas :

1. Apakah *data warehouse* ini memudahkan dalam proses analisis data ? "*Data warehouse* ini cukup membantu dalam proses analisis data penjualan, penerimaan, dan pengembalian. *Data warehouse* ini ternyata juga mempercepat proses analisis."
2. Apakah *data warehouse* ini sudah menyajikan informasi yang diperlukan oleh pihak manager ? " Untuk saat ini iya. Karena kita masih belum tahu kebutuhan untuk beberapa tahun kedepan. Tetapi untuk saat ini, sistem ini sudah memenuhi kebutuhan kami."
3. Apakah informasi yang diperoleh sudah dapat membantu dalam menentukan strategi produksi dan distribusi ? "Dengan bantuan sistem ini, proses penentuan produksi dan distribusi sudah jadi lebih mudah dan akurat lagi. Apalagi grafik yang ditampilkan sudah bersifat dinamis."
4. Bagaimana pendapat pengguna tentang tampilan antarmuka pada aplikasi ini ? "Tampilan aplikasi ini sudah cukup bagus. Tapi mungkin kedepannya bisa dibuat supaya lebih menarik lagi."

4.4.6 Saran Pengujian *Beta*

Berikut adalah saran yang diberikan oleh Ir. Dedi Supriyadi, direktur CV. Saswco Perdana :

“Mungkin yang bisa ditambahkan di bagian analisisnya. Semakin banyak jenis grafik yang bisa ditampilkan akan jadi semakin bagus. Juga mungkin filternya bisa ditambah sehingga informasi yang tampil bisa dibuat lebih spesifik lagi.”

4.4.7 Kesimpulan Pengujian *Beta*

Berdasarkan hasil wawancara diatas menunjukkan bahwa tujuan dari pembangunan *data warehouse* ini sudah terpenuhi. Untuk kebutuhan-kebutuhan yang muncul dikemudian hari, maka akan dimasukkan ke tahap *maintenance*.